BALTIC CONFERENCE

Advanced Topics in

Telecommunication

Riga, 31.08.-01.09.2007

Universität Rostock 2008

Herausgeber: Prof. Dr. Clemens Cap
Wissenschaftsverbund IuK
“Informations- und Kommunikationstechnologie“ (IuK)
Universität Rostock

Erstellung der Druckvorlage:
Prof. Dr. Clemens Cap
Robert Kühn

Entwurf des Umschlagbildes:
Dr. Christine Bräuning

CIP-Kurztitelaufnahme:

ISBN:

Bezugsmöglichkeiten: Universität Rostock
Institut für Informatik
Frau Kerstin Krause
Albert-Einstein-Str. 21
18059 Rostock

Universität Rostock
Wissenschaftsverbund IuK
Frau Dr. Christine Bräuning
Albert-Einstein-Str. 23
18059 Rostock

Druck: Universitätsdruckerei Rostock

Table of Contents

Preface

In 2005, the University of Bremen, the University of Lübeck, the ISNM - International School of New Media at the University of Lübeck, and the University of Rostock combined forces for the first Baltic Sommer School in Technical Informatics (BaSoTi). Supported by a sponsorship by the German Academic Exchange Service (DAAD - Deutscher Akademischer Austausch Dienst), a series of lecture was offered between August 1 and August 14, 2005 at Gediminas Technical University at Vilnius, Lithuania. The goal of the Summer School was to intensify the educational and scientific collaboration of northern German and Baltic Universities at the upper Bachelor and lower Master level.

In continuation of the successful program, BaSoTi 2 was again held at Vilnius, from July 31 to August 14, 2006.

To turn the Summer School into a true Baltic event, BaSoTi 3 took place in Riga, Latvia, at the Information Systems Management Institute, from August 26 to September 10, 2007 and BaSoTi 4 presently is planned for August 8 to August 23, 2008 at the University of Tartu. Estonia.

After the second Summer School, the lecturers of BaSoTi felt the need for opening the event also for PhD students. The goal was to give young, aspiring PhD candidates the possibility to learn to give and to survive an academic talk and discussion, to get to know the flair and habits of academic publishing and to receive broad feedback from the reviewers and participants. Therefore the Summer School was augmented by the Baltic Conference on Advances in Telecommunication.

This book of proceedings is proof of the fine results produced by the participants and lecturers of BaSoTi 3.

Clemens H. Cap
Rostock, December 2007.

A Simulation Methodology for Distributed Storage

Peter Sobe, Moritz Baum, Sergej Heckel and Jan Krueger
University of Luebeck
Institute of Computer Engineering
Email: sobe@iti.uni-luebeck.de

Abstract: An eventbased simulation system is introduced, designed for performance assessment of distributed storage systems. Such storage systems employ the communication network to connect several distributed storage units to a virtual storage system that provides higher capacity, better performance and that is tolerant against storage unit failures. By simulation, several data distribution strategies and redundancy layouts can be analyzed, particularly focusing on effects caused by load imbalances and asynchronous operation of storage units. In this paper, a methodology is introduced to (i) simulate the effects of scaling the distribution degree and (ii) the effects of access load balancing techniques. This helps to design proper data and redundancy layouts for distributed data storage systems.

1 Introduction

Due to performance and dependability issues, storage systems are often built from several storage units. Depending on the implementation, these units can be connected by an I/O bus, a storage area network or by the communication network, as already present in a distributed computer system. For the first time, this conceptwas employed for RAID[KGP89] (Redundant Array of Independent Disks) systems at the level of magnetic disk spindles that worked in a synchronized way. Later, these principles have be adopted for distributed storage systems, as focused in this paper. By interleaving of data blocks, distributed storage systems allow to parallelize storage access and thus to provide better performance. This technique is commonly referred as data striping and can be applied with (i) different granularity - i.e. the size of blocks that are assigned to a single storage unit and (ii) with a different striping factor, i.e. the number of storage units involved. Striping offers performance improvements as long as the network between the storage units and the accessing process is faster than the access rate of a single storage unit. Another issue for distribution is the deletion-tolerant coding that allows to tolerate faults of storage units and increases the storage dependability. A condition for efficient deletion-tolerant codes is that data is distributed across several storage units that fail independently. With a high distribution grade, the amount of redundant data can be held relatively small related to the amount of effective user data.
In this paper, particularly the choice of data and redundancy layout and the understanding of effects that come with data distribution are focused. Whereby dependability can be assessed analytically, realistic performance is subject to be simulated or evaluated experimentally. In this paper, an eventbased simulation system is introduced, particularly designed for distributed storage system assessment.

2 State of the Art

A short state of the art summary related to distributed storage technology shall be given in 2.1. In 2.2, a second part, relations to performance evaluation techniques are given.

2.1 Distributed Storage

Initially, parallel storage was introduced by RAID[KGP89] systems, tighly coupled to magnetic disk technology. Relatively slow disks were combined to faster arrays. Data reliability was addressed by parity codes. Several levels of RAID systems got standardized by the RAID advisory board. Level 0 implements striping, level 3/4 combine data striping with a parity code that allows to tolerate a single disk failure. With level 5, the concept of redundancy interleaving was introduced that avoids bottlenecks with respect to disk access load. RAID level 6 extends level 5 by a second redundancy disk to tolerate two failures, for instance by using a Reed/Solomon[IR60, Pla97] code. Later, distributed systems were used for parallel data storage, e.g. PVFS[Tea03]. Along with that, deletion-tolerant coding developed to rely on higher number of storage resources, i.e. a higher distribution grade. Another effect is the disappearing synchroneity of the storage ressources used.
Only a few distributed systems combine parallel data access and deletion-tolerant codes. A couple of distribution and coding variants are implemented in the NetRAID[Sob03, SP06] middleware. This systems allows to store date reliably in a distributed system. Measurements showed that the access bandwidth scales well up to a number of 8 . . . 16 storage units and then does not grow further, unless saturating the full bandwidth of the communication network. These results accord to observations on other distributed storage systems that as well employ moderate striping factors. A major reason for that should be seen in the asynchronously operating storage units.
Deletion-tolerant codes strongly benefit from a higher distribution degree. The code rate in relation the the number of tolerated storage unit failures increases with increasing distribution degree. Beside, efficient XOR-based codes, e.g. Low density parity check codes [Gal63], approach an optimumof computational effectivenesswith increasing striping factors. Thus, it is highly demanding to understand the behavior of storage systems composed of a relatively large number of units. Some related techniques were used in the past for multimedia storage systems, particularly to provide the bandwidth for video storage using data striping across a few disks. Staggered striping [BGMJ93] limits the striping factor for each file. The striping group, i.e. the set of units used for storage, is shifted across a larger number of storage units. This strategy helps the distribute data and access load, while not increasing the striping factor too much.

2.2 Performance Evaluation of Storage Systems

As other distributed systems, the performance of distributed storage systems can be modeled by techniques like queuing networks or GSPN (generalized stochastic petri nets). Such description techniques allow to handle models analytically, but also may be simulated.

Queuing networks - base on single queuing systems with a source and a server. A queuing network is present, if at least two severs are included in a system, and if jobs are transfered along connections from one to another server. Analytical methods are present to handle closed networks, i.e. such networks with a constant number of jobs circulating in the system. Simulation tools are common, e.g. GPSS/H (general purpose simulation system) with a simulation–oriented programming language.

Generalized Stochastic Petri Nets - base on place/transition networks with tokens that reside on places and are created and consumed by the 'firing' of transitions. A GSPN is an extension of a petri net with timed transitions that fire after an exponentially distributed delay. Queuing networks can be expressed by GSPN. Further, extension with colored objects exist. GSPN are used for verification of parallel systems, but also for system behavior assessment. For a limited number of network models, analytical solutions can be found. Simulation of models is always possible by playing the token game, even though this can be time-consuming.

In spite many existing simulation systems we decided to design another one, called ***SIMple***. It is mainly based on the event-based simulation techniques described in [Gra92] and focused to simulations of parallel but connected activities, as needed for storage system principles.

3 Objective of Simulations

Simulations are related to the performance of distributed storage systems. Reliability is addressed as a quality that requires a particular redundancy layout and thus influences performance. Commonly, the better the ratio of redundant data amount to original data amount, the more reliable a storage system can be.

Mainly, quantitative studies related to two effects shall be taken.

- ***How many units can get employed efficiently for data striping?*** - By increasing the striping factor, ideally the data access bandwidth could be scaled up to the network bandwidth. Then as much data is transported to/from the storage units as the network can carry. This would require the synchronous operation of all storage units. In practice, such a scaling is hindered by asynchronous operation of the storage units. By simulation, this effect shall be quantified.
- ***How significant is the effect of redundancy interleaving?*** - Some distribution schemes interleave redundancy blocks on several storage units to overcome access

load bottlenecks for update operations. For example, RAID level 5 interleaves parity information of succeeding blocks onto several disks. Similar principles can be applied to other coding and distribution schemes, e.g. for Reed/Solomon-coded storage layouts. This code allows to add M redundant storage units to N data storage units (with M) and then to tolerate M faulty storage units. The coding principles are well studied (e.g. [BKK 95, Pla97]). The redundancy interleaving effects are often not considered, or left as implementation issues for efficient systems.

4 Simulation System

For modeling, we use an event-driven simulation system with queuing network elements. A network is defined by connecting queuing network elements by edges. Jobs are transfered between network elements (so called entities) along edges. Whereby the main entities are sources and servers, a couple of entities for controlling job flow are added. Related to storage systems, jobs express access activities directed to storage units. Entities represent accessing processes, storage units and relations between them.

4.1 Simulation Model Entities

All entities are simulated under an event-based control. The symbols for these different entity classes are shown in Figure 1.

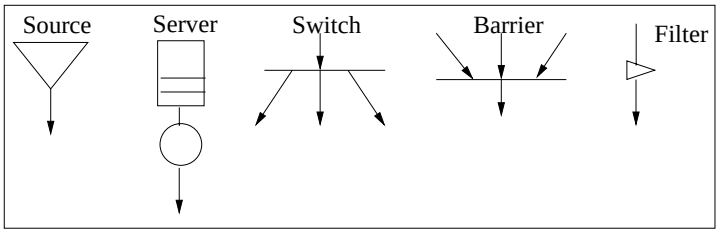

Figure 1: Simulation entities

Entities are connected by edges with other entities. Several edges may form the input to an entity so that jobs from different origins arrive at the entities input. Each entity can send jobs along an outgoing connection to other entities. When there is no outgoing connection, the entity acts as a sink for jobs. Entities can be parameterized and equipped with several measurement probes. The functionality of each class, the parameters and relevant measurement probes are described as follows:

- ***Source*** - A source generates jobs with a specified rate. The time between generating jobs is exponentially distributed. This can be used for classical queuing network

simulation. It is also possible to parameterize a source to generate solely a single job. In this case, the job is created at time zero and is commonly injected into a closed queuing network, i.e. a network with a constant number of circulating jobs. All created jobs can be assigned to a particular job class and later on, being distinguished from other jobs.

- ***Server*** - A server is a processing unit that processes jobs with a certain service rate. A FIFO queue is included to buffer incoming jobs until they get processed. The service time is exponentially distributed. A server can be equipped with a variety of measurement probes, e.g. queue length, server load or number of tasks processed.

- ***Switch*** - A switch receives jobs at a single input edge and sends out jobs along several output edges. Different variants of switches can be chosen. A clone switch multiplies each incoming job onto all outgoing edges. A round robin switch distributes incoming jobs among all outgoing edges. Here, the number of jobs is not multiplied, instead the jobs are distributed in a round robin strategy. A similar one is the random switch that distributes jobs randomly among outgoing edges.

- ***Barrier*** - A specified number of jobs is collected and then reduced to a single job by a barrier element. For that operation, incoming jobs are held in buffers related to the incoming edges. For the output of a job, incoming jobs (i) are always taken from different incoming edges/buffers and (ii) have to be out of the same job class. The output job class is chosen according to the class of the selected incoming jobs.

- ***Filter*** - Jobs can be assigned to a job class, when generated by the source. A filter entity is used to receive incoming jobs, forward jobs of a specified class and to discard all other jobs.

Figure 2 depicts a system with two servers. Jobs generated by a source are randomly assigned to these tow servers. Such a model is an appropriate abstraction of two storage servers that share the load of a single application server. For more complex simulations, one may add servers to represent additional data storage units and a number of redundancy storage units. Network structures with switches, barriers and filters, together with several job classes can be used to express a wide range of access strategies.

4.2 Processing Technology

Simulation models are specified in a textual way. A particular syntax allows to fetch the model description by the JAVA dynamic class loader and create objects for all necessary simulation entities.

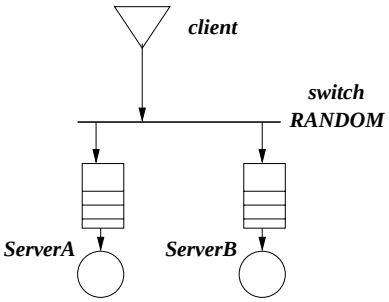

Figure 2: Small example of a simulation model.

```
client : Source { generationRate = 5.0; }
switch : RandomSwitch;
serverA : Server {
  serviceRate = 3.0;
  * averageWaitTime;
  * tasksProcessed;
}
serverB : Server {
  serviceRate = 3.0;
  * averageWaitTime;
  * tasksProcessed;
}
client -> switch;
switch -> serverA;
switch -> serverB;
```

The listing above shows the textual description of the system given in Figure 2. Probes for reporting the average waiting for jobs and the number of processes jobs are assigned to both servers. The network structure is described by a list of entity connections.

4.3 Methodology

The effects caused by asynchronous operation of the storage units are modeled by the network depicted in Figure 3, left part. It clones jobs to several servers, collects the results by a barrier. These jobs represent connected accesses, i.e. accesses that belong together and have to be finished before the next activity is initiated by the client. The network contains a closing loop edge that models a new access after each fulfilled access. The results presented in the next section were got by scaling the number of servers and correspondingly increasing the service rate. The number of fulfilled accesses is expressed by the number of jobs collected after a barrier. This can be easily monitored by a measurement probe.

Congestion effects caused by redundancy updates (a.k.a. small update problem) is modeled by cloning jobs into a redundancy update access and another access directed to a single data storage unit that is chosen randomly. The networks used for studying these ef-

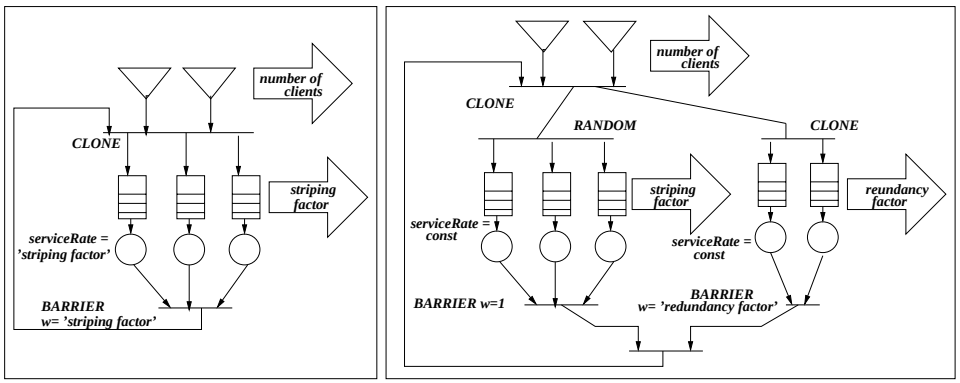

Figure 3: Networks for scaling experiments.

fects are similar to that one shown in the right part of Figure 3. Experiments cover systems without the rightmost server as a system without redundancy as the basis for comparison. The redundancy interleaving technique as well as the staggered striping approach is modeled by distributing accesses in a regular manner onto the servers. These jobs represent connected accesses that have to be joined after serving. This can only be implemented using job classes. Circulations of different job classes are shown in Figure 4. In this very small example, a system with two data and a single redundancy storage unit is shown. Redundancy is interleaved across the three storage units.

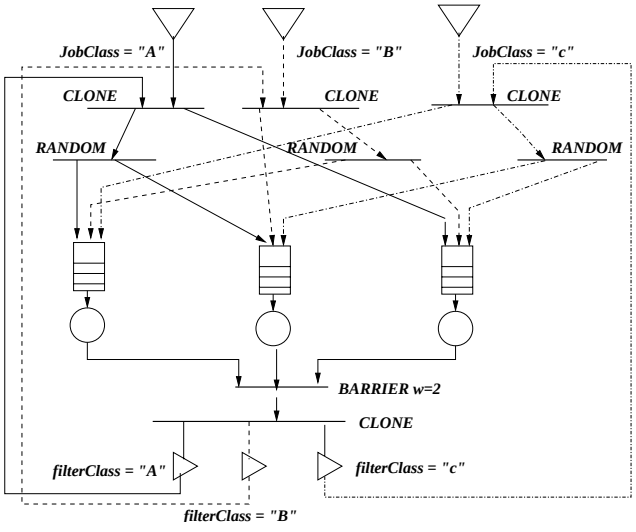

Figure 4: Network to discover effects of redundancy storage interleaving.

5 System Analysis

The influence of the striping factor on the access performance is depicted in Figure 5. The performance increase (speedup) is found to be significantly lower than the number of resources, i.e. the number of storage units. A single accessing client that generates connected accesses can not fully utilize the storage units. Using more clients, i.e. more circulating jobs in the system leads to more fulfilled accesses in total. This can be explained by the utilization of time gaps by other accesses. This effect is assumed to intensify with growing number of clients in the system. Thus, several combined accesses should be used to utilize the system in a better way.

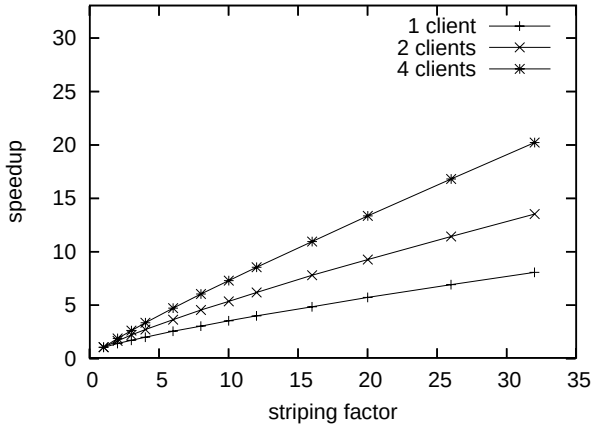

Figure 5: Scaling of striping factor

The cost of redundancy update for small write operations and the effects of redundancy interleaving are shown in the following plot (Figure 6). Accesses are randomly distributed across storage servers. System configurations with 5, 8, 12 and 16 storage units were analyzed - these numbers correspond to common disk array configurations. Without redundancy (RAID level 0), the number of accesses scales well with the system size. A single dedicated unit for redundancy introduces a bottleneck, as expected. This situation is present for RAID level 4. This bottleneck can be diminished by interleaving the redundancy, as shown for RAID level 5.
Systems with multiple redundancy units such as Reed/Solomon codes profit from interleaving of the redundancy as well, as shown in Figure 7. Nevertheless, the advantage is smaller compared to systems with single redundancy unit. This follows from overlaps in the redundancy assignments which can not be avoided as long the interleaving takes place within the original set of data and redundancy storage units. An adaptation of the staggered striping principle is capable to solve this problem. By distributing 7+2 data and redundancy blocks onto $2 \times (7+2)$ storage units with a staggered allocation of data and redundancy blocks we could reach 13700 accesses instead of 8500 accesses for the non-staggered allocation. This number is equal to the number of accesses reached with

interleaving of a single redundancy storage unit.
As a consequence, for system swith frequent small updates, a the classical redundancy interleaving technique should be combined with staggered block allocation onto a larger number of storage units.

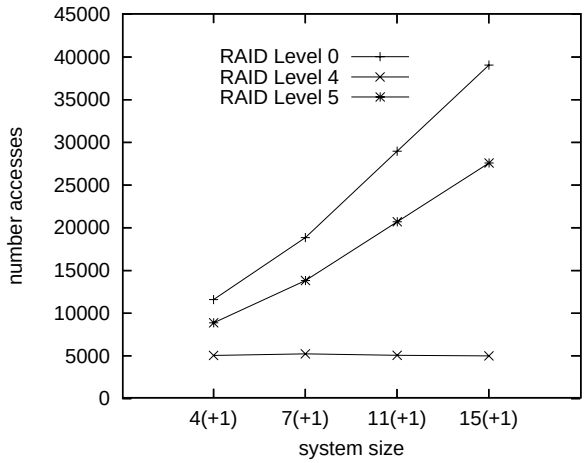

Figure 6: Influence of a single redundancy storage unit without (level 4) and with redundancy interleaving (level 5). Striping without redundancy at all (level 0) is shown for comparison.

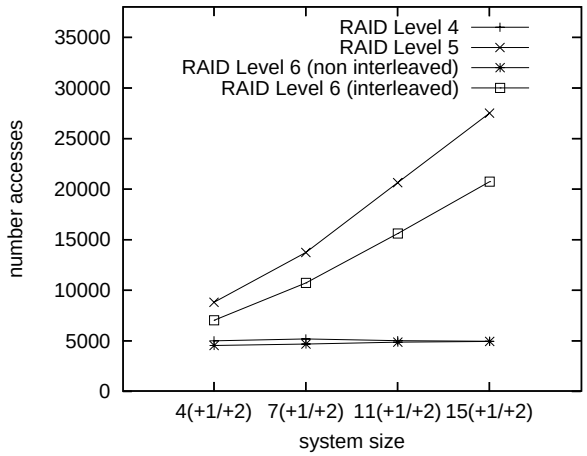

Figure 7: Influence of redundancy interleaving for two redundant storage units (level 6). The schemes with a single redundancy unit (level 3 and 5) are shown for comparison.

6 Summary and Future Work

An event-based simulation system with specific elements for storage system modeling was introduced. Such a system gives insights into effects of asynchronous operation and performance of such a distributed system. Particularly, asynchronous operation of units that are required to provide functionality collectively limit the performance gain. As a second aspect, the influence of redundancy interleaving on the update performance was studied. As result, staggered allocation of data and redundancy blocks onto a number of storage units larger than the originally needed number is advised. We plan to use the simulation system for performance prediction of storage techniques with a high distribution grade but also with a high ratio of redundancy (e.g. low density parity check codes). Obviously, the simulation system has to be extended to allow more realistic service time distributions. Parameters will be taken from real storage unit operation scenarios. In this way, widely distributed storage layouts, e.g. used for global storage can be evaluated.

References

[BGMJ93] S. Berson, S. Ghandeharizadeh, R. Muntz, and X. Ju. Staggered Striping in Multimedia Information Systems. Technical Report CSD-930042, University of California, Computer Science Deptartment, December 1993.

[BKK 95] J. Blomer, M. Kalfane, M. Karpinski, R. Karp, M. Luby, and D. Zuckerman. An XOR-based Erasure–resilent Coding Scheme. Technical Report TR-95-048, International Computer Science Institute, August 1995.

[Gal63] R.G. Gallager. *Low-Density Parity-Check Codes*. MIT Press, Cambridge, MA, 1963.

[Gra92] T. Grans. *Simulation - Strukturiert und objektorientiert programmiert*. BI Wissenschaftsverlag Mannheim, Leipzig, Wien, Zuerich, 1992.

[IR60] G. Solomon I. Reed. Polynomial Codes over Certain Finite Fields. *Journal of the Society for Industrial and Applied Mathematics [SIAM J.]*, 8:300–304, 1960.

[KGP89] R. Katz, G. Gibson, and D. Patterson. Disk System Architectures for High Performance Computing. In *Proceedings of the IEEE*, pages 1842–1858. IEEE Computer Society, December 1989.

[Pla97] J. S. Plank. A Tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-like System. *SOFTWARE - PRACTICE AND EXPERIENCE*, 27(9):995–1012, Sept. 1997.

[Sob03] P. Sobe. Data Consistent Up- and Downstreaming in a Distributed Storage System. In *Proceedings of the International Workshop on Storage Network Architecture and Parallel I/Os*, pages 19–26. IEEE Computer Society, 2003.

[SP06] P. Sobe and K. Peter. Comparison of Redundancy Schemes for Distributed Storage Systems. In *Proceedings of the 5th IEEE International Symposium on Network Computing and Applications (NCA)*. IEEE Computer Society, 2006.

[Tea03] PVFS2 Development Team. Parallel Virtual File System, Version 2. http://www.pvfs.org/pvfs2/pvfs2-guide.html, 2003.

Erasure-tolerant Codes for Rule-based Grid Storage Systems

Kathrin Peter
Zuse Institute Berlin
Computer Science Research
Email: kathrin.peter@zib.de

Abstract: Distributing data in a wide area storage system requires to cope with unreliable components. Because of non-reliable resources, data may be temporarily unavailable or even get lost. Nevertheless a user of a grid storage system should have unlimited access to his data even if some components are down. Replication is no solution because it takes a multiple of the storage space of each file while the reliability increases only moderate.

Our approach is to use erasure-tolerant codes like Reed-Solomon which can tolerate a higher number of simultaneous unavailable resources with less redundancy compared to replication[1]. Using an existing rule-based grid storage system we show how the system must be extended to make the storage reliable. Other storage systems with similar architectures could also use this approach to provide more reliable storage.

1 Introduction

Today distributed systems are used to accomodate the need for computational power. Grids as an example accomplish compute resources distributed over a large number of institutes. Joining a virtual organisation opens users the possibility of transparent access to all resources in the grid. Among computational resources there is also a need of storing data in a fast, secure and reliable way. Grid storage systems aim to provide transparent access to data distributed in wide area. A grid storage system opens the possibility of data sharing. Researchers get easy access to all grid storage resources and may choose between different views to their data, e.g. with a file-browser. Users need only one account in the storage system to access data at different sites or to share data with other users or groups. A lot of existing wide-area and grid storage systems already provide this transparent access[2].

However the reliability of storing data in those systems is neglected. Some systems offer the possibility to replicate data to other resources but this is accomplished only with a high overhead of redundant data [WK02]. The use of erasure-tolerant codes is a reasonable way to make storage reliable and a common used technique in RAID systems [PGK88] for local cluster storage systems. But in contrast there exists no wide-area storage system which already applies erasure-tolerant codes in a transparent way.

In this paper we show how an existing storage system *iRODS* [MRW+07] can be extended by additional rules for higher encoding functionality. An outline of the encoding rule is

[1] Adding m redundant data blocks to a number of n original data blocks can tolerate up to m simultaneous erased data blocks.

[2] Grid storage systems which provide transparent access are for example Storage Resource Broker (SRB) [BMRW98] and the successor system: integrated Rule-Oriented Data System (iRODS) [MRW+07].

given and we explain how to integrate the Reed-Solomon implementation. Furthermore we show in general how encoding effects the reliability of a storage system compared to replication. To the best of our knowledge there exists no wide-area storage system that uses erasure-tolerant codes for increasing storage reliability and is really used in production systems.

2 Reliable data storage in grid environments

Grid Storage Systems give transparent access to a large number of resources. Their aim is to provide a logical view to the data and to hide information about real physical resources. Users for example want to put data into the system and get it out without considering which and where resources are running. Data objects in the system should be available even if some resources are temporarily down, disconnected or failed.

Data can be stored centralized at one resource or distributed by splitting up the data object into blocks and distribute them across different resources. This mechanism called striping is also used in cluster storage systems for faster access [CLRT00] but has the disadvantage of decreased reliability dependent on the number of storages. Figure 1 shows the reliability $R_G(d, n)$ of a storage system with a number of storages between one and hundred and a runtime from zero to fifty days.

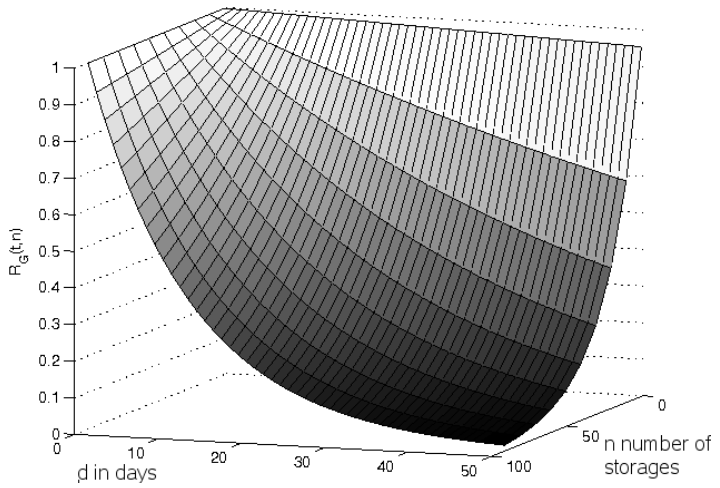

Figure 1: Reliability without redundancy (RAID 0)

$R(d) = e^{-\frac{1}{1095} * d}$ reliability of one disk with MTTF 1095 days and runtime d.

$R_G = R(d)^n$ reliability of the system with n disks.

As visible the reliability decreases exponentially with the number of storages n and also

with the uptime in days d. RAID (RAID 1 - RAID 6) [PGK88] are common used techniques to increase the reliability of distributed storage systems by adding redundancy. Contrary to parity and more sophisticated codes replication is the typical way to make data storage in WANs like peer-to-peer and grid systems more reliable and faster. One reason is the trade-off between higher computational effort, metadata management and reliability [WK02]. Our focus is the reliable storage of data in a grid, which is a requirement in a lot of different research communities. One example is MediGRID with heterogeneous applications from medical and biomedical researchers [KPS+07]. Especially in medicine long-term storage of patient data is required by law.

3 Rules for reliable data storage in iRODS

Erasure-tolerant coding is added to iRODS by implementing rules which contain encoding and storage steps (detailed information on iRODS see sections 3.2 and 3.3).

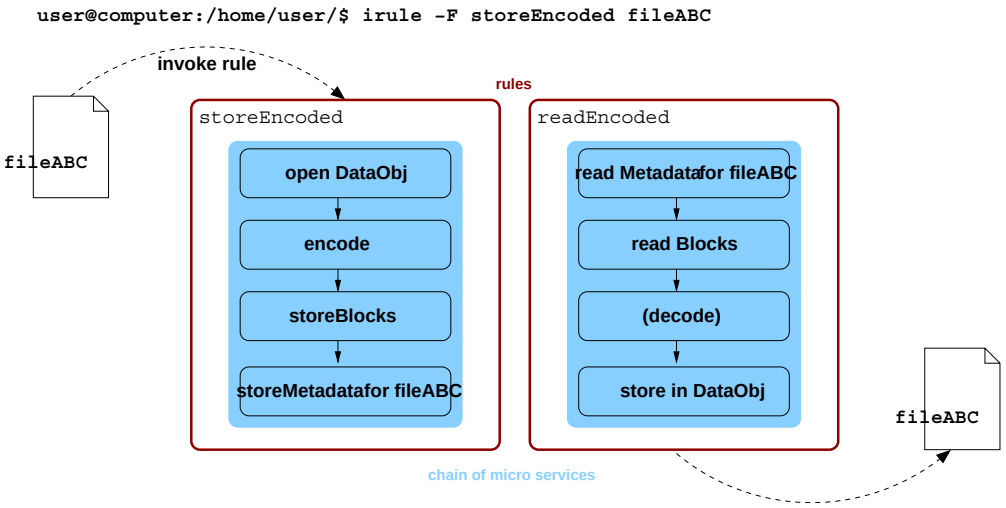

Figure 2: Invoke encoding rule for put and get

Rules can be used instead or additionaly to the standard icommands like `iput` for example [3]. The brief overview in figure 2 shows the steps inside of an encoding rule `storeEncoded` that stores data into iRODS and a rule `readEncoded` that reads data from iRODS. The data object is at first encoded by using the Reed-Solomon encoding operation. The result are original data blocks and redundant data blocks. Next datablocks are stored across different resources. Metadata is necessary to store the information which blocks belong together. For reading all available blocks have to be found. Decoding is necessary only if some data blocks are not available. Depending on the configuration it is

[3] iput copies a file from the local filesystem into iRODS

possible to recover a given number of erased original data blocks with the help of redundant data blocks.

The advantage of using iRODS rules is that encoding can be done transparent and hidden from the user inside of the rules. We achieve transparency because the call of our rule in figure 2 is similar to the call of any other rule in iRODS. Here we assume iRODS to operate properly, independent of failed storage resources as long central resources in the system are not faulty (eg. central metadata catalog).

3.1 Reed-Solomon Codes

Reed-Solomon is a non-binary cyclic block code [IR60]. It is applied for channel coding and storage systems [Pla97]. A deletion-tolerant variant of this code interprets data and redundancy as a system of linar equations. Encoding is done by a matrix vector multiplication:

$$\begin{bmatrix} A \end{bmatrix} \begin{bmatrix} D \end{bmatrix} = \begin{bmatrix} E \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 \\ f_{1,1} & f_{1,2} & \cdots & f_{1,n} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ f_{m,1} & f_{m,2} & \cdots & f_{m,n} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \\ b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \tag{2}$$

A is called generator matrix, D is the vector with original data words, E is the result vector with original and redundant words. Formulae (1, 2) express this equation system interpretation of data and check words. In this way, missing data can be recalculated by expressions that are got by solving the linear equation system in the case of an error. The rows in matrix A and the elements of vector E that belong to faulty storages are deleted.. It is possible to tolerate up to m simultaneous failures and we get a $n \times n$ square matrix A' after deletion. Data words in D can be recalculated by a matrix-vector multiplication of the reversed matrix A' and the modified vector E' (formula 3).

$$\begin{aligned} & \begin{bmatrix} A' \end{bmatrix} \begin{bmatrix} D \end{bmatrix} = \begin{bmatrix} E' \end{bmatrix} \\ \Leftrightarrow \quad & \begin{bmatrix} D \end{bmatrix} = \begin{bmatrix} A' \end{bmatrix}^{-1} \begin{bmatrix} E' \end{bmatrix} \end{aligned} \tag{3}$$

It is necessary to calculate in Galois-field arithmetics because of the finite floating point precision. For our calculations we use a word size of $\omega = 8$ Bit and a Galois field with 256 elements (GF(2^8)). It is recommended to choose a word size as a multiple of 8 (one byte). The wordsize ω has impact on the maximal system size: $2^\omega \geq n + m$. We implemented the encoding and decoding algorithm and integrated it in the distributed storage system NetRAID [SP06]. These encoding and decoding C-functions are the basis for the micro-service implementation in iRODS.

3.2 Introduction to iRODS (integrated Rule-Oriented Data System)

iRODS [MRW$^+$07] as the successor system of SRB (Storage Resource Broker) [BMRW98] is developed at San Diego Supercomputing Center (SDSC). Currently SRB is used as the datamanagement component in a lot of projects and SDSC will offer migration tools to switch from SRB to iRODS in future. iRODS has a client-server architecture with one central metadata catalog and at least one server to access the catalog (figure 3).

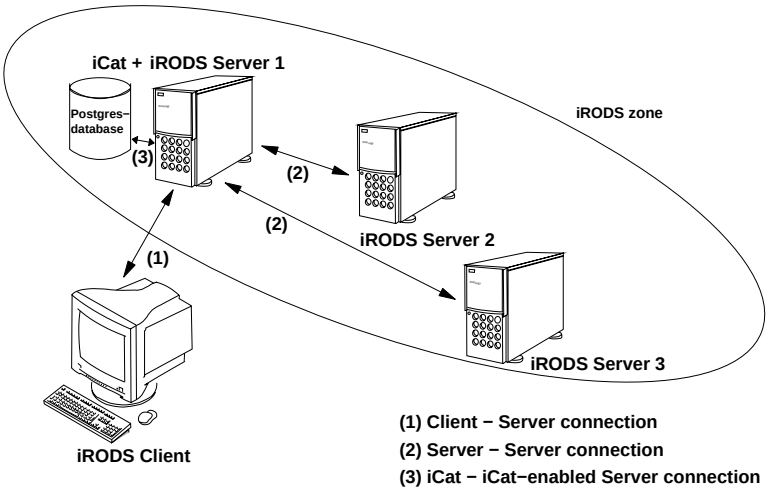

Figure 3: iRODS system example

The system is extendable by installing iRODS server at other resources and including resource information into the metadata catalog. The metadata catalog (called *icat*) is the most important system component because it contains all informations about users, resources and data objects. A user can access the system with the unix-like command line interface *icommands* and similar to SRB there will exist other APIs and graphical user interfaces.

The advantage of iRODS is the extensibility of the system capabilities by defining rules. User groups can add functionalities encapsulated into rules and share them with other user groups (section 3.3).

3.3 Rules and Micro Services in iRODS

The architecture of iRODS enables system programmers to add sophisticated functionalities. iRODS provides users with an API called *icommands* which can be used for example to put data into the system (`iput`), getting data out (`iget`) or replicating data to other resources (`irepl`). Additionally to icommands users can invoke particular rules by calling the API command: `irule RULEID`. Depending on conditions the rule invokes an action. Actions are workflows or chains of smaller tasks which are other actions or micro-services. A micro-service encapsulates a certain function (figure 2). Parameters can be passed between services. System programmers can expand the rule base by implementing own rules. Different conditions and workflows can be integrated into these rules and contain calls of basic or added micro-services.

3.4 Technical details

In this paragraph we give more details on our first version of integrating data encoding with the Reed-Solomon algorithm in a grid storage system. The implementation is still in progress and will probably change again in future because the iRODS system itself is currently under development too and only available in version 0.9.2. We present a concept of integrating codes into this rule based system. A full iRODS system version 0.9.2 is installed at a single machine. This is the main installation with the *icat metadata catalog* [4] and an *icat server* that interacts with the metadata catalog. To integrate other resources one need to install a *non-icat server* at those machines. These servers are configured to belong to one *zone* (*icat metadata catalog* + *icat server*).

Our design wraps the standard put and get commands from iRODS through scripts that encode data and store it at iRODS.

Two problems of distributed storage in contrast to standard replication in iRODS must be solved:

1. Avoid for single points of failure: There must not exist a single metadata file or a single data file which is lost if one resource fails. Metadata provide information which blocks belong together and belong to one file respectively.
2. Transparent data storage: Encoding should not be visible to the user. If a user reads data in his iRODS home directory it should look as a single stored file. Other iRODS access functions should work properly.

We developed two models for encoded storage.

Script 1: Store and Encode delayed

- `iput FILE` stores FILE in iRODS and creates all default metadata.

[4] default: postgresql-8.2.3 database

- `irule encodeDataR1` calles a rule to execute the encoding steps:
  - Open file
  - Encode data
  - Create $n+m$ files for data blocks and redundant blocks in a separate encoding directory. The encoding directory is not visible to users.
  - Create metadata (attribute-value pairs) to each file with its coding information.

Script 1 uses the icommand `iput` to create the file itself and metadata automatically in iRODS. A disadvantage is the single point of failure if this main-file is not available. Replication of this file could help to avoid that. An transparent view is possible because the main-file is the only visible data object for the user while blocks can be stored in a separated encoding directory distributed across storage servers.

Script 2: Real Distribution

- no single datafile is created
- `irule encodeDataR2` calles another encoding rule:
  - Encode data
  - Create $n + m$ files for data blocks and redundant blocks
  - Add metadata (attribute-value pairs) to each file

    ```
    mainFile = FILE
    datablock = D2
    schema = RS 5+2
    ```

  - if the user acesses a file it has first to be checked wether the distributed flag is set or if the file is stored regular.
  - if it is stored in a distributed way all files with attributes `mainFile = FILE` have to be accessed and combined to the requested output file.

The advantage of this distributed variant is no single point of failure. But there is a higher effort to create the transparent view for the user.

Finally we analyze the theoretical increase of reliability by coding data with a Reed-Solomon code in contrast to replication. By knowing the number of independent storages and the probability of each storage to be available we calculate the availability of a data object stored according to a certain schema. In this example we compare the availability of replicated data with Reed-Solomon encoded data.

We define the following parameters:

n : Number of data blocks

m : Number of redundant blocks. For equal comparison we set m as a multiple of n.

$all := n + m$ Number of blocks (original and redundant blocks).

p : The probability that one node is available. $p_{node} = e^{-\lambda t} = 0.8331$ with $\lambda = \frac{1}{MTTF}$, $MTTF = 1095$ days and t=200 days runtime.

p_{repl} : Availability of replicated data.

p_{RS} : Availability of Reed-Solomon encoded data.

r : Replication factor (1=one original block, 2=one original block and one replica).

In table 1 we calculate the availability of a data object which is striped across 8 data nodes. Replication adds copies of each stripe to additional storage nodes. Schema $8 + 8$ adds 8 redundant blocks to 8 original blocks (one replica for each block) and $8 + (2 * 8)$ adds 16 redundant blocks (two replicas for each block). With Reed-Solomon codes we can add an arbitrary number of redundant blocks but for fair comparison we add also at first 8 and then 16 redundant blocks to the 8 data blocks and compare it to replication.

Schema	no redundancy	8+4	8+8	8+2*8
Replication	0.8331	-	0.7976	0.9634
Reed-Solomon	0.8331	0.9634	0.9996	1.0

Table 1: Availability of replicated and RS-encoded data objects

As visible adding 8 redundant blocks to 8 data blocks with replication is less reliable than encoding them with Reed-Solomon. For the Reed-Solomon encoded data much more failure combinations can be tolerated. We calculate these results using the following formulae 4 and 5:

$$p_{repl} = \left(\sum_{k=0}^{r-1} \binom{r}{k} p^{r-k} * (1-p)^k \right)^{\frac{n}{r}} \tag{4}$$

$$p_{RS} = \sum_{k=0}^{m} \binom{all}{k} p^{all-k} * (1-p)^k \tag{5}$$

4 Related work

Erasure-tolerant codes can be divided into different classes depending on the complexity of coding operations [PGK88]. Replication of data does not require any additional computation and is used in peer-to-peer and grid storage systems like SRB [BMRW98]. RAID 3 coding or parity coding adds exactly one data block to an arbitrary number of original data blocks. The redundant data block is the XOR combination of the original data blocks. Parity codes are used by RAID 3 or RAID 5 for example in cluster systems for parallel distributed data storage [Sob03].

More sophisticated schemes like Reed-Solomon code [Pla97] need a higher calculation effort. NetRAID implements nearly all kinds of parity and Reed-Solomon code. Cluster-RAID [Kal04] uses Reed-Solomon to make the local disk more reliable. For wide area networks there are a lot of examinations and comparisons of replication against Reed-Solomon [WK02]. The focus is the degree of fault tolerance, update effort, and additional network traffic. Real deployments of Reed-Solomon code in wide area systems exist for Oceanstore [KBC+00] and are described in [PMSN06]. Oceanstore [KBC+00] applies Reed-Solomon for the archival layer. In [PMSN06] original data is splitted into blocks. Next redundant blocks are calculated. Datablocks are distributed in the network transfered by gridFTP [ACF+05]. Data sharing is difficult because metadata information about the location of datablocks is stored at client side and not available in the system.

5 Conclusion

We introduce two models of integrating Reed-Solomon code into the rule-based storage system iRODS. We show the expected increase of reliability by using codes in contrast to replication. Additionaly on the top of our encoding rules could be build up higher functionalities. For example depending on the choosen degree of reliability the system should choose the encoding and distribution parameters autonomous. Before storing data blocks at arbitrary resources the system can check the actual availability and load of all integrated resources and decide for the best resource.

References

[ACF+05] Bill Allcock, Ann Chervenak, Ian Foster, Carl Kesselman, and Miron Livny. Data Grid tools: enabling science on big distributed data. *Journal of Physics: Conference Series*, 16:571–575, 2005.

[BMRW98] Chaitanya Baru, Reagan Moore, Arcot Rajasekar, and Michael Wan. The SDSC Storage Resource Broker. In *CASCON '98: Proceedings of the 1998 conference of the Centre for Advanced Studies on Collaborative research*, page 5. IBM Press, 1998.

[CLRT00] Philip H. Carns, Walter B. Ligon III, Robert B. Ross, and Rajeev Thakur. PVFS: A Parallel File System for Linux Clusters. In *Proceedings of the 4th Annual Linux Showcase and Conference*, pages 317–327, Atlanta, GA, 2000. USENIX Association.

[IR60] G. Solomon I. Reed. Polynomial Codes over Certain Finite Fields. *Journal of the Society for Industrial and Applied Mathematics [SIAM J.]*, 8:300–304, 1960.

[Kal04] Sebastian Kalcher. Optimization of a Distributed Fault-Tolerant Mass Storage System for Clusters, 2004. Diplomarbeit, Universitt Heidelberg.

[KBC+00] John Kubiatowicz, David Bindel, Yan Chen, Patrick Eaton, Dennis Geels, Ramakrishna Gummadi, Sean Rhea, Hakim Weatherspoon, Westly Weimer, Christopher Wells, and Ben Zhao. OceanStore: An Architecture for Global-scale Persistent Storage. In *Proceedings of ACM ASPLOS*. ACM, November 2000.

[KPS+07] Samatha Kottha, Kathrin Peter, Thomas Steinke, Julian Bart, Jürgen Falkner, Anette Weisbecker, Fred Viezens, Yassene Mohammed, Ulrich Sax, Andreas Hoheisel, Thilo Ernst, Dietmar Sommerfeld, Dagmar Krefting, and Michael Vossberg. Medical Image Processing in MediGRID. 2007.

[MRW+07] Reagan W. Moore, Arcot Rajasekar, Michael Wan, Wayne Schroeder, Yannis Katsis, Dayou Zhou, Alin Deutsch, and Yannis Papakonstantinou. Constraint-based Knowledge Systems for Grids, Digital Libraries, and Persistent Archives. Technical report, San Diego Supercomputer Center, University of California, San Diego, May 2007. Final Report.

[PGK88] David A. Patterson, Garth Gibson, and Randy H. Katz. A Case for Redundant Arrays of Inexpensive Disks (RAID). In *SIGMOD '88: Proceedings of the 1988 ACM SIGMOD International Conference on Management of Data*, pages 109–116, New York, NY, USA, 1988. ACM Press.

[Pla97] J. S. Plank. A Tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-like Systems. *SOFTWARE - PRACTICE AND EXPERIENCE*, 27(9):995–1012, September 1997.

[PMSN06] M. Pitkanen, R. Moussa, M. Swany, and T. Niemi. Erasure Codes for Increasing the Availability of Grid Data Storage. *Telecommunications, 2006. AICT-ICIW '06. International Conference on Internet and Web Applications and Services*, 0, 2006.

[Sob03] P. Sobe. Data Consistent Up- and Downstreaming in a Distributed Storage System. In *Proceedings of. Int. . Workshop on Storage Network Architecture and Parallel I/Os*, pages 19–26. IEEE Computer Society, 2003.

[SP06] P. Sobe and K. Peter. Comparison of Deletion-tolerant Codes for Distributed Storage Systems. In *NCA06 Proceedings*. IEEE Computer Society, 2006.

[WK02] H. Weatherspoon and J. Kubiatowicz. Erasure Coding vs. Replication: A Quantitative Comparison. In *IPTPS*, 2002.

Integrating Economic Psychology into Data Mining Methods in the Context of a Supermarket Recommender System

Anne Gutschmidt[1]
Email: anne.gutschmidt@uni-rostock.de

Abstract: We introduce a new concept of a supermarket recommender system. Position tracking, economic psychology and market basket analysis are combined to improve present recommendation methods.

1 Introduction

Recommender systems present product or link suggestions, calculated from the users' former behavior and preferences. Treating each user individually, they are a means of attracting and maintaining the users' attention and ensuring customer loyalty. We describe how a recommender system can be applied in the "real" physical world, specifically, in a supermarket. Whereas existing work often concentrates on the technical implementation of a recommender system, we focused on creating a general concept which brings together totally different knowledge areas: economic psychology and data mining.
Using a Wireless Local Area Network (WLAN) and Personal Digital Assistants (PDA) attached to the shopping carts we deduce from the position data what kind of purchase decision a customer is currently involved in. The class of purchase decision reveals whether a person has a disposition to act habitually or spontaneously, or whether the person does a lot of deliberating before a decision. This knowledge is combined with a data mining method called association analysis. Common recommender systems often rely only on statistical relations they find in the data [AT05]. We use the purchase decision classes to filter those recommendations calculated with the association analysis which do not semantically fit. Thus, product suggestions based on causal relations are promoted.
During the last years, there have been several approaches to increasing customer loyalty by using new technologies in supermarkets. The Metro Group, a large German retailer, introduced the "Future Stores" where several technologies, especially RFID and WLAN, enable new services for the supermarket customers [Loe05]. The system proposed in [CFGK05] presents supermarket customers with a predicted shopping list on a cart-mounted Tablet PC. Loyalty cards enable access to former purchases of the customers. However, we do not want the customers to give away personal shopping data via loyalty cards nor scan the chosen products which, we claim, will make the customers be more open towards our system.
In the next section we will summarize the knowledge our recommender system is based on, followed by our concept of the system. In the last section we will give a short conclusion.

[1] Supported by the German Research Council (DFG), Graduate School 466

2 The Essential Components of the Recommender System

The essential components our recommender system is built of are the principle of *purchase decision classes*, the *location tracking via WLAN* and the *calculation of product suggestions based on association analysis*.

The Purchase Decision Process. In economic psychology, the term purchase decision process includes everything happening from the perception of a need to the actual purchase and consumption of a product. For our recommender system, we use a fundamental classification principle concerning decision behavior originating from the 1950s and 1960s. The different classes of purchase decisions are distinguished by the extent of cognitive and emotional involvement and the level of automation of actions. The classification comprises four groups of decision behavior:
First, ***extensive*** decisions are the most complex and include much cognitive effort. Usually, the decision is made for the first time. At the beginning, the consumers only have a rough idea about what they need. The idea gets more precise while they collect information about the products until a final decision is made. Secondly, ***limited*** decisions are similar to extensive decisions, yet, the cognitive effort is limited as the term already implies. The purchasers are aware of a small group of product alternatives between which they have to decide. They already have experiences and knowledge they can use to make their decision. Emotional processes and automatic behavior do not play a role. Thirdly, the ***habitual*** deciders have a disposition towards routine purchases. They always buy the same products and the same brands to reduce risks. Cognitive effort and emotional involvement with the purchase are not part in this class of decision behavior. These deciders act rather automatically. Finally, ***impulsive*** deciders represent the class in which emotional processes are of high importance. They react in a spontaneous and automatic way; i.e. they perceive a stimulus and react without thinking [KRW03][Kat51][HS69].
We will show that this classification can be applied to the different ways customers move in the supermarket.

Tracking Customers Using WLAN. Inside the supermarket, we track customers with the help of a WLAN and a PDA attached to each shopping cart. At least three access points have to be installed in the shop. The PDA uses the signals coming from the access points to determine its own position and sends it to a server where a self-developed Profile Manager software creates a movement profile. This profile comprises the time of entering and leaving the store, the overall length of the stay and the customers' path through the place and the speed. Instead of using point coordinates, the position tracking refers to areas in the supermarket. The areas are as small as the positioning system allows and of equal size. We tested the Profile Manager with the Ekahau Positioning Engine whose accuracy lies at approximately one meter.[2] So, normally the areas correspond to product shelves and the according product categories. A customer's path is depicted as a sequence of areas. Moreover, every stay is recorded, that means the areas where the cart came to a halt. The customers' speed is calculated by counting the number of areas crossed without halt and relating them to the overall time the person walked through the store.

[2] Specification of the Ekahau Positioning Engine to be obtained on http://www.ekahau.com.

Calculating Recommendations. Most methods used with recommender systems come from the area of data mining, which is about finding patterns in huge data sets. We chose a commonly used method called association analysis or association rule mining. Association rules take the form "If the customer buys product A, he will be likely to buy product B as well." To find the rules, a database is needed containing the purchases, in our case the paths of former customers. Statistical indicators called support and confidence reflect a rule's statistical significance and strength, respectively. Minimum values for support and confidence have to be set to enable the decision which association rule is relevant. If the rule above has sufficient support and confidence, a new customer choosing A will get B as recommendation [HK01].

3 The Concept of the Recommender System

3.1 Deriving Purchase Decision Classes

We identify the class of purchase decider by the customers'movements. It specifies which precise product to recommend from the shelf calculated with the association analysis and how to present it. We determine a number of rules for the derivation of a purchase decision class. The required information is the customers' speed, at what shelves they come to a halt, how long they stay there and the number of times they stop during a certain period of time. The number of stays per time period gives hint whether a customer stops often or seldom compared to other customers. All three parameters are either high or low with a customer with a fixed bound dividing the range of speed, of the duration of stay and of the number of stays into low and high. These bounds strongly depend on the size of the shop and have to be tested in each supermarket. If the supermarket is, for instance, small with little space between the shelves, the customers will probably move more slowly than in a huge shop with broad aisles. They will also stay more briefly and halt more often. The three movement characteristics, speed, duration of stay and number of stays, determine the type of purchase decider in the following way:
Habitual deciders know exactly what they want. They move fast and single-mindedly approach the products they want. They do not hesitate at the shelves but continue their shopping right after putting the product into the cart. ***Limited*** deciders know what they need, but they have not committed themselves to a certain product. Therefore, they move quickly, yet they stay long at the shelves to consider product alternatives. ***Extensive*** deciders move slowly through the store to inspect the offers. They take their time at the product shelves to study the product alternatives to come to a decision. ***Impulsive*** deciders move slowly through the supermarket as a way of wandering and looking at the offers. They may both stay long and shortly at the shelves; e.g. they may shortly halt to look at a product or they may take their time to search discount offers at a rummage table. At this point, we cannot distinguish the impulsive deciders who stay long at the shelves from the extensive deciders. Hence, we include the number of stays during a determined time period. We claim that impulsive deciders stop considerably more often than extensive deciders. In contrast to impulsive deciders, extensive deciders focus their attention on only

few items. They will identify a product group and concentrate on the according shelf.
We defined the types of decision behavior as referring to the customers' overall behavior in the supermarket. The question now is how to react to each type of purchase decider. Exemplary sources of the fundamentals upon which we based the following ideas are [KRW03] and [KRE04]. We chose five recommendation parameters which must be adapted to the respective decider:

The type of product. We distinguish between an absolutely new product (innovation), a substitute of a new brand, a discount offer and any cross-selling product discovered through association rule mining. Habitual buyers will be most attracted by discount offers, unknown products like innovations will be risky for them. Impulsive purchasers will be particularly open to strong stimuli which could be represented by new products or discount offers. Extended and limited deciders are generally open to all kinds of products.
The level of detail. Every purchase decision class is linked with a different ability of information reception. The latter depends on the according information need and cognitive effort. Thus, short messages are advisable for impulsive and habitual buyers, whereas limited and extensive deciders will welcome detailed information.
Recommendation variety. If the person is thinking a lot about the purchase decision, as it is the case with extensive and limited purchasers, too many different recommendations of different products will be overcharging. Habitual and impulsive buyers may get more recommendations to increase their interest.
Recommendation repetition. Extensive and limited deciders will probably get annoyed by too many repetitions due to their cognitive involvement. With habitual and impulsive purchasers, in contrast, repetitions are needed to accomplish learning effects.
Recommendation location. Sometimes it may be useful to the buyer to get advice at the location he is currently staying at. This is especially the case with limited and extensive deciders as they stand in front of a product shelf to choose out of a set of product alternatives. Recommendation variety and repetition rate, again, depend on the shop's size. The latter determines where to set a boundary dividing the range into low and high analogously to the way low and high speed is determined, for instance.

3.2 Integrating Purchase Decision Classes and Association Analysis

Using association analysis, the product shelves a customer is most likely to go to next can be calculated based on the product shelves formerly visited. To determine which product shelf is best suited, which exact product to recommend and how to present the recommendation, the purchase decision class is needed. During a shopping session, the system will work as follows: First, the customer will have to walk through the store until the system has enough data to output a recommendation. If, for example, a customer, identified as a habitual decider, is standing in front of a cheese shelf, the association analysis could reveal that the best recommendation locations are the bread shelf and the soft drinks shelf. At the bread shelf a discount offer is available, at the soft drinks shelf a product substitute and any product the sales manager wants to promote are available. Since it is a habitual buyer, the most suitable product is the discount offer, in this case the bread. The system will

adapt the presentation modus to the purchase decider's needs. Furthermore, the movement characteristics are re-checked continually. If the behavior changes in a way such that the purchase decision type also changes, the recommendation strategy has to be adapted.

4 Conclusion

We introduced a concept of a supermarket recommender system whose product suggestions are not based on statistical patterns alone. Using position tracking data, the customers' current way of decision making is deduced to select the most suitable recommendations from those calculated with association analysis. The purchase decision class also determines how to best present a recommendation.
With our concept we suggest a novel combination of two different science fields, data mining and economic psychology. Due to this novelty many new questions arise that have to be answered: Will the customers accept a system that watches their moves and outputs personalized advertisement all along their way through the shop? Is a deduction of buying behavior from movement behavior really reliable? How to deal with possible error sources like customers leaving their shopping carts, or idle times when people are stopping to have a chat with a friend they have just met in the aisle? These questions can be answered by field tests of a system prototype and by surveys. The recommendation quality can be checked with the help of common indicators like accuracy and precision [AT05]. The aim of this paper is to promote the idea of integrating causal relations and semantically interpretable information into existing data mining technologies. Unfortunately, this has been done too rarely with recommender systems. Our concept can be transferred to the Internet as well, where even more data like mouse, scroll or keyboard events are available.

References

[AT05] Gediminas Adomavicius and Alexander Tuzhilin. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17:734–749, 2005.

[CFGK05] Chad Cumby, Andrew Fano, Rayid Ghani, and Marko Krema. Building Intelligent Shopping Assistants Using Individual Consumer Models. In *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces*, pages 323–325, New York, NY, USA, 2005. ACM Press.

[HK01] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, San Franciscio, CA, 2001.

[HS69] J. Howard and J. Sheth. *The Theory of Buyer Behavior*. John Wiley & Sons, New York, NY, 1969.

[Kat51] G. Katona. *Psychological Analysis of Economic Behavior*. McGraw-Hill, New York, NY, 1951.

[KRE04] W. Kroeber-Riel and F. Esch. *Strategie und Technik der Werbung*. Kohlhammer, Stuttgart, Germany, 2004.

[KRW03] W. Kroeber-Riel and P. Weinberg. *Konsumentenverhalten*. Vahlen, Munich, Germany, 2003.

[Loe05] C. Loebbecke. Emerging Information Systems Applications in Brick-and-mortar Supermarkets: A Case Study of Content Provision Devices and RFID-Based Implementations. In *Pacific Asia Conference on Information Systems (PACIS)*, Bangkok, Thailand, July 2005.

Strategies for Context-Aware Data Distribution in Heterogeneous and Dynamic Device Ensembles

Henry Ristau[1] / Clemens H. Cap
Chair for Information and Communication Services
University of Rostock, Rostock, Germany
Email: henry.ristau / clemens.cap@uni-rostock.de

Abstract: We analyze strategies available in current research according to their ability to provide context-aware data distribution in heterogeneous and dynamic environments as they start to emerge in the areas of ubiquitous computing. Therefore we define a taxonomy to divide these strategies into distinct groups and analyze each group. We identify to be solved problems for these groups and conclude that slim algorithms without middleware that act on local information could very well provide context-aware data distribution especially in these environments.

1 Introduction

Efficient data distribution from a data provider to a number of consumers has been researched for different networked applications like peer-to-peer computing for the Internet [MM02] and data distribution in sensor networks [HKB99].

The idea of context-awareness is reflected in newer approaches for data distribution. Here data is distributed more efficiently utilizing context information as it is available in form of metadata [HKN$^+$05]. Data is therefore represented by an *entity* provided with context. The entity is only valid in that context and must be manipulated to reflect another context. The challenges of context-aware data distribution go beyond only efficiently distribution, storage and retrieval of data but also need to consider how data can be manipulated to match another context and be available for other applications.

A simple example is a sensor actuator network in a building that contains different temperature sensors, temperature displays and a fire alarm station that can trigger fire-alarms. The sensors provide temperature values as entities and the unit of measurement, location information and time stamps as context. A temperature value is only valid in the provided context and must be processed to fit another context. For example a value of degree Celsius can be processed into a value of degree Fahrenheit and reflect a new context which in turn can be understood by another temperature display. Another more complex manipulation would transform three temperature values of more than 100 degrees Celsius from the same room in a small time interval to a fire alarm for that level of the building.

The instructions that are used to process data to match another context we are going to call a *procedure* with the term *system-procedures* for the set of all available procedures in the system at a certain time point. The data, that can be manipulated by a certain procedure,

is identified by its context or parts of its context.

In the following we describe the scenario for our analysis and give an overview of related work. In the next section we introduce a novel set of requirements to characterize and evaluate context-aware data distribution aproaches followed by an analysis of currently available approaches in that field. In the last section we present conclusions and ideas for future work.

1.1 Scenario

We assume a very heterogeneous and dynamic ensemble of nodes. Nodes are added and removed and their connection points change without prior notice. Therefore no single node knows the global topology of the ensemble and the system-procedures have to change over time to adapt to changes in the ensemble. No uniform communication technology exists that allows direct communication between all nodes. But we assume that at least one communication path exists between any two nodes involving a number of forwarding nodes. There is no central node in the ensemble where procedures can be entered, managed and stored. The procedures are entered and extended in different places and must be kept in a distributed way throughout the ensemble.

A field, that approximates our assumptions very well, are ambient intelligence systems and ubiquitous computing [Wei99] with the initial example being just a tiny fraction of a complete distributed system. The purpose of such a system is to assist the users in every aspect to reduce the administrative part of their daily work. Therefore more sensors and actuators are necessary as well as the integration of stationary and mobile computers into the ensemble.

1.2 Related Work

Van Bunningen et al. [vBFA05] describe their vision of context for ubiquitous data management from a very user-centric perspective. They divide strictly between context as information describing real objects measured by sensors and metadata as data about data. They also divide between database side which delivers the data and sensor networks which deliver the context. The question of how data is enriched with context and provided to other systems or the user is not answered.

Feng et al. [FAJ04] divide in their vision of context-aware data management for ambient intelligence between user-centric and environmental context. They do not take metadata into account and do not provide any form of proactivity. Every information from the context-aware data management system is queried by the user based on his current context.

2 Context-Aware Data Distribution

For a distributed system to proactively provide the user with information and assistance, it is important that the correct data is at the correct node at the correct time. This means that every consumer receives the correct elements as efficient as possible. To provide this behaviour, we define five requirements for context-aware data distribution: *Completeness*, *regularity*, *relevance*, *efficiency* and *flexibility*. The first two are guarantees, that can either be provided by a data distribution strategy or not and the last three are values between zero and one that characterize the behaviour of a strategy.

If a data distribution algorithm can guarantee the delivery of every information that is available in the distributed system or can be processed from available information utilizing the system-procedures and that is required by a consuming node, it provides completeness. Our initial example requires completeness for the fire alarm monitor. As soon as there are enough temperature values to generate a fire alarm, that alarm must reach the monitor to be able to exactly estimate the size of a fire.

Regularity is a weaker form of completeness where only one information for a given context must be delivered. In the example, a temperature display does not need all temperature values from a room but at least one in a given time interval and thus requires regularity from the algorithm.

If a node is removed from a route from provider to all consumers, as provided by a data distribution algorithm, and the data on that route would not reach all consumers anymore, than that node is relevant. Now relevance is the average across all nodes of the number of routes a node is relevant for over the number of routes it is involved in.

To express the efficiency of a data distribution algorithm, a link-metric [DPZ04] can be used that mirrors the usage of resources along a chosen path. The sum of weights of all links in a route can be put into ratio with that smallest sum possible for a valid route. Doing this for all routes in average, we gain an efficiency value for that algorithm. The flexibility of such an algorithm can be expressed by the sum of the weights on all routes for data transmission in ratio to the sum of all links used by the algorithm for communication. An algorithm that needs less communication to provide routes has a higher flexibility.

2.1 Context-Aware Data Distribution Strategies

Table 1 shows a simple two-dimensional taxonomy to divide data distribution approaches into six abstract classes. One dimension, the rows in the table, differs between the two possible data flow approaches *push* (data transmission without prior request) and *pull* (data transmission upon request). The second dimension, the columns, maps the *number of communication steps* involved in the procedure of sending data, direct data transmission (one step), registration at and data transmission through a middleware (two steps) as well as registration and address lookup at a middleware followed by direct data transmission (three steps).

Table 1: Two dimensional taxonomy to classify different data distribution strategies. The $\rightarrow$ represents the flow of information with S being the information Source, D the Destination and M a Middleware acting as intermediary.

	direct approaches	**middleware based approaches**		**description**
	one step	**two steps**	**three steps**	
push	"Natural"	"Register"	"Destination Lookup"	class name
		$D \rightarrow M$	$D \rightarrow M$	registration step
			$S \rightarrow M \rightarrow S$	lookup step
	$S \rightarrow D$	$S \rightarrow M \rightarrow D$	$S \rightarrow D$	data transmission step
pull	"Request"	"Query"	"Source Lookup"	class name
		$S \rightarrow M$	$S \rightarrow M$	registration step
			$D \rightarrow M \rightarrow D$	lookup step
	$D \rightarrow S \rightarrow D$	$D \rightarrow M \rightarrow D$	$D \rightarrow S \rightarrow D$	data transmission step

Representatives of the "natural" approach are flooding algorithms [ZF06]. They provide a very high flexibility acting on local information only and require no additional communication. They allow regularity and completeness if desired. The disadvantage of flooding algorithms is a very low relevance and therefore a low efficiency because all data is sent to every node independent of whether it is needed there or not.

In sensor databases [LZZ06] the "request" approach is utilized. The relevance is higher because only requests are flooded and there are less requests than replies in sensor networks. Through publish/subscribe methods the number of request can be further reduced with the risk to loose data if the systems topology changes. Data aggregation on the flow is also provided for such algorithms [SS04]. However, distributed context-awareness is not possible through this approach because the consumer must request all data and procedures to generate the information it needs and therefore needs to know all available system-procedures.

The "register" class of approaches is represented by publish/subscribe systems in current research which provide distributed registries [YZH07] and context-awareness [FR07]. These approaches provide a much higher relevance and efficiency because no flooding is necessary anymore. However they rely on a structured middleware which needs to be consistent at all times and therefore lowers the flexibility significantly.

The same goes for "query" approaches like tuple-spaces which are also provided in a distributed form [JXJY06] and with context-awareness [PBC05]. They additionally rely on polling and therefore are less efficient and provide no proactivity.

The "lookup" approaches are represented by most modern peer-to-peer systems [MM02] because the principle is very efficient in the distribution of large entities with very few context information like files in a homogeneous network environment like the Internet. Both are not provided by our scenario and therefore "lookup" approaches are very inefficient here and suffer from the same problem as the "request" approaches because context-awareness can not be implemented in a middleware that does only store the context but not the entities.

Table 2 summarizes the analysis of all classes and presents the requirements for our scenario in the last column. The most important requirements are flexibility and the ability to process data according to context in communication. The efficiency of the data distri-

Table 2: Comparative evaluation of data distribution approaches: + + perfectly possible; + possible but not provided in all heterogeneous and dynamic environments; - provided only in special environments; - - not provided at all; NYA - not yet available; NDR - no distributed realization possible.

Requirement	"Natural"	"Request"	"Register"	"Query"	"Lookup"	Scenario Requirements
Completeness	+ +	+	+	+	+	+
Regularity	+ +	+	+	+	+	+
Relevance	$\ll 1$	< 1	~ 1	~ 1	~ 1	< 1
Efficiency	$\ll 1$	< 1	< 1	$\ll 1$	$\ll 1$	< 1
Flexibility	~ 1	< 1	$\ll 1$	$\ll 1$	$\ll 1$	~ 1
Context-awareness	NYA	NDR	+ +	+ +	NDR	+ +

bution strategies and therefore the relevance of received information elements should be considerably high to spare resources. The system should be able to guaranty regularity and completeness for situations where it is needed.

3 Conclusion

Concluding from the analyzed requirements a very good approach currently available to realize context-aware data distribution in our scenario lies in the "register" class of strategies. However the overhead generated by keeping a middleware consistent prevent such a strategy from fulfilling the high requirements in flexibility. We therefore target to improve relevance and efficiency of approaches from the "natural" class by utilizing context-information. We also target to provide these strategies with means of in communication data processing to create slender algorithms for context-aware data distribution based on local information in heterogeneous and dynamic device ensembles.

References

[DPZ04] Richard Draves, Jitendra Padhye, and Brian Zill. Comparison of routing metrics for static multi-hop wireless networks. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 133–144, New York, NY, USA, 2004. ACM Press.

[FAJ04] Ling Feng, Peter M.G. Apers, and Willem Jonker. Towards Context-Aware Data Management for Ambient Intelligence. In *Proceedings of the 15th International Conference on Database and Expert Systems Applications (DEXA)*, volume 3180/2004, pages 422–431, 2004.

[FR07] Davide Frey and Gruia-Catalin Roman. Context-Aware Publish Subscribe in Mobile ad Hoc Networks. In *Proceedings of the 9th International Conference on Coordination Models and Languages*, 2007.

[HKB99] Wendi Rabiner Heinzelman, Joanna Kulik, and Hari Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 174–185, New York, NY, USA, 1999. ACM Press.

[HKN+05] N. Honle, U.-P. Kappeler, D. Nicklas, T. Schwarz, and M. Grossmann. Benefits of Integrating Meta Data into a Context Model. In *Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on*, pages 25–29, 8-12 March 2005.

[JXJY06] Yi Jiang, Guangtao Xue, Zhaoqing Jia, and Jinyuan You. DTuples: A Distributed Hash Table based Tuple Space Service for Distributed Coordination. In *Grid and Cooperative Computing, 2006. GCC 2006. Fifth International Conference*, pages 101–106, Oct. 2006.

[LZZ06] Guohua Liu, Shuzhi Zhang, and Dongming Zhang. Research of the Query Technology in Wide Area Sensor Databases. In *Computational Intelligence for Modelling, Control and Automation, 2006 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on*, pages 128–128, Nov. 2006.

[MM02] Petar Maymounkov and David Mazires. Kademlia: A Peer-to-Peer Information System Based on the XOR Metric. In *Peer-to-Peer Systems: First InternationalWorkshop, IPTPS 2002 Cambridge, MA, USA, March 7-8, 2002. Revised Papers*, 2002.

[PBC05] Gian Pietro Picco, Davide Balzarotti, and Paolo Costa. LighTS: a lightweight, customizable tuple space supporting context-aware applications. In *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*, pages 413–419, New York, NY, USA, 2005. ACM Press.

[SS04] Mehdi Sharifzadeh and Cyrus Shahabi. Supporting spatial aggregation in sensor network databases. In *GIS '04: Proceedings of the 12th annual ACM international workshop on Geographic information systems*, pages 166–175, New York, NY, USA, 2004. ACM Press.

[vBFA05] A.H. van Bunningen, L. Feng, and P.M.G. Apers. Context for ubiquitous data management. In *Ubiquitous Data Management, 2005. UDM 2005. International Workshop on*, pages 17–24, 4 April 2005.

[Wei99] Mark Weiser. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(3):3–11, 1999.

[YZH07] Xiaoyu Yang, Yingwu Zhu, and Yiming Hu. Scalable Content-Based Publish/Subscribe Services over Structured Peer-to-Peer Networks. In *Parallel, Distributed and Network-Based Processing, 2007. PDP '07. 15th EUROMICRO International Conference on*, pages 171–178, 7-9 Feb. 2007.

[ZF06] Y. Zhang and M. Fromherz. Constrained flooding: a robust and efficient routing framework for wireless sensor networks. In *Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference on*, volume 1, page 6pp., 18-20 April 2006.

Performance study of an Interleave Division Multiple Access Scheme

Sebastian Vorköper
University of Rostock
Institute of Communications Engineering
Email: sebastian.vorkoeper@uni-rostock.de

1 Abstract

Within the recent years, IDMA, a special form of CDMA, has received considerable attention as a promising approach for next generation wireless systems. In IDMA each user is assigned a unique interleaver in order to separate the different users. Unfortunately, their separation requires multiuser detection (MUD) techniques, which were so far viewed as a costly option. However, with the progress in iterative processing techniques, low-complexity MUD techniques have attracted a lot of attention in IDMA receiver design. In this contribution different IDMA system concepts are studied and compared in terms of performance and complexity.The performance investigations are carried out by both computer simulations as well as EXIT charts.

2 Introduction

The requirements for transmission capacity for speech, data and multimedia information will probably increase continuously in the future. With the limitation of available resources such as transmit power or bandwidth, the demand to increase the spectral efficiency of future transmission systems is clearly recognizable. Within the last years a new possibility to increase the capacity and quality of wireless transmission has been highlighted: Systems with multiple transmit and receive antennas have been developed and form multiple-input multiple-output systems (MIMO) (HTW03). They can be seen as a promising approach to increase both the achievable capacity and integrity of wireless systems. However, nearly at the same time, another technique called IDMA has attracted a lot of attention. Here, the separation of the different users is done by unique interleavers, which require cost-intensive multiuser detection techniques at the receiver side. This makes IDMA a considerable candidate for the Uplink. In order to reduce the resource requirements at the basestation, different low-complexity MUD solutions have been introduced in (EBBS98) and (PLWL03).

IDMA inherits many advantages from CDMA (PLWL03) such as diversity against fading and mitigation of the worst-case other-cell user interference problem. Some analysis of IDMA with different detectors and decoders have been done in (LP06) and (VKMP04). A comparison between IDMA and other multiple access schemes in terms of the bit-error

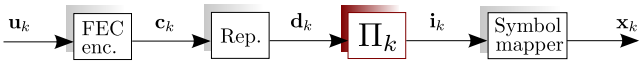

Figure 1: IDMA transmitter structure.

rate were done in (KB05).

The main idea of IDMA is to separate every layer by interleaving the spreaded coded information sequence with a unique interleaver. So, it is possible to transmit the different layers at the same time in the same frequency and separate them on the receiver side.

IDMA is strongly related to DS-CDMA. It was shown in (KB05) that with increasing number of iterations IDMA performs better in approaching the single user bound in AWGN channels. It even outperforms DS-CDMA in multipath channels and in near far scenarios in the first iteration.

The disadvantage of IDMA is the higher numerical calculation effort on the receiver side. This can be disregarded if the receiver possess enough capacity, i.e. a basestation for example. Thus IDMA should be preferred over DS-CDMA in mobile communication environments especially in the uplink area.

The considered system model is introduced in chapter 3. This includes the IDMA-transmitter and receiver as well as the channel assumptions. Furthermore, the different receiver modifications are investigated in order to achieve a better performance and/or stability in the iterative interference cancellation process. Chapter 4 gives a short introduction in the EXIT-chart analysis, which is used for the performance study and the results in chapter 5.

3 System model

The structure of the considered system model is depicted in Fig. 1. The layer-specific binary information sequence $\mathbf{u}_k \, \epsilon\{0,1\}^{N_u \times 1}$ (with $k = \{1, \cdots, K\}$) of the length N_u is encoded with a non-systematic, non-recursive convolutional encoder of the rate R_c^{cc}, resulting in the coded information sequence $\mathbf{c}_k \, \epsilon\{0,1\}^{N_c \times 1}$. This sequence will then be spread using a repetition code of rate R_c^{rc} resulting in the sequence $\mathbf{d}_k \, \epsilon\{0,1\}^{N_d \times 1}$. Finally, this sequence will be interleaved using a pseudo random interleaver and results in the sequence $\mathbf{i}_k \, \epsilon\{0,1\}^{N_d \times 1}$. Each layer k has its unique interleaver Π_k in order to separate the different users, whereby the layer-specific interleavers are assumed to be known at the receiver side. After BPSK symbol mapping, the layer-specific sequence $\mathbf{i}_k \, \epsilon\{0,1\}^{N_d \times 1}$ is mapped onto the sequence $\mathbf{x}_k \, \epsilon\{+1,-1\}^{N_d \times 1}$. These layer-specific sequences are transmitted synchronously over an AWGN channel.

The transmitted layer-specific sequences $\mathbf{x}_k \, \epsilon\{+1,-1\}^{N_d \times 1}$ result in the received signal $\mathbf{y}$, which can be composed of the layer-specific parts as follow

$$\mathbf{y} = \sum_{k=1}^{K} \mathbf{x}_k + \mathbf{n} \ . \tag{1}$$

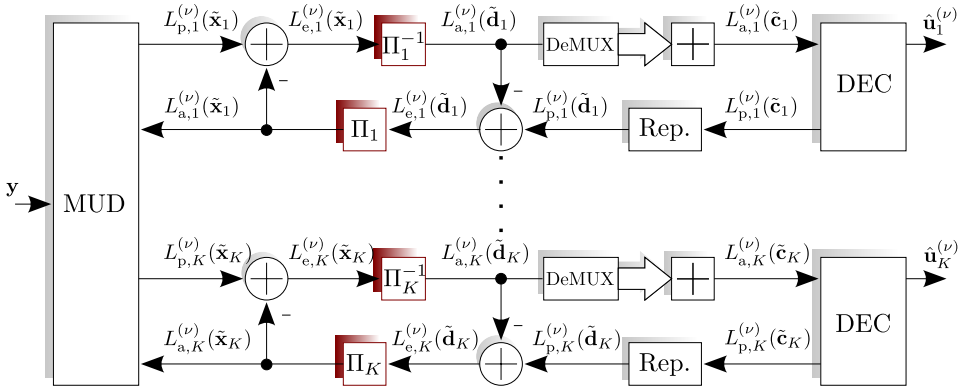

Figure 2: IDMA receiver structure

Based on the non-orthogonal signatures, multi-layer interference is still feasible. Different receivers are known from the literature (e. g. successive (SIC) and parallel (PIC) interference cancellation) (EBBS98), (Hag96) et al.. Applying a turbo-detection scheme, consisting of a SIC or PIC unit, seems to be a popular solution in order to separate the different layer-specific signals. Within this process, soft-information, e. g. log likelihood ratios (LLR's) are exchanged between the multiuser detector (MUD) and the decoder (DEC).

The multiuser detector calculates the layer-specific a-posteriori LLR sequence $L^{(\nu)}_{\mathrm{p},k}(\tilde{\mathbf{x}}_k)$ for each symbol, starting at the iteration $\nu = 1$. Subtracting the a-priori LLR's $L^{(\nu)}_{\mathrm{a},k}(\tilde{\mathbf{x}}_k)$ from $L^{(\nu)}_{\mathrm{p},k}(\tilde{\mathbf{x}}_k)$ and deinterleaving the obtained extrinisic log likelihood ratios $L^{(\nu)}_{\mathrm{e},k}(\tilde{\mathbf{x}}_k)$ will result in the a-priori LLR's $L^{(\nu)}_{\mathrm{a},k}(\tilde{\mathbf{d}}_k)$ for the demultiplexer. After demultiplexing and summing up the LLR's over the repeated codesymbols, the decoder will generate hard-decisions $\hat{\mathbf{u}}^{(\nu)}_K$ of the transmitted information sequence $\mathbf{u}_k$ as well as soft-information $L^{(\nu)}_{\mathrm{p},k}(\tilde{\mathbf{c}}_k)$, which will be fed back after repitition and interleaving as a-priori LLR's $L^{(\nu)}_{\mathrm{a},k}(\tilde{\mathbf{x}}_k)$ for the MUD. Thus, this information can be used to improve the layer-specific a-posteriori LLR in the further detection process. The described algorithm can be applied in a successive or parallel interference cancellation unit.

Applying a SIC, all layers are processed iteratively. At the iteration ν, the layer k has knowledge of all other layers from iteration $\nu - 1$ and from all completed layers at the current time instant. After processing the last layer K, the algorithm starts again with the first layer.

In contrast to the SIC, using a PIC, layer k at iteration ν has only knowledge of all layers from the previous $\nu - 1$ iteration. So, every layer at iteration ν is processed on its own. Both algorithms can be repeated as long as a certain stopping criteria is reached.

In this paper, three different receiver realizations were investigated.

The first receiver structure is depicted in Fig. 2, whereas the layer-specific a-priori infor-

mation can be calculated according to:

$$L_{\mathrm{e},k}^{(\nu)}(\tilde{\mathbf{d}}_k) = L_{\mathrm{p},k}^{(\nu)}(\tilde{\mathbf{d}}_k) - L_{\mathrm{a},k}^{(\nu)}\tilde{\mathbf{d}}_k) \ . \tag{2}$$

Here, the extrinsic LLR's $L_{\mathrm{e},k}^{(\nu)}(\tilde{\mathbf{d}}_k)$ are formed taking the symbols from the repeated code symbols into account, instead of using the coded symbols. Since the signal-to-noise ratio (SNR) after reversing the repetition of the coded symbols is higher then before, only a fraction of the a-priori LLR value's is subtracted in (2).

For the second receiver setup, the whole a-posteriori LLR's from the decoder are feed back as a-priori information for the detector resulting in $L_{\mathrm{e},k}^{(\nu)}(\tilde{\mathbf{d}}_k) = L_{\mathrm{p},k}^{(\nu)}(\tilde{\mathbf{d}}_k)$. So, the MUD will not only receive the extrinsic, but also the complete a-priori LLR's from the decoder.

In contrast to receiver one and two, the third receiver passes only extrinsic LLR's from the decoder to the detector with $L_{\mathrm{e},k}^{(\nu)}(\tilde{\mathbf{c}}_k) = L_{\mathrm{p},k}^{(\nu)}(\tilde{\mathbf{c}}_k) - L_{\mathrm{a},k}^{(\nu)}(\tilde{\mathbf{c}}_k)$.

4 EXIT-Chart analysis

Analyzing serially concatenated codes with the bit error rate (BER) function, as applied in the IDMA interference cancellation scheme, is a time consuming task. This is due to the calculation complexity of the BER and the different behaviors of each detection and decoder combinations. EXtrinsic Information Transfer (EXIT) charts by S. ten Brink in (tB01) can give a good behavior estimation with the ability of subsequently calculate the BER.

The basic idea behind EXIT-charts is the exchange of mutual information between the components of a concatenated system. Therefore, the mutual information of the inner and outer decoder have to be calculated.

This is done by following the approach of Stephan ten Brink in modelling the mutual information of the a-priori LLR's at the decoder input and measuring the mutual information of the extrinsic LLR's at the decoder output. To do so, it is assumed that for large interleavers the a-priori values remain fairly uncorrelated and the probability density functions of the extrinsic output values approach Gaussian-like distribution with increasing number of iterations (tB01).

An EXIT chart is now obtained by plotting the transfer characteristics for both the detector and the decoder within a single diagram, where the axes have to be swapped for one of the constituent decoders (tB01) (normally the outer one for serial concatenation).

It is clearly seen in Fig. 2, that the extrinsic mutual information $I_{\mathrm{e},k}\left(\mathbf{x}_k; L_{\mathrm{e},k}(\tilde{\mathbf{x}}_k)\right)$ of layer k doesn't only depend from the channel SNR, the system load $\beta = K \cdot R_c^{rc}$ and its own a-priori information $I_{\mathrm{a},k}\left(\mathbf{x}_k; L_{\mathrm{a},k}(\tilde{\mathbf{x}}_k)\right)$, but also from the a-priori information of all

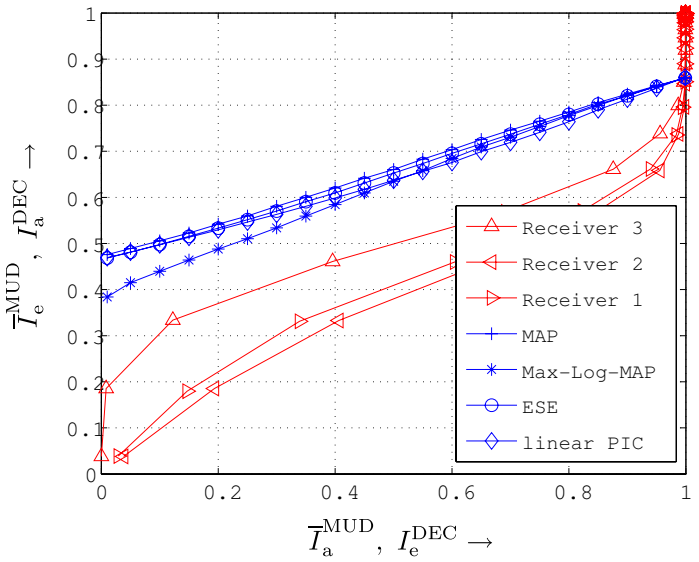

Figure 3: EXIT-chart with different multiuser detection algorithms

other layers. For simplicity, we will refer to the different mutual information as follow

$$I_{\mathrm{a},k}^{\mathrm{MUD}} = I_{\mathrm{a},k}^{\mathrm{MUD}}\left(\mathbf{x}_k; L_{\mathrm{a},k}(\tilde{\mathbf{x}}_k)\right) \tag{3a}$$

$$I_{\mathrm{e},k}^{\mathrm{MUD}} = I_{\mathrm{e},k}^{\mathrm{MUD}}\left(\mathbf{x}_k; L_{\mathrm{e},k}(\tilde{\mathbf{x}}_k)\right) \tag{3b}$$

$$I_{\mathrm{a},k}^{\mathrm{DEC}} = I_{\mathrm{e},k}^{\mathrm{MUD}} \tag{3c}$$

$$I_{\mathrm{e},k}^{\mathrm{DEC}} = I_{\mathrm{a},k}^{\mathrm{MUD}} \quad . \tag{3d}$$

Thus, the extrinsic mutual information of layer k, at the multiuser detection, is a function of

$$I_{\mathrm{e},k}^{\mathrm{MUD}} = f\left(\mathrm{SNR}, \beta, I_{\mathrm{a},1}^{\mathrm{MUD}}, \ldots, I_{\mathrm{a},K}^{\mathrm{MUD}}\right) \quad . \tag{4}$$

This will inevitably lead to a multidimensional EXIT-chart (NLW$^+$07), (TNH06).

For PIC, all layers have nearly identical a-priori and extrinsic mutual information, which makes it possible to utilize the mean mutual information $\overline{I}_{\mathrm{a}}^{\mathrm{MUD}}$ and $\overline{I}_{\mathrm{e}}^{\mathrm{MUD}}$ for the EXIT-charts.

5 Results

The EXIT-chart in Fig. 3 shows the mean transfer charts of the different multiuser detectors together with the different transfer charts of a half rate decoder with constraint length $L_c = 3$, generator polynomials in octal notation $g_1 = 5_8$ and $g_2 = 7_8$ and varied a-posteriori

handling (different receiver implementations). The input and output of the decoder transfer charts were swaped due to (3c) and (3d). It is seen that all suboptimal multiuser detection implementations, except the Max-Log-MAP algorithm, will almost perform as good as the optimal MAP criteria if $\beta \leq 1$. The ESE as investigated by PING, (PLWL03) gives a slightly better performance compared with the linear PIC described in (Kuh04). The Max-Log-MAP multiuser detector will deliver less extrinsic information in the lower a-priori information area assuming a low system load, e. g. $\beta \leq 1$. With increasing a-priori information it will improve and outperform other suboptimal MUD. With increasing system load, e. g. $\beta > 1$, the performance of ESE and linear PIC decreased whereas the Max-Log-MAP MUD shows a neglectable performance degradation (Vor07).

All three transfer charts of the implemented receivers were plotted in Fig. 3 together with the different MUD algorithms. It is clearly seen, that receiver two is supposed to outperform the other implementations by the numbers of iterations for achieving the maximal possible mutual information. However, measurements in (Vor07) have shown that this modification is highly instable and unpredictable at a certain system load and SNR. This is due to the fact, that a wrongly decoded information sequence is feed back with high LLR values to the multiuser detector. Now, the MUD will return a wrong estimation of the transmitted sequence in the next iteration or for the following layers. This will lead to an overall wrong detection and decoding of the transmitted information sequences for every layer. The third receiver delivers the best IDMA receiver strategy concerning the stability. However the realization results in a poor performance since more iterations are needed to achieve the maximal possible mutual information per layer and therewith the maximal possible layer separation. Analyzing the convergence criteria of this receiver realization, the information exchange between the detection and decoder is stopped early. This is due to the fact that, in conjunction with a low SNR and high system load, the MUD transfer chart will intersect the DEC transfer chart from receiver three sooner than for the other implementations.

A good tradeoff between stability and performance delivers the first receiver. The detection and decoding scheme is quite as stable as for receiver three and the performance is comparable with receiver two (Vor07).

6 Conclusion

It was shown that the EXIT-chart analysis is a good tool to predict the iterative interference cancellation process within the IDMA detection and decoding scheme. This applies especially for a PIC with a small system load and a high SNR. Alternating the receiver, by the way the a-posteriori LLR's from the decoder is handled, leads to a stabilization of the system. This associates with the cost of performance loss by the numbers of iterations. For this reason, the receiver introduced by Kusume in (KB05), represents a good tradeoff between detection and decoding performance and stabilization.

References

[EBBS98] H. Elders-Boll, A. Busboom, and H.D. Schotten. Implementation of linear multiuser detectors for asynchronous CDMA systems by linear interference cancellation algorithms. In *Vehicular Technology Conference, 1998. VTC 98. 48th IEEE*, volume 3, pages 1849–1853vol.3, 18-21 May 1998.

[Hag96] J. Hagenauer. Forward error correcting for CDMA systems. In *Spread Spectrum Techniques and Applications Proceedings, 1996., IEEE 4th International Symposium on*, volume 2, pages 566–569vol.2, 22-25 Sept. 1996.

[HTW03] A. Hottinen, O. Trikkonen, and R. Wichman. *Multi-antenna Tranceiver Techniques for 3G and Beyond*. Wiley, 2003.

[KB05] K. Kusume and G. Bauch. CDMA and IDMA: Iterative Multiuser Detections for Near-Far Asynchronous Communications. In *Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on*, volume 1, pages 426–431, 11-14 Sept. 2005.

[Kuh04] V. Kuhn. Analysis of iterative multi-user detection schemes with EXIT charts. In *Spread Spectrum Techniques and Applications, 2004 IEEE Eighth International Symposium on*, pages 535–539, 30 Aug.-2 Sept. 2004.

[LP06] Lihai Liu and Li Ping. A Comparative Study on Low-Cost Multiuser Detectors. In *Communications, 2006 IEEE International Conference on*, volume 11, pages 4947–4952, June 2006.

[NLW+07] S. X. Ng, W. Liu, J. Wang, M. Tao, L.-L. Yang, and L. Hanzo. Performance Analysis of Iteratively Decoded Variable-Length Space-Time Coded Modulation. In *Communications, 2007. ICC '07. IEEE International Conference on*, pages 5921–5926, 24-28 June 2007.

[PLWL03] Li Ping, Lihai Liu, K. Y. Wu, and W. K. Leung. Interleave division multiple access (IDMA) communication systems. In *Proc. 3rd International Symposium on Turbo Codes & Related Topics*, pages 173–180, 2003.

[tB01] S. ten Brink. Convergence behavior of iteratively decoded parallel concatenated codes. *Communications, IEEE Transactions on*, 49(10):1727–1737, Oct. 2001.

[TNH06] R.Y.S. Tee, S.X. Ng, and L. Hanzo. Three-Dimensional EXIT Chart Analysis of Iterative Detection Aided Coded Modulation Schemes. 2006.

[VKMP04] N. Varnica, A. Kavcic, X. Ma, and L. Ping. Density Evolution and LDPC Code Optimization for Interleaver Division Multiple Access. In *Global Mobile Congress, Shanghai, China*, 2004.

[Vor07] Sebastian Vorköper. Entwicklung und Analyse eines auf dem Zugriffsverfahren "Interleave Division Multiple Access" basierenden Übertragungssystems. Master's thesis, University of Rostock, Germany, 2007.

Resource Allocation for Distributed MIMO Multi-hop Wireless Networks

Yidong Lang, Carsten Bockelmann,
Dirk Wübben, and Karl-Dirk Kammeyer
Department of Communications Engineering
University of Bremen, Germany
Email: {lang, bockelmann, wuebben, kammeyer}@ant.uni-bremen.de

Armin Dekorsy
Bell Labs Europe
Alcatel-Lucent AG
Nuremberg, Germany
Email: dekorsy@alcatel-lucent.com

Abstract: Distributed MIMO technology has gained significant attention in industry and academia recently, due to its ability to increase capacity drastically and its inherent attribute of scalability for wireless mesh networks. In this paper we briefly overview the concept of distributed MIMO and investigate the end-to-end ergodic channel capacity of a distributed MIMO multi-hop network. By formulating the resource allocation problem as a concave optimization problem, we are able to obtain the solution of optimal power and bandwidth allocation in a very efficient way.

1 Introduction

In this paper an end-to-end scenario in a wireless multi-hop network is considered, where a source communicates with the destination via a number of relays. In order to avoid interference between the relaying hops, orthogonal access schemes like frequency-division multiple access (FDMA) or time-division multiple access (TDMA) are usually used. However, it can be shown that both access schemes achieve the same capacities [2], so that only FDMA will be considered for simplicity. At each relaying node the decode-and-forward relaying protocol is applied, where the data will be first detected and decoded completely, then re-encoded and transmitted to the next relaying nodes [3]. Recently, it was shown that the channel capacity of a wireless mesh network can drastically be increased by applying MIMO techniques with respect to spatially separated relaying nodes [1]. To this end, several relays are used to form a virtual antenna array (VAA). The end-to-end connection is therefore accomplished through a number of topologically imposed VAAs.

Since the data will be transmitted to the destination through a number of hops, an optimal resource allocation strategy should assign fractional power and bandwidth to each hop such that the end-to-end capacity is maximized. In this paper the end-to-end ergodic capacity for a distributed MIMO multi-hop network will be studied. With respect to an approximated expression of the ergodic capacity, we will derive the optimal resource (power and bandwidth) allocation strategy for a given distributed MIMO multi-hop network. This

strategy is shown to be of low complexity and to achieve near-maximum end-to-end ergodic capacity.

The remainder of the paper is organized as follows. In Section 2 the concept of distributed MIMO scheme is briefly overviewed. A concave optimization problem for maximizing the end-to-end capacity is formulated in Section 3. Some results are presented in Section 4. Finally, conclusions are given in Section 5.

2 Distributed MIMO Multi-hop Networks

A system model of a distributed MIMO multi-hop network is depicted in Figure 1, where a source node communicates with a destination node via a number of relaying nodes. Some spatially separated relaying nodes are formed into virtual antenna arrays (VAAs), which allows to increase capacity by applying space-time processing techniques, e.g space-time block codes [1]. For the further investigation a fixed network topology is assumed, i.e. the task of combining nodes to a VAA is not considered. As the data is transmitted from the source node through a number of VAAs to the destination node, such a network is referred to as a distributed MIMO multi-hop network. Note that there is no receive cooperation but only transmit cooperation between the relaying nodes of one VAA. In other words, each node in kth VAA receives signals transmitted by the nodes in the $(k-1)$th VAA, where the signals are space-time encoded cooperatively. Thus, the transmission can be modeled as a multiple-input single output (MISO) scheme. Note that the kth VAA serves as receive antenna array at the kth hop while as transmit antenna array at the $(k+1)$th hop.

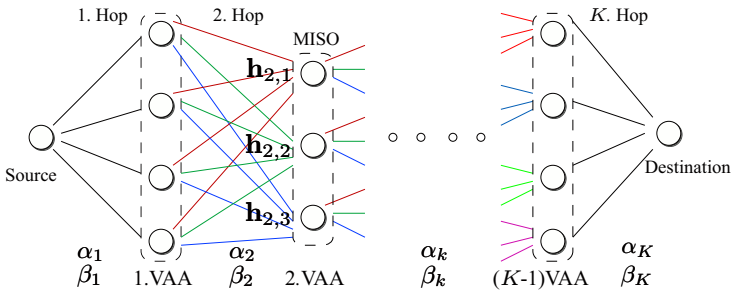

Figure 1: System model of distributed MIMO multi-hop networks

We summarize the encoding, relaying and decoding process for a given distributed MIMO network topology shortly as follows,

- **Source node**: Broadcasts the data to the nodes of the first VAA with bandwidth fraction α_1 and power fraction β_1.
- **Relaying nodes at the kth hop**: The data is decoded at each node at the kth VAA and re-encoded according to a given space-time code of length T with bandwidth

fraction α_k (FDMA) and power fraction β_k. All transmit nodes of one VAA use same bandwidth and transmission power.

- **Destination node**: Finally, the data is space-time decoded.

To produce a mathematical representation of the distributed MIMO multi-hop system, let k index the hop, t_k, r_k denote the number of the transmit nodes and the receive nodes within the kth hop, respectively. Let $\mathbf{X}_k \in \mathbb{C}^{t_k \times T}$ denote the space-time encoded signal matrix from the t_k nodes in the kth hop, then the received signal at the jth node $\mathbf{y}_{k,j} \in \mathbb{C}^{1\times T}$ can be represented by the equation

$$\mathbf{y}_{k,j} = \sqrt{\frac{\gamma_k \beta_k P}{t_k}} \mathbf{h}_{k,j} \mathbf{X}_k + \mathbf{n}_{k,j}, \tag{1}$$

where $\mathbf{n}_{k,j} \sim \mathcal{N}_C(0,\, N_0) \in \mathbb{C}^{1\times T}$ is the Gaussian noise vector, P is the total power available for the network and N_0 is the power spectral density of the noise. The complex channel realization from the transmit nodes to the jth receive node within the kth hop is denoted as $\mathbf{h}_{k,j} \in \mathbb{C}^{1\times t_k}$. The elements of $\mathbf{h}_{k,j}$ obey the same uncorrelated Rayleigh fading statistics, i.e. complex zero-mean circular symmetric Gaussian distribution with variance 1. The pathloss at the kth hop is given by $\gamma_k = (\frac{1}{d_k})^\epsilon$, where d_k is the distance between the transmit nodes and the receive nodes at the kth hop and ϵ denotes the pathloss exponent within range of 2 to 5 for most wireless channels.

According to the relaying process discussed above, the optimization problem to maximize the end-to-end ergodic capacity C_{e2e} results in finding the optimal bandwidth fraction $\boldsymbol{\alpha}^* = [\alpha_1^*, \ldots, \alpha_K^*]^T$ and power fraction $\boldsymbol{\beta}^* = [\beta_1^*, \ldots, \beta_K^*]^T$ where $\alpha_k^*,\, \beta_k^* \in [0,\, 1], k = 1, \ldots, K$ that satisfy

$$(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) = \arg\max_{\boldsymbol{\alpha},\, \boldsymbol{\beta}} C_{e2e}(\boldsymbol{\alpha},\, \boldsymbol{\beta}). \tag{2}$$

Note that the Shannon capacity forms an upper bound and is therefore a useful measurement of the performance of the distributed MIMO multi-hop system.

3 Maximization of End-to-end Channel Capacity

The ergodic capacity of a MIMO channel was elegantly derived by Telatar [4]. The Shannon capacity of a MISO system according to (1) can be expressed as

$$C_{k,j} = \alpha_k W \, \mathrm{E}_{\mathbf{h}_{k,j}} \left\{ \log_2 \left(1 + \mathbf{h}_{k,j} \mathbf{h}_{k,j}^H \frac{\beta_k P \gamma_k}{\alpha_k t_k W N_0} \right) \right\}, \tag{3}$$

where W denotes the total bandwidth of the system. The ergodic capacity of the kth hop is dictated by the worst MISO channel $C_k = \min_j (C_{k,1}, \ldots, C_{k,j}, \ldots, C_{k,r_k})$. It is assumed that the relaying nodes belonging to the same VAA are spatially sufficiently close as to justify a common pathloss γ_k. Hence, each MISO system within the same hop has the same ergodic capacity, so that $C_k = C_{k,j},\, \forall\, j$.

Using $\log_2(1+x) \approx \sqrt{x}$ [5], the MISO channel capacity (3) can be approximated by

$$C_{k,j} \approx \sqrt{\frac{\beta_k P \alpha_k W \gamma_k}{t_k N_0}} \mathrm{E}_{\mathbf{h}_{k,j}} \left\{ \sqrt{\mathbf{h}_{k,j} \mathbf{h}_{k,j}^H} \right\} = \sqrt{\frac{\beta_k P \alpha_k W \gamma_k}{t_k N_0}} \frac{\Gamma(t_k + 1/2)}{\Gamma(t_k)} \tag{4}$$

where $\mathbf{h}_{k,j}\mathbf{h}_{k,j}^H$ is a gamma distributed random variable with $2t_k$ degrees of freedom. It is well-known that $\mathrm{E}_{\mathbf{h}_{k,j}} \left\{ \sqrt{\mathbf{h}_{k,j} \mathbf{h}_{k,j}^H} \right\} = \frac{\Gamma(t_k+1/2)}{\Gamma(t_k)}$ holds [5], where $\Gamma(\cdot)$ denotes the complete Gamma function. We now check the concavity of (4) in the joint arguments, the power fraction β_k and bandwidth fraction α_k. For simplicity we describe (4) as

$$C_k = \sqrt{\frac{\beta_k P \alpha_k W \gamma_k}{t_k N_0}} \frac{\Gamma(t_k + 1/2)}{\Gamma(t_k)} = \sqrt{\alpha_k \beta_k} \cdot A \tag{5}$$

where $A = \sqrt{\frac{PW\gamma_k}{t_k N_0}} \frac{\Gamma(t_k+1/2)}{\Gamma(t_k)}$. So that, the first-order partial derivatives, second-order partial derivatives and second-order mixed derivatives of C_k with respect to α_k, β_k are given as follows

$$\begin{aligned}
\frac{\partial C_k}{\partial \alpha_k} &= \frac{A}{2} \sqrt{\frac{\beta_k}{\alpha_k}} \\
\frac{\partial C_k}{\partial \beta_k} &= \frac{A}{2} \sqrt{\frac{\alpha_k}{\beta_k}} \\
\frac{\partial^2 C_k}{\partial \alpha_k^2} &= -\frac{A}{4} \frac{\sqrt{\beta_k}}{{\alpha_k}^{3/2}} \\
\frac{\partial^2 C_k}{\partial \beta_k^2} &= -\frac{A}{4} \frac{\sqrt{\alpha_k}}{{\beta_k}^{3/2}} \\
\frac{\partial^2 C_k}{\partial \alpha_k \partial \beta_k} &= \frac{\partial^2 C_k}{\partial \beta_k \partial \alpha_k} = \frac{A}{4\sqrt{\alpha_k \beta_k}}
\end{aligned} \tag{6}$$

To show the concavity of the C_k, we note that (for $\alpha_k > 0,\ \beta_k > 0$) the Hessian matrix is

$$\begin{aligned}
\nabla^2 C_k(\alpha_k,\, \beta_k) &= \begin{bmatrix} -\frac{A}{4} \frac{\sqrt{\beta_k}}{{\alpha_k}^{3/2}} & \frac{A}{4\sqrt{\alpha_k \beta_k}} \\ \frac{A}{4\sqrt{\alpha_k \beta_k}} & -\frac{A}{4} \frac{\sqrt{\alpha_k}}{{\beta_k}^{3/2}} \end{bmatrix} \\
&= -\frac{A}{4\alpha_k^{3/2} \beta_k^{3/2}} \begin{bmatrix} \beta_k^2 & -\alpha_k \beta_k \\ -\alpha_k \beta_k & \alpha_k^2 \end{bmatrix} \\
&= -\frac{A}{4\alpha_k^{3/2} \beta_k^{3/2}} \begin{bmatrix} \beta_k \\ -\alpha_k \end{bmatrix} \begin{bmatrix} \beta_k \\ -\alpha_k \end{bmatrix}^T \preceq 0
\end{aligned} \tag{7}$$

hence, C_k is proven to be jointly concave in the power faction β_k and band fraction α_k.

Due to decode-and-forward relaying protocol, the destination node can decode the signals correctly if and only if the signals are correctly decoded at each hop. Thus, the end-to-end ergodic capacity C_{e2e} is determined by the smallest capacity C_k [1]

$$C_{e2e} = \min_k (C_1, \ldots, C_k, \ldots, C_K). \tag{8}$$

Furthermore, the $\min$ function is concave and nondecreasing. According to the theory of the concavity of a composition function [6], a composition function $f(x) = h(g(x))$ is concave if h is concave and nondecreasing, and g is concave. Here, f is C_{e2e}, h is the min function, g is C_k. Clearly, C_{e2e} is jointly concave in $(\boldsymbol{\alpha}, \boldsymbol{\beta})$. Then, a concave optimization problem for maximizing the end-to-end channel capacity can be formulated as follows

$$\begin{aligned} \text{maximize} \quad & C_{e2e} = \min_k (C_1, \ldots, C_k, \ldots, C_K) \\ \text{subject to} \quad & \sum_{k=1}^{K} \beta_k = 1 \quad \text{and} \quad \sum_{k=1}^{K} \alpha_k = 1. \end{aligned} \tag{9}$$

With the total power and total bandwidth constraints, increasing any one capacity C_k inevitably reduces the others. The minimum is therefore maximized if all capacities C_k, $\forall\, k$ are equated, i.e. $C_1 = C_2 = \cdots = C_K$. By using the constraints in (9) and the approximation (4) a simple expression of the optimal bandwidth and power fraction follows

$$\alpha_k = \beta_k = \frac{\sqrt{d_k^{\epsilon}} G_k}{\sum_{m=1}^{K} \sqrt{d_m^{\epsilon}} G_m}, \tag{10}$$

where $G_m = \frac{\Gamma(t_m)\sqrt{t_m}}{\Gamma(t_m+1/2)}$ is introduced for convenience. It can be shown that $G_k \approx 1$ holds [7] and consequently a suboptimal but simpler solution of the power and bandwidth fraction can be obtained

$$\alpha_k = \beta_k \approx \frac{\sqrt{d_k^{\epsilon}}}{\sum_{m=1}^{K} \sqrt{d_m^{\epsilon}}}, \tag{11}$$

which only depends on the distances d_k.

4 Results

In order to analyze the proposed optimization strategy, a distributed MIMO multi-hop system consisting of 5 hops with $[1, 2, 3, 4, 5, 1]$ denoting the number of nodes per VAA is investigated. The distances between the hops are $[1, 1, 2, 2, 1]$ km. Figure 2 shows the ergodic capacity for different resource allocation strategies. We can see that the optimized power and bandwidth allocation according to (10) for the distributed MIMO system clearly outperforms the equal power and bandwidth allocation ($\alpha_k = \beta_k = \frac{1}{K}$, $\forall\, k$), the traditional SISO multi-hop transmission ($t_k = r_k = 1$, $\forall\, k$) and the direct transmission (the source node communicates with destination node directly without any relaying nodes). Note that even the suboptimal solution based on (11) achieves near-optimum performance.

Table 1 shows the optimal power and bandwidth fraction according to the closed form solution (10). The same results can also be achieved by applying common optimization tools for (9). We can see, that hops with large distance require more power and bandwidth than others.

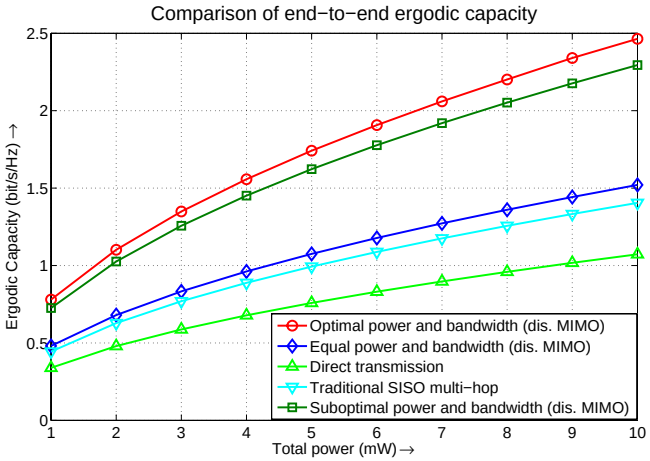

Figure 2: Ergodic channel capacity of a distributed MIMO multi-hop network for different resource allocation strategies. Network topology: 5 hops with nodes assignment $[1,\ 2,\ 3,\ 4,\ 5,\ 1]$ per VAA and distance $\mathbf{d} = [1,\ 1,\ 2,\ 2,\ 1]$ km, pathloss exponent $\epsilon = 3$.

Hop	1. Hop	2. Hop	3. Hop	4. Hop	5. Hop
Distance	1 km	1 km	2 km	2 km	1 km
Fractions $\alpha_k = \beta_k$	0.1263	0.1189	0.3250	0.3175	0.1124

Table 1: Power and band fraction according to (10).

5 Conclusion

In this paper we have briefly introduced the concept of distributed MIMO schemes, which allows the application of MIMO capacity enhancement techniques over spatially adjacent nodes. A concave optimization problem has been formulated for optimal resource allocation to maximize the end-to-end capacity of distributed MIMO multi-hop networks. Finally, we demonstrate that the optimal resource allocation strategy leads to a strong increase in ergodic capacities.

References

[1] M. Dohler, A. Gkelias and H. Aghvami. *"A Resource Allocation Strategy for Distributed MIMO Multi-Hop Commonication Systems"*. IEEE Communications Letters, Vol. 8, No. 2, pp. 98-101, February 2004.

[2] A. Goldsmith. *"The capacity of downlink fading channels with variable rate and power"*. IEEE Transactions on Vehicular Technology, Vol. 46, pp. 569-580, August 1997.

[3] J. Lanemann, D. Tse and G. Wornell. *"Cooperative Diversity in Wireless Networks: Efficient Protocols and Outage Behavior"*. IEEE Transactions on Information Theory, Vol. 50, No. 12, pp. 3062-3080, December 2004.

[4] I.E. Telatar. *"Capacity of multi-antenna Gaussian channels"*. European Transactions on Telecommunication, Vol. 10, No. 6, pp. 585-595, December 1999.

[5] M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover Publications, June 1965.

[6] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambrigde University Press, 2004, New York, USA.

[7] Y. Lang. *"Resource Allocation for Distributed MIMO in Wireless Networks"*. Diploma thesis, University of Bremen, Germany, September 2007.

Extension of an InfiniBand Host Channel Adapter Model and Performance Analysis

F. Auernhammer, A. Doering, M. Gabrani, P. Sagmeister
IBM Research Zuerich
Saeumerstrasse 4 / Postfach
CH-8803 Rueschlikon, Switzerland
Email: fau, ado, mag, psa@zurich.ibm.com

A. Herkersdorf
TU Munich
Arcisstrasse 21
D-80333 Munich, Germany
Email: herkersdorf@tum.de

Abstract: The requirements for I/O hardware components become increasingly demanding both in performance and in functionality leading to highly complex implementations, difficult to test and to upgrade. In this paper we use a C-written model of an InfiniBand Host Channel Adapter and use it to evaluate both the use of a high level language to model a complex ASIC design and the actual design. With the model we are able to illustrate the strengths and identify the bottlenecks of our architectural choices leading to early architectural changes and a more performant hardware design.

1 Introduction

Designing and testing an I/O ASIC has become a very cumbersome task due to a number of factors, including compliance to standards, performance and functionality requirements, future-proofness, cost and time-to-market constraints. Prior knowledge of the implementation effects of architectural choices of any design is of high value, making the use of system simulators an attractive option. Yet, one has to be very careful in the choices and the interpretation of the results.
In this paper we describe our enhancements to an InfiniBand Host Channel Adapter (IB HCA) model and evaluate its performance. With the first generation of InfiniBand adapters available today and improved versions on the way, it is fundamental to analyze existing implementations in order to detect bottlenecks and evaluate possible architectural alternatives in order to keep up with the ever increasing demands in link speed and quality of service already specified in the next version of the InfiniBand Architecture [Ass02].
The scope of this work is therefore aimed at providing a basis for further analysis of an InfiniBand host channel adapter by extending an existing model to comply as far as possible with the standard and to architectural choices based on implementation efficacies.
The HCA model used is written in C and embedded in a full system PowerPC simulation environment. Thereby we show the advantages and possibilities offered by today's system simulation environments that make them most attractive not only for software develop-

ment and system on chip simulations but also development of complex IO ASICs.
The simulation speed of the abstract event driven system simulator is seen as more apt to running extensive simulations than a VHDL-simulator. Yet the biggest advantage of this approach is the ease of implementing architectural changes. In the C-model changes to the system architecture can be realized more easily and rapidly than in VHDL.
Using the model we were able to make a competent decision on selecting between processor bus access schemes, identify sources of contention and propose alternative architectural options leading to performance improvement with respect to transmit rate. Having such input as early as the design phase is among the greatest benefits of implementing such models.
The paper is outlined as follows. In Section 2 we briefly present the state-of-the art in system simulators and detail the system simulator we use to model an HCA based on IB architecture that we overview in Section 3. In Section 4 we describe our architectural choices, our model and the extensions we made. Performance evaluation results are illustrated and discussed in Section 5. We conclude the paper with directions for further improvements in Section 6.

2 System simulators

Long before major processor vendors like IBM with their "systemsim" line of PowerPC [IBM07a] and Cell [IBM07b] system simulators and SUN with SAM (SPARC architectural model [Mic07]) for the T1 processor started offering full system simulation environments for their processors, system simulators were developed for hard- and software development. One of the first to appear was the SimOS [Sta07] simulator that was also used as basis for IBM's systemsim simulator. Simics [MCE+02] and M5 [BHR03] are two other successful full system simulators that were developed over the past supporting different processor architectures and operating systems while offering also networking capabilities.
With the ever increasing cost and complexity for testing enhancements in software and hardware for new processor and other functional units' generations, the trend goes towards using full system simulators for these tasks. The simulators differ especially in the degree of abstraction which is always a trade-off between simulation accuracy and performance. Therefore it is essential to select the most suitable carefully dependent on the actual requirements.
There are many advantages in favor of full system simulators. First of all they offer the possibility of pre-hardware software development. Using system simulators with an appropriate hardware-model allows software development, debugging and tuning in parallel to the development of the real hardware and thus development cycle times can be reduced considerably. Writing abstract models for a simulation environment can also ameliorate the architecture definition and help in hardware verification, the first by admitting fast and easy implementation of alternative designs or detecting problems beforehand, the second by extraction of traces and test cases for individual units that can be used for test benches for VHDL or Verilog code. Another interesting aspect is that system simulation environ-

ments allow for much more visibility and analysis in existing hardware. The simulation environment used in this work for example permits extensive analysis of pipeline effects, cache hit-rates, processor-internal operations during instruction execution and many other parameters. Last but not least a system simulator can be very convenient especially for software developers since they no longer need the real hardware to develop applications, lowering development costs and possibly attracting a broader community.

2.1 The Mambo system simulator

The utilized simulation environment called Mambo [BPS03] is an IBM proprietary full-system simulation toolset for the PowerPC architecture. It is completely written in the C programming language. Therefore it can be run on many platforms and operating systems, including Linux, AIX and MacOS-X.
The simulator supports the PowerPC 970 64-bit processor [IBM07a] as well as numerous 32-bit derivates, notably 405, 440 and 750. It is not limited to a certain system complexity but is rather expandable to simulate even supercomputer-like systems such as BlueGene/L [BPS03] and can be easily adopted for newer processor designs like the Cell [IBM07b] processor.
Speaking of system simulators it is important to distinguish between simulated and real hardware. Therefore the PowerPC system simulated by Mambo on top of the simulation stack in Figure 1 can be entitled ”virtual” whereas the computer system on which the simulation runs, forming the basis of the simulation stack, is the ”real” system.
The foundation of the simulation environment is an event driven system simulator providing supplementary C functionality to facilitate modeling hardware, that is different functional units working independently from each other, such as synchronization calls and resource-gating. Furthermore, different clock-domains, in our setup as shown in Figure 2 the PowerPC, the bus and the HCA, can be handled very conveniently by defining the slower clocks as fractions of the fastest. This allows for easy analysis of the impact of changing clock-speeds without the need to adapt the interfacing between clock-domains since it is handled by the system simulator.
The systemsim simulation environment has even some more distinct features. For most processors it offers two different simulation modes: simple and tempo. The simple mode simulates a purely functional model, executing one instruction per cycle, whereas the tempo mode can be used for accurate timing and power analysis. Therefore this model takes into account pipeline effects as well as cache dependencies. Switching between the two modes is possible during simulation runs, offering the possibility to advance to a certain position in program code taking advantage of the faster simulation of the simple mode and then switching to tempo mode for both accurate timing and power analysis. Furthermore single events can be tracked back to their origin both in source and assembly code. Another great advantage in this respect is the repeatability of simulations by the use of the same seed. The combination of these two features facilitates the task of finding a failure since simulation runs can be repeated accurately and by using different seeds, first conclusions about the origin of the failure can be drawn. More information on Mambo can be

found in [BPS03] and [SBP+03].
The HCA model is integrated in the full system simulator. For our study the environment

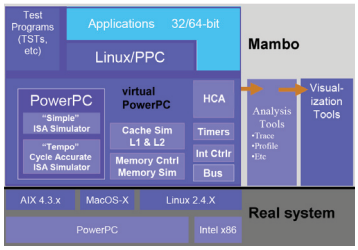

Figure 1: Simulation environment stack

includes a 64-bit PowerPC core with main memory, system bus and the actual host channel adapter. There exist numerous configuration possibilities for all aspects of the simulated system as well as analysis tools. The main advantage of this setup is that it offers the possibility to run test cases for analyzing the host channel adapter model directly in a Linux-environment running on the simulated processor and providing an API to access the host channel adapter functionality. As will be seen in the presentation of the InfiniBand Architecture and the HCA model, the InfiniBand Architecture is very complex since it covers four layers, from the physical to the transport, of the OSI model and provides various protection mechanisms. Therefore the host channel adapter has to be correctly set up for running tests on it which requires much interaction between the user and the device. This process is greatly alleviated through the use of the Linux driver compared to a VHDL- or SystemC-model which would both require the initialization and handling of requests from the channel adapter in a testbench.
As the ultimate motive of this work is being able to test alternatives to the system architecture, it is desirable to have quickly changeable interfaces between different units. C offers this possibility because interfacing is mainly implemented by passing structures containing all necessary data whereas VHDL-models use data-buses complicating changes in architecture as well as monitoring process data.
One drawback is that the behavior of state machines cannot be modeled in all detail. With regard to the complexity of the InfiniBand Architecture however, interdependencies between different units and system memory has much bigger effect on the system performance compared to the single state machine delays. Therefore, altogether the C-model's advantages greatly outweigh its deficiencies.
For this work we use a setup as displayed in Figure 2. The Mambo system simulator runs

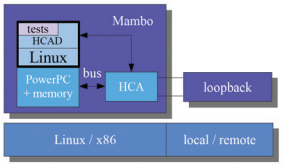

Figure 2: Mambo-HCA setup

on a x86-based Linux computer. The testcases for analysis and performance evaluation use a Linux host channel adapter driver (HCAD) for interfacing the HCA and are executed in the Linux OS running on the simulated PowerPC within the system simulator. An external loopback is provided in order to test both the send and receive processing. The loopback can be either local or on a remote computer. In our further work we envisage therefore extending this setup to multiple simulated systems interconnected with each other since our current setup implies high correlation between send and receive side operations.

3 InfiniBand Architecture overview

InfiniBand is a high-performance serial computer bus intended for both internal and external connections. Due to its high bandwidth and low latency, combined with low processor load, it is mainly used as external link in High Performance Computing (HPC) environments such as clusters and mainframes. It can be paralleled to high-speed interconnects such as Myrinet, Quadrics [LCJW+03] and SCI [HR99].
The InfiniBand Architecture (IBA) is based on a point-to-point switched I/O fabric and was intended for two major interconnect environments:

- Module-to-module connection in systems (for example through add-in slots)
- Chassis-to-chassis in a data-centre environment

It therefore defines both reliable and unreliable messaging (send and receive) for datagram- and connection-oriented communication and memory manipulation (remote DMA) without software intervention in the data movement path using zero-copy mechanisms.
The InfiniBand Architecture defines a System Area Network (SAN) for connecting multiple independent processor platforms (i.e. host processor nodes), I/O platforms and I/O devices (see Figure 3). The IBA SAN is a communication and management infrastructure supporting both I/O and inter processor communication for one or more computer systems. An IBA system is thus suitable for small servers with one processor and a few I/O devices as well as for massively parallel supercomputer installations.
Connection to the SAN is established through so-called Channel Adapters (CA). There are two types of channel adapters defined: Host Channel Adapters (HCAs) and Target Channel Adapters (TCAs). The HCA provides a consumer interface with all the functionality specified by the IBA whereas IBA does not specify the semantic of the consumer interface for a TCA. HCAs are therefore mainly used for connecting single processors or processor nodes to an InfiniBand fabric whereas TCAs are used for I/O units such as storage devices and InfiniBand-bridges where the full InfiniBand functionality is not required. The channel adapter, schematically shown in Figure 4 provides multiple instances of the communication interface to its consumers in the form of queue pairs (QP) (a), comprised of a send (c) and receive (b) queue. Consumer work requests having the form of Work Queue Elements (WQE) are queued up thereupon which the hardware processes autonomously, similar to the virtual interface architecture (VIA) [CC97]. Work queue elements don't contain the messages but rather pointers to the message location in main memory and what to

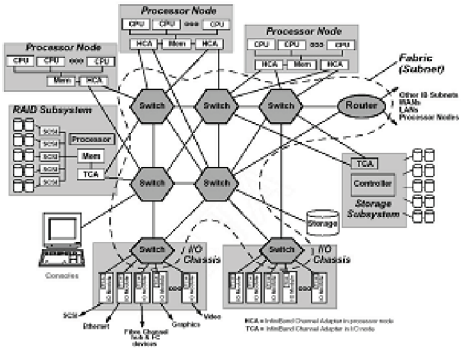

Figure 3: IBA System Area Network (SAN)

do with it. The queue elements stay in system memory until the channel adapter is able to begin processing. Work queues are always created in pairs, therefore the name Queue Pair (QP), one for send operations and one for receive operations. In general, the send work queue holds instructions that cause data to be transferred between the consumer's memory and another consumer's memory, and the receive work queue holds the instructions about where to place data that is received from another consumer. A QP can therefore be characterised as a bi-directional message transport engine that can be directly accessed by the user without operating system interference. Each consumer can create as many QPs as necessary. In spite of QPs sharing the same Queue Pair Context (QPC), the send and receive queue can partly differ in their initialization. The Queue Pair Context contains information such as the service type of the queue pair and the addresses of the according send and receive queue in main memory.

There is a third type of queues, called Completion Queue (CQ). Completion queues are

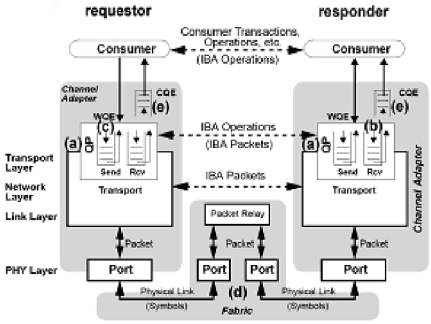

Figure 4: InfiniBand communication

created separately from QPs, that encapsulate send and receive queues, and have an own Completion Queue Context (CQC). Transactions, no matter if send or receive, can be configured to create a Completion Queue Element (CQE) on the completion queue specified in the QP context for signaling work-completion to the system.

4 The Host Channel Adapter Model

4.1 Architectural choices

An initial C-model of an InfiniBand compliant HCA was written to facilitate driver and software development. We based our work on this model and extended it as explained in section 4.3. The implementation is dataflow oriented and based on following architectural decisions:
The model is connected to the PowerPC over a single bus which is shared by all host channel adapter units.
The packet processing reflects the dataflow of the InfiniBand Architecture. Thus it is divided into send side and receive side processing, complementing the send queue and the receive queue of a queue pair, independent from each other.
There is a number of resources needed by both the send and receive side, notably the queue pair context (QPC), the address translation (AT) and the completion queues (CQ). Therefore those units are implemented as single shared resources.
Two ports, shared between send and receive side, are used for connecting to the InfiniBand SAN.
All queues are implemented as circular buffers that reside in main memory. Only a small part of the contexts is cached in the host channel adapter, the others reside in a backing-store in main memory.

4.2 The architecture in detail

The full architecture of the HCA model that was developed under the requisites presented above is shown in Figure 5.
The bus unit is the main communication interface between the host channel adapter and the PowerPC-memory compound. All memory accesses both in read and write are accomplished through this unit for all queue and data accesses. However, the host channel adapter configuration registers are accessed directly by C-functions in this model. The whole host channel adapter functionality can be configured through these registers as well as the setup of new queue pairs, completion queues and event queue contexts. The structure of the configuration registers is modelled as it would be designed in hardware, that is all registers are implemented with a width of 64-bit and bits and bit-strings hold the configuration data.
The WQE-dispatcher has several functions in the model. First of all it is called to notify the host channel adapter hardware about new WQEs attached to a QP. All ports instantiated in the model feature a work-list for each supported service level (SL), a facility to allocate different connection priorities for different QPs. Queue pairs with outstanding work are attached to one of those lists according to the port and service level configuration in the queue pair context. The WQE-dispatcher checks that the QP does not still reside on the list from previous additions and attaches the QP to the list. Another function is comprised in the WQE-dispatcher for QP scheduling. It determines which QP will be processed next

by a send queue processor, based on SL arbitration mechanisms. Every time a send queue processor finishes work on a QP, it calls the scheduler in order to check for outstanding work.
The QP context is essential for most units of the host channel adapter since all processing tasks depend on the configuration of the QPs. Because of the InfiniBand Architecture defining a very large number of QPs, most of them would be stored in system-memory or external backing store while only recently used ones reside in a limited on-chip cache. The C model on the contrary is implemented without backing store at the moment and uses only a limited number of QP contexts.
The address translation unit (ATU) is essential for the host channel adapter to provide virtual addressing necessary for memory protection mechanisms specified by the IBA.
The send queue processor (SQP) as one of the main two units responsible for the perfor-

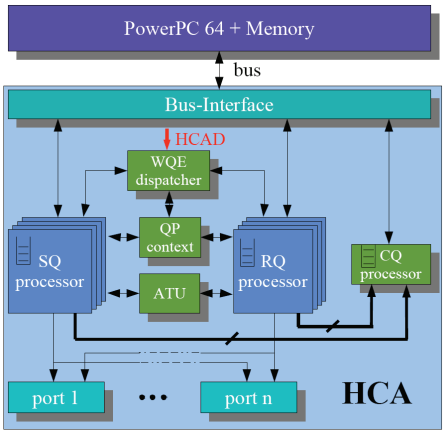

Figure 5: HCA model structure

mance of the host channel adapter can be instantiated several times in order to improve latency and throughput of the system. It waits for the WQE dispatcher to indicate outstanding work. Through calling the QP scheduler integrated in the WQE dispatcher, it receives the QP it has to work on. Since the send queue processor will affect changes to the queue pair context, every QP may only be active in one send queue processor at a time. Therefore higher bandwidth for a communication channel can only be achieved by using different queue pairs, preferably attached to different ports. Each send queue processor is essentially the same and can handle any service type as well as RDMA operations. Processing begins with the fetch of the first WQE from the QP's send list. The WQE essentially contains a data descriptor with data address and length. In order to translate the given address into a real memory address the send queue processor calls the address translation unit. Afterwards the packet processing can begin. The data that is to be transferred is fetched from main memory over the bus, the header generated and CRC checksums calculated. After finishing with all send related operations the send queue processor submits a completion queue element to the completion queue processor for the completion queue specified in the queue pair context. After a specified number of transmitted data bytes, the

processing of WQEs is interrupted in order to check for outstanding work on other queue pairs and service levels.
The receive queue processor (RQP) being the other unit mainly defining the host channel adapter's performance can be as well instantiated several times. It waits for wakeup calls from the port to begin processing of incoming packets. Those are checked for correct header setup and further processed if the check passes successfully. Subsequently the receive queue processor calls the ATU to translate the memory address specified in the next WQE on the receive queue of the QP indicated by the received packet. In contrast to send queue processors, receive queue processors keep working on the same queue pair as long as packets are available for it in the receive buffer in order to speed up processing by avoiding context changes and blocking of the input buffers. Still the queue pair context architecture restricts the use of a queue pair in only one receive queue processor.
The number of physical ports is another degree of freedom in the system with the only restriction that there has to be at least one. Common host channel adapters provide one or two ports. The model used in this work uses two ports for establishing the loopback. The port-speed complies with a 12-lane double data rate InfiniBand link with 60 Mbps and is monitored to assure realistic transmission rates and estimate system speeds for different testcases.
The completion queue processor plays a crucial part for the whole host channel adapter. In our model, all work requests, send and receive, create a completion queue element through this unit in order to notify the system of work completions. Therefore in larger host channel adapter systems with a fair amount of send queue processors and receive queue processors instantiated, the completion queue manager will have to handle all requests quickly enough in order not to block the queue processors which would have a negative impact on the system performance. The completion queue processor takes the requests from the queue processor for completion queue element posts, creates a completion queue element and attaches it to the specified completion queue in main memory.
Summarizing the features of the model presented above, it offers a scalable architecture for send and receive side processing by providing the possibility to instantiate send and receive queue processors several times. The available connection bandwidth can as well be scaled by using more ports. The control units on the other hand are shared and limited to one instance for each.

4.3 Extensions

The host channel adapter model used was originally written for developing and testing the driver for an InfiniBand host channel adapter. It implements the architectural choices made but is mainly focused on correct data structures and data flow. Furthermore it originally used only one send and receive queue processor. For analyzing the architecture as it would be available in real hardware and evaluate its performance, we therefore enable the use of multiple send and receive queue processors and extend the model with necessary control flow elements.
The original HCA model for example correctly implements the functionality of the com-

pletion queue processor but neglects that there is only one such unit present in the system. Thus receive or send queue processors can post a completion queue element at the same time. We consequently adapt the model in order to queue CQE post requests up and process them sequentially. The same scenario is possible for most units and we ensure therefore, first of all, that every unit in the model can handle only one request from other units at a time just like in real hardware. Other requestors are kept waiting meanwhile and are served sequentially.
In order to be as flexible as possible for further extensions, we introduce an arbitration for the CQ-processor which features two advantages:
On the one hand the implementation reflects a realistic hardware model, i.e. only one SQP or RQP can post a CQE at a time and meanwhile other SQPs and RQPs ready to post a CQE have to wait until the completion queue processor is able to process their request. On the other hand the implementation is held flexible, offering the possibility to test the impact of architectural changes. It therefore allows testing different scheduling algorithms such as load-balancing between SQPs and RQPs as well as instantiating multiple completion queue processors. For this work these capabilities are not yet exploited and the HCA model used complies with the model shown in Figure 5.
With respect to the performance evaluation, we introduce an identification structure for the different units throughout the model. Especially since SQPs and RQPs can be instantiated several times they have to be clearly distinguishable in order to track requests back to the originator. In the course of these changes a general identifier structure is inserted that can eventually be extended for monitoring purposes as well.
With the addition of those control flow elements, the C model behavior is comparable to a behavioral VHDL model. Delays integrated in the different units can be used and adapted in order to simulate processing times as well as to test how changes of these delays will affect the system performance.

5 Performance Evaluation

5.1 Monitoring

Because of the asynchronous interface to the HCA through queues and the arbitration mechanisms implemented in hardware, the host channel adapter is a self-contained system. Therefore, external monitoring, for example in the Linux-kernel, does not suffice for analyzing the HCA model since it does not offer much insight about internal mechanisms and dependencies.
Analyzing the model architecture, due to the dataflow oriented architecture, send queue and receive queue processors are dependant on all other units, called subunits in the sequel, while defining the performance of the HCA. The minimum time to send or receive a packet consists of the time it takes to fetch a WQE over the bus, get the according QP context from the CQ-manager, translate the address contained in the WQE using the ATU, fetch data from memory or write it and finally post a completion queue element. The optimal performance is achieved with no wait times during the requests to the different

subunits. However, with several SQPs and RQPs instantiated and the correct implementation of the control flow, contention for these supplementary units jeopardizes the system performance.
Analysis of the HCA model architecture is therefore established by gathering important data of the system, especially idle and active times as well as the originator of the request for the subunits. Using these, the average usage is calculated for fast evaluation of its degree of utilization. Furthermore every work queue element is tracked during processing, recording the times elapsing while the handling SQP or RQP has to wait for the different subunits. Comparing the total processing time to the smallest possible indicates the overall degree of contention and the trace allows for determining where it takes place and how important it is.

5.2 Testcase setup

The first performance tests of the host channel adapter model are carried out using the host channel adapter driver written for the original model. In the present performance evaluation we concentrated on generic single packet size traffic. Each testcase uses one packet size and six queue pairs in order to be able to fully load an HCA model with six send queue and receive queue processors instantiated in it. One queue pair per send-receive queue processor pair is necessary since one queue pair can only be active in one send queue processor and one receive queue processor at a time. The traffic is artificial but seems appropriate to identify hot spots in the system architecture while verifying the changes to the control flow we make. Furthermore, so far, there is almost no information on ”real” InfiniBand traffic available. Driver performance issues also prevented us from using standard benchmarks. However, we would not expect a single PowerPC processor to be able to generate enough traffic to fully load host channel adapter models with several send and receive queue processors instantiated, necessary to evaluate the architecture.
The packet payload sizes in the testcases range from 128 up to 4096 bytes and we run each testcase on different model setups, the smallest being a host channel adapter model instantiating only one send queue and one receive queue processor, the largest using six of each type. Furthermore we run this set of testcases both for unreliable datagram and reliable connected services. We always use two ports because of the setup using an external loopback. In our model, receive queue processors operate slightly faster than send queue processors. Therefore, we can always use the same number of send queue and receive queue processors without introducing a supplementary bottleneck on the receive side which we confirmed with tests.

5.3 Results

Using the testcases described above, we analyse how the architectural decisions described in 4.1. affect the HCA model performance including the control flow extensions presented above.

First of all we tested two different implementations for the bus unit: the first, whose results are shown in Figure 6, returns acknowledges for memory-write commands to the requesting unit directly after sending the data over the bus. The second waits for the memory controller to acknowledge that the data was successfully stored in main memory and issues the acknowledge to the requesting unit only thereafter. Results for this case are shown in Figure 7. The figures show the evolution of the transfer rates for unreliable datagram services with increasing resources available in the model for different packet payloads. It can be interpreted both as packet rate or bandwidth of the host channel adapter referenced to the rate measured for one send and receive queue processor pair. Ideally, the model with 6 SQPs and RQPs instantiated would also achieve six times the rate of the reference model. For 4096 bytes payload for example, the maximum transfer performance attains approximately 5 respectively 3 times the traffic generated by a single send queue-receive queue processor pair.

The results show on the one hand that the system performance is heavily dependent on

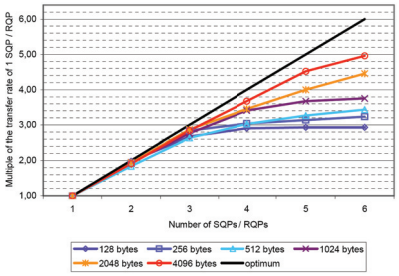

Figure 6: UD normalized transfer rates

the bus-unit latency. Using the first, more aggressive model improves transfer rates by 50 to 60% for all packet sizes in larger systems. Because of the superior performance, we use the first implementation in all further tests.

On the other hand, the overall performance does not scale as desired. Both figures show

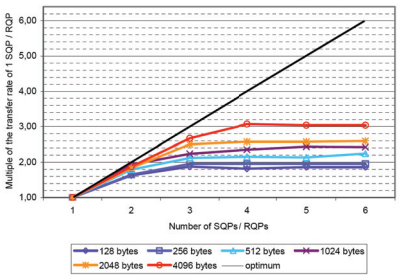

Figure 7: UD rates with slow bus-model

very characteristic fanning out of the transmission rates with saturation for more complex systems. This behavior can be tracked back to the completion queue processor.

Figure 8 shows an analysis of this unit. The results are derived from a test run with unreli-

able datagram packets of 4096 bytes payload. Light blue (■) are the completion queue wait times in cycles. Dark blue (◆) depicts the number of completion queue posts still waiting for processing. Black (bold line) finally is the load of the completion queue processor. The numbers above the different sectors indicate the number of send and receive queue processor pairs instantiated which is, at the same time, the number of simultaneously active QPs. Hence, each sector can be compared to the according number of send queue and receive queue processors in Figure 6 (○) which directly relates the effects of the completion queue processor contention on the transfer rate. Since there is only one completion queue

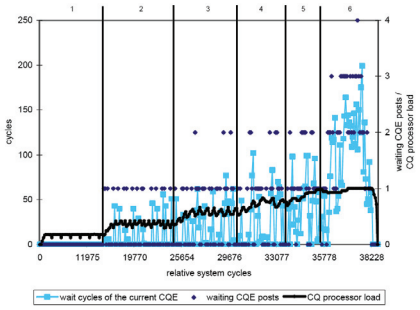

Figure 8: CQ processor behavior

processor present in the system, the load can range from 0 to 1.0. For only one queue pair sending, displayed in the first sector up to 11975 relative system cycles, no completion queue elements are ever waiting for an element post due to the processing scheme and processing times: while the send queue processor posts its completion queue element, the receive queue processor writes the received data to main memory. When finished, it posts a completion queue element. At this time, the send queue processor is busy fetching data for the next packet from main memory. Thus there is no overlap in the completion queue element post process. When two send queue processors are active, overlapping of posts to the completion queue from send queue or receive queue processors is possible. While increasing processing resources up to five send and receive queue processors active, the load of the completion queue processor increases gradually but never actually reaches one (system cycles 11075 through 35778). Therefore also the completion queue wait times are limited while slightly degrading the transmission rate nevertheless. However, for six send queue processors, the completion queue processor gets overloaded, having a constant load of 1.0 (sector 6). Furthermore the number of completion queue element posts waiting increases to constantly 2 to 4 and the number of cycles a CQE has to wait for processing increases considerably. Thus the completion queue processor in our model can handle about 10 queue processors active simultaneously for 4096 byte packets. Due to shorter processing times for smaller packets, this number decreases for smaller packet sizes which is responsible for the fanning in Figure 6 and 7.
Figure 9 offers insight into when the point of overload is actually reached, dependent on the service type and the packet payload used. For the unreliable datagram service, from 4 send queue processors instantiated in the system on, the maximum wait time for posting a completion queue element increases linearly with about 110 to 120 cycles per send queue

/ receive queue processor pair. In our simulation, this is the time the completion queue processor needs to process two CQEs, one for sending and one for receiving a packet. The two numbers for 128 and 256 bytes payload packets and 6 send queue processors to break ranks can be attributed to the absence or insufficient duration of simultaneous processing of 6 queue pairs in the system, due to driver restrictions. Therefore, as well as the bus-

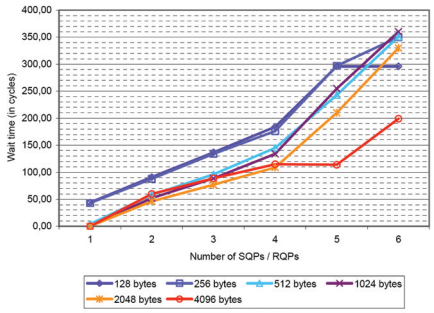

Figure 9: UD max. CQE post wait times

unit implementation and latency, also the single completion queue processor in the host channel adapter architecture has a heavy impact on its performance. Using at least two as already envisaged by our extensions should deliver better performance but possibly shifts the bottleneck to another subunit.
Reliable connected transmissions, shown in Figure 10, are less critical for the completion queue processor arbitration than unreliable datagram transmissions because the processing of send queue elements takes much longer. It includes processing of the data packet as well as the time the responder's receive queue processor needs to check the context of the received packet as well as storing the received data in main memory and returning an acknowledge message. Furthermore the load of the receive queue processors is only about half that of the send queue processors - for unreliable datagram services they are almost identical. Most of the send queue processor execution time however is spent waiting for acknowledges and data. Hence, the normalized transfer rates shown in Figure 10 are almost optimal, contention on the completion queue processor is not a limiting factor. The absolute transfer rates however are much smaller, only about 60% of unreliable datagram services. This number is likely to deteriorate even more within a larger fabric. Allowing the send queue processor to proceed with processing other WQEs while waiting for acknowledges or introducing a supplementary unit in the architecture for handling acknowledges could considerably increase the performance of reliable services.

Considering future-proofness and scalability of the architecture, the biggest problem is presented by the lack of information about real InfiniBand traffic. Therefore it is not easy to exactly define the starting point for improvements, whether it should be focused on many small packets, on big packets or an evenly divided mixture. Using bigger on-chip caches for example could be applied in any scenario since it reduces the number of cast-outs and thus the latencies in packet processing. For testing the effects of such changes to the HCA architecture, the model can be easily adapted and the performance results evaluated.

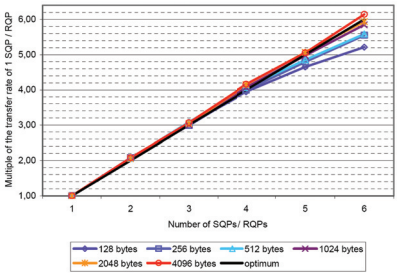

Figure 10: RC normalized trasnfer rates

6 Conclusions and Outlook

In this paper we use a C-written system model to evaluate the performance of a complex I/O hardware component architecture, more specifically of an InfiniBand host channel adapter. We show in this work that the host channel adapter model presented, embedded in the Mambo full system simulator, allows for fast and easy analysis of the host channel adapter architecture including the possibility to use real world applications for performance analysis.
Using a dataflow oriented model with control flow elements presented some major problems for more accurate integration of the control flow in the beginning. While small setups scale rather well in our simulations, surpassing a distinct number dependent on the service type, contention especially on the completion queue processor causes the transmission rate to saturate. From an architectural point of view, the host channel adapter performance is limited for transmission of smaller packet sizes by the single completion queue processor slowing down the whole system, especially in unreliable datagram connections. Therefore we implement this part as flexible as possible in order to be able to increase the number of completion queue processors in the system. Furthermore the bus model plays a crucial role for this analysis. Therefore we will look further into improving its accuracy.
The performance issues for small packets are also due to the InfiniBand Architecture which is based on using different lists for the communication between the system and the channel adapter. This entails a huge overhead for example in bus crossings, which can be several times the actual payload for small packets, restricting the maximum transfer rate for them. Improvements in this respect will also be a major point of investigation for our future work.
Scaling the presented architecture up for the next InfiniBand specification of quad-data-rate transmission and therefore increasing the number of send, receive and completion queue processes will cause increasing contention in the HCA, reducing the efficiency of resource increases. Therefore an approach with several small, efficient processing blocks, each having context managers but sharing caches for consistency reasons, could be considered instead. The model can therefore be adapted in order to test the effects on the system performance.

7 Acknowledgements

We want to thank Peter Walker from IBM Austin for providing the original host channel adapter model and his support in setting up the simulation environment and Thomas Wild from the Institute for Integrated Systems at the Technical University of Munich for his technical support during the conduct of this work.

References

[Ass02] InfiniBand Trade Association. InfiniBand Architecture Specification, Release 1.1, November 2002.

[BHR03] N. Binkert, E. Hallnor, and S. Reinhardt. Network-Oriented Full-System Simulation using M5. *CAECW*, 2003.

[BPS03] P. Bohrer, J. Peterson, and H. Shafi. Mambo: Advances in PowerPC System Simulation. *ISPASS Workshop*, March 2003.

[CC97] Intel Compaq and Microsoft Corporation. Virtual Interface Architecture Specification, Version 1.0, December 1997.

[HR99] H. Hellwagner and A. Reinefeld. *SCI: Scalable Coherent Interface, Architecture and Software for High Performance Computer Clusters*. Springer, 1999.

[IBM07a] IBM. Full-System Simulator for IBM PowerPC 970 2006. http://www.alphaworks.ibm.com/tech/systemsim970, viewed 23 July 2007.

[IBM07b] IBM. IBM Full-System Simulator for the Cell Broadband Engine Processor 2005. http://www.alphaworks.ibm.com/tech/cellsystemsim, viewed 23 July 2007.

[LCJW+03] J. Liu, B. Chandrasekaran, W. Jiang J. Wu, S. Kini, W. Yu, D. Bantinas, P. Wyckoff, and D.K. Panda. Performance Comparison of MPI Implementations over InfiniBand, Myrinet. In *Quadrics. Supercomputing Conference*, page 58, 2003.

[MCE+02] P. S. Magnusson, M. Christensson, J. Eskilson, D. Forsgren, G. Hallberg, J. Hogberg, F. Larsson, A. Moestedt, B. Werner, and B. Werner. Simics: A full system simulation platform. *Computer*, 35(2):50–58, 2002.

[Mic07] Sun Microsystems. OpenSPARC T1 2006. http://opensparc-t1.sunsource.net/download_sw.html, viewed 23 July 2007.

[SBP+03] H. Shafi, P. J. Bohrer, J. Phelan, C.A. Rusu, and J. L. Peterson. Design and validation of a performance and power simulator for PowerPC systems. *IBM Journal R&D*, 47(5/6), September / November 2003.

[Sta07] Stanford University. Stanford. SimOS - The Complete Machine Simulator 2006. http://simos.stanford.edu, viewed 23 July 2007.

Using Cramer-Rao-Lower-Bound to Reduce Complexity of Localization in Wireless Sensor Networks

Dominik Lieckfeldt, Dirk Timmermann
Department of Computer Science and Electrical Engineering
Institute of Applied Microelectronics and Computational Engineering
Richard-Wagner-Str. 31
18119 Rostock-Warnemünde
Email: dominik.lieckfeldt@uni-rostock.de

Abstract: The Cramer-Rao-Lower-Bound (CRLB) is investigated via simulations for a wireless sensor network that consists of nodes with known position, so called beacons, and nodes whose positions are not known, so called unknowns. For those networks, the fundamental bound on the accuracy of localization based on received signal strength is investigated using the CRLB. Thereby, we answer the question whether it is possible to determine a subset of available beacons for calculating the CRLB without significantly decreasing the achievable accuracy of localization. It is assumed that more beacons are available than are needed to ensure unambiguous localization. Simulations for a general scenario show that it is possible in many cases to decrease significantly the number of beacons used while ensuring that the increase of CRLB is only marginal.

1 Introduction

In wireless sensor networks (WSN), location is a valuable parameter which enables a variety of applications. For example, in order to react precisely on emerging phenomena like fires in forests or hazardous environmental conditions in sewer systems, the geographic position of the affected region has to be available. Furthermore, location information allows for target-oriented routing in WSN as, for instance, wake-up commands can be delivered exclusively to those nodes being in the area of interest.
In real applications due to perturbations, position cannot be determined exactly but has to be estimated using, for example, measurements of received signal strength (RSS) as indicators for distances between unknowns and beacons . Considering wireless communications between nodes of the WSN, phenomena like multipath, interference and noise contribute to the perturbations, which results in erroneous estimates of position. The more measurements vary, the greater the uncertainty in the position estimate will be. The Cramer-Rao-Lower-Bound (CRLB) poses a lower bound on the variance of any unbiased estimator [Kay93]. In practice, the CRLB is used to evaluate estimators by comparing their performance with the ultimate variance bound posed by the CRLB.
As the CRLB for RSS-based localization is an indicator for the best accuracy achievable, we use it to compare localization accuracy using all available beacons and using just a subset of beacons. As the computational complexity and also the communication effort

of localization largely depend on the number of beacons, this concept eventually supports energy-aware WSNs. As a first study, we use the CRLB to select the subsets to investigate the potential benefits of this approach. We conducted simulations for scenarios with a relatively large number of beacons and investigate the change of CRLB when decreasing the number of beacons used. Assuming a large number of beacons is justified as in real WSNs, unknowns will become beacons with more or less accurate estimates of position as soon as they have performed localization using the surrounding beacons. Therefore, it is likely, in later states of the WSN, that one node desiring to estimate its position or to improve it, will have a large number of neighboring nodes with known or estimated positions to choose from. The results of simulations indicate that in the majority of cases a considerable number of beacons can be discarded without significantly increasing CRLB.

1.1 Related Work

The problem of selecting a subset of beacons in order to optimize localization in terms of computational complexity has hardly been studied in the literature. There are papers trying to improve accuracy by weighting range measurements according to their variance and distance [LZZ06, CPI06, BRT06]. Others apply tests to detect outliers in order to exclude them from calculations or just choose the nearest beacons for estimation of position [OLT04]. However, these approaches do not exclude insignificant beacons from calculations and communication related to localization. Furthermore, while the impact of geometry has often been stated in the literature, we are not aware of any work which includes geometry for selection of beacons. Being the first study of this topic, we investigate the impact of excluding a number of beacons from calculation of CRLB in a WSN to access the maximal localization accuracy. The scenario setup and a short review of CRLB are given in section 1.2 and 1.3, respectively. Section 2 presents results of simulations. Conclusions and directions for future work follow in section 3.

1.2 Scenario

The scenario considered consists of m nodes, so called beacons, whose positions are known by means of, for example, GPS measurements and n nodes, called unknowns, whose positions shall be determined. This results in a total of $N = m + n$ nodes which are randomly distributed over an area. Nodes are enumerated by indices starting with unknowns $i, j \in [1, 2, ..., n-1, n, n+1, , n+m]$. Without loss of generality, we limit the consideration to 2D but extension to 3D is straight forward. The true positions of nodes are $z_i = (x_i y_i)^T$ with distances $d_{i,j} = \|zi - zj\|_2$. Estimated parameters are indicated by $\widetilde{x}$. All nodes are equipped with transceivers enabling them to communicate with each other using radio frequencies (RF) and to determine the received signal strength (RSS) $P_{i,j}$ between nodes i and j ($i \neq j$), which are in direct neighbourhood.

1.3 Reviewing the CRLB

The derivation of the CRLB for RSS has been studied for example in [PAICO03] and we just introduce the bound here for the case of n=1 unknown. The CRLB for RSS based localization assuming a log-normal shadowing model for the channel is:

$$o^2 = \frac{1}{b} \frac{\sum_{i=2}^{N} d_{1,i}^{-2}}{\sum_{i=2}^{N-1} \sum_{j=i+1}^{N} \left(\frac{d_{1\perp i,j} d_{i,j}}{d_{1,i}^2 d_{1,j}^2} \right)^2} = E\{(x-\widetilde{x})^2 + (y-\widetilde{y})^2\}$$

$$b = \left(\frac{10 n_p}{\sigma_{rss} \ln 10} \right)^2$$

The CRLB expresses a bound on the mean square error of position estimates averaged over x- and y-directions, which means: the larger σ^2 the lower is the maximally achievable localization accuracy. σ^2 depends on the distance between unknown and beacons and the "geometric condition" of the triangle with vertices at the positions of the unknown and beacons i and j. Important for the "geometric condition" is the parameter $d_{1\perp i,j}$ which denotes the shortest distance from the unknown to the line between beacons i and j. The well-known channel parameters n_p and σ_{rss} denote the path loss exponent and the standard deviation of the received signal strength. In [PIP+03] these parameters have been determined based on indoor experiments and we use these results ($n_p = 2,3$, $\sigma_{rss} = 3,92dBm$) for our investigations. As the fraction in the denominator has dimension of $(rangeunit)^2$, σ^2 scales with distance even if geometry is kept the same. This is a major drawback of localization based on RSS in comparison with Time-of-Arrival (TOA) because it leads to larger errors the farther beacons and unknowns are separated.

2 Simulation and Results

As demonstrated in the former section, geometry is an important point for CRLB and thus for localization accuracy. Our simulations investigate the impact of discarding some beacons for the case of m=13 beacons and one unknown (n=1). In the following, we use the lower bound on the standard deviation, which is the square-root of the original CRLB as it has the same unit as the distances and can therefore more intuitively be related to real localization errors. Unknown and beacons are uniformly distributed over an area 1000 times. For every deployment s,

$$\sigma^{(s)}(k) \left(s = 1, \ldots, \binom{m}{k} ; k = 3, \ldots, m \right)$$

is calculated, which is the standard deviation of the location estimate for the one unknown using k of the m totally available beacons. For the case of $k = m$, all of the 13 available

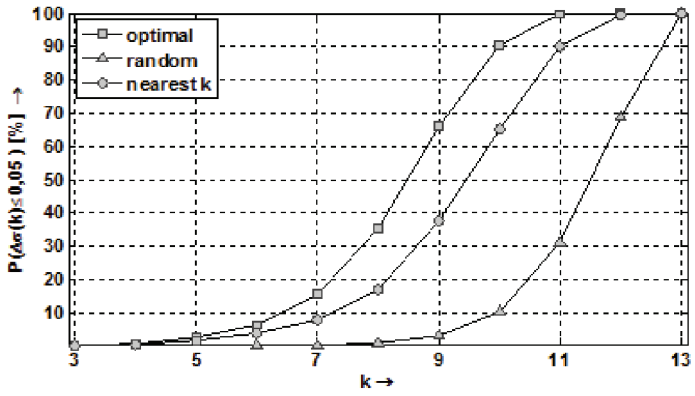

Figure 1: Occurrence of only small increase of localization error (Δs = 0,05) when considering only k of the m=13 totally available beacons.

beacons are used. If $k < m$, only a subset is considered whereby there are $\binom{m}{k}$ possible combination for choosing a subset. For every deployment and for every combination, the following ratio is calculated:

$$\Delta\sigma^{(s)}(k) = \frac{\sigma^{(s)}(k)}{\sigma^{(s)}(m)} - 1$$

For a specific subset of beacons, the smaller this ratio is, the smaller the decrease of localization accuracy will be when considering only this subset. As we are interested in guaranteeing that discarding some beacons will introduce only small additional errors, we define a threshold on this ratio and investigate via simulations how often this threshold holds. We choose $\Delta\sigma^{(s)}(k) \leq 0,05$ as a reasonably small threshold, which actually means that discarding some beacons shall not increase σ by more than 5% relative to $\Delta\sigma^{(s)}(m)$. The question to be answered is: "How likely is it that for a random deployment there exists at least one of the total $\binom{m}{k}$ combination to choose a subset of k beacons without violating this threshold?". We limit our presentation to the particular case of m=13 for convenience and due to limited space. However, the results presented are similar for other numbers of beacons.

The curve for "optimal" selection in figure 1 depicts the fraction of all 1000 simulated deployments for which it was possible to find at least one combination to limit the number of beacons to k without violating $\Delta\sigma^{(s)}(k) \leq 0,05$. For example, for the case of choosing a subset consisting of k=9 beacons, it is possible in 66% of the simulated deployments to find at least one combination for which the threshold holds. However, in real applications there will be not enough information available at each beacon to perform an optimal selection. A first trivial approach would be to randomly choose subsets. The performance of this approach is depicted by curve "random" in figure 1.

In the literature, as stated in section related work, a common approach is to choose the

nearest k beacons. This approach is a considerable improvement concerning the random selection. Nevertheless, there is still space for further improvements, as this approach does not consider the geometry of the situation.

3 Conclusions and Future Work

The goal of our study was to investigate whether it is possible to discard a number of beacons while keeping the increase of the CRLB below a specific small threshold. We used the CRLB to determine the importance of one particular beacon for localization and included the beacons with the greatest contribution to localization accuracy first. The simulations conducted indicate that it is possible to discard significant fractions of beacons without decreasing the maximally achievable localization accuracy significantly for a specific deployment. Regarding the results presented, 9 of the total 13 beacons could be discarded in 66% of the simulated deployments for ideal selection. Therefore, in 66% of cases it is possible to significantly reduce the complexity and the communication effort of localization. Despite a substantial improvement of success rate, choosing the nearest k beacons still leaves space for further improvements with regards to ideal selection. This is due to the fact, that choosing the nearest beacons does not consider the geometric condition of the situation. Although, we only presented the case of m=13 beacons, we obtained similar results for other numbers of beacons.
Next steps will include development and analysis of heuristics to perform the task of selecting subset of beacons given a specific abstract target accuracy (for instance: ”low”, ”medium”, ”high”). This will enable applications to adapt localization accuracy to contextual parameters, e.g. importance and energy level, of a specific node. A key issue will be the comparison of ”include nearest k beacons” approach with approaches that include geometric information of the situation. Furthermore, we will apply the idea presented to other localization schemes. We suspect that geometric information can be efficiently used for selection of beacons especially for time of flight based schemes as the CRLB does not scale with distance here but also depends on geometry.

References

[BRT06] Jan Blumenthal, Frank Reichenbach, and Dirk Timmermann. Minimal transmission power vs. signal strength as distance estimation for localization in wireless sensor networks. In *3rd IEEE International Workshop on Wireless Ad-hoc and Sensor Networks*, pages 761–766, New York, USA, Juni 2006.

[CPI06] Jose A. Costa, Neal Patwari, and Alfred O. Hero III. Distributed weighted-multidimensional scaling for node localization in sensor networks. In *ACM Transactions on Sensor Networks, 2(1)*, pages 39–64, February 2006.

[Kay93] Steven M. Kay. *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. Prentice Hall PTR, 1993.

[LZZ06] Juan Liu, Ying Zhang, and Feng Zhao. Robust distributed node localization with error management. In *MobiHoc '06: Proceedings of the seventh ACM international symposium on Mobile ad hoc networking and computing*, pages 250–261, New York, NY, USA, 2006. ACM Press.

[OLT04] E. Olson, J. J. Leonard, and S. Teller. Robust range-only beacon localization. In *Proceedings of Autonomous Underwater Vehicles*, 2004.

[PAICO03] N. Patwari, M. Perkins A. III, N. Correal, and R. O'Dea. Relative location estimation in wireless sensor networks. In *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, volume 51, pages 2137–2148, August 2003.

Model for On-chip Storage and Exchange Data Paths

Andreas C. Döring
IBM Research GmbH, Zurich Research Laboratory
Säumerstrasse 4
CH-8803 Rüschlikon, Switzerland
Email: ado@zurich.ibm.com

Abstract: Many architectures for on-chip busses, networks-on-chips, buffers, and similar structures have been proposed and discussed in the past. Performance and cost have mainly be evaluated by synthesis to a given technology or even complete implementation. This makes comparison between different architectures difficult because the effort gone into optimization, the design style, the quality of the gate libraries applied and similar effects blur the results. Furthermore, the results are only valuable for a short time span, as it would be increasingly difficult to evaluate an alternative architecture in an outdated technology. Therefore, in this paper a model for representing data buffer and exchange paths is presented. It allows the comparison of architectures and low-level structures with incorporation of technical characteristics such as logic depth per clock cycle, fan-in and fan-out, chip area, and dynamic power. Popular components, crossbar switch, and FIFO buffers are shown in various flavors in this model. The model leads to fundamental questions, that have so far not been discussed.

1 Introduction

The strong growth of networking in industry and households has led to a growing number of chips that integrate a switching functions. This includes also chipsets for personal computers and consumer products. Furthermore, the integration of different components into one system-on-chip (SoC) requires an interconnection structure that has similarities with switches.

Examples for this kind of structures are the Element Interconnect Bus (EIB) in IBM's Cell™ Broadband Engine [Cla06], the Blackford chipset for Intel-processor-based servers [Rad07], or the switching core of the 12-port 10-Gbps Ethernet switch from Fujitsu [Hor06]

For the implementation of switching functions a wide design space is available, exploited in products, and under discussion. In particular for SoCs, the question whether a bus, a crossbar-like structure, or a network is the best option is currently disputed. Evaluating the cost and speed of a particular architecture can be done at different abstraction levels.

For example, in [Wu02] a crossbar switch is presented, and its performance is evaluated on the basis of an implementation, in this case in the TSMC $0.25\mu\mathrm{m}$ technology. To compare these results on the same level with a different architecture, such as a multi-stage network (e.g. [Wak68]), it would be necessary to implement this alternative structure in the same technology. This would be very cumbersome and would not even be sufficient because the design style (standard cell, full custom design, and others) would impact the

resulting performance significantly. Therefore, a more abstract model is required that provides the relevant information on area, clock speed, latency, and power consumption. Such an abstract model has to neglect other aspects.

In this paper a model is presented that covers data exchange and clocked storage. It is described in detail in the next section. The data words are treated symbolically, therefore, the bandwidth of the model can be scaled by adjusting the word width. This scaling is one of the reasons why the model also neglects control of the data path. Already the control of a crossbar switch is a challenging problem and many algorithms for the problem have been published, e.g. [Tam93, Gup99]. However, the control logic is needed only once and its relative cost can be reduced by increasing the data-path width. In fact, modern architectures use wide internal switching paths, e.g.. the EIB uses several 128-bit-wide data paths. Some graphics cards use even wider busses. In consequence, the use of more efficient data paths becomes more attractive even if their control is more complex.

However, systems are normally not built on the basis of simple elements, but rather use more abstract building block such as First-In-First-Out (FIFO) buffers, random-access-memories (RAM) or crossbar switches. In Section 3 these basic building blocks are represented in the model. This will reveal any deficiencies in current implementations.

Using these building blocks, the EIB and the crossbar switch from Wu et al. [Wu02] will be analyzed on the basis of the model in Section 4. In Section 5 some open problems will be stated, the shortcomings of the model discussed, and an outlook for further studies given.

2 The Data-Transport Model

An on-chip interconnection structure serves several tasks:

1. Bridging physical distances
2. Arbitration for shared resources and scheduling of outstanding requests
3. Flow control for adjusting sender demand to receiver capabilities
4. Data transport from a set of inputs to a set of outputs
5. Data buffering to adjust temporal speed differences, availability
6. Decoding the address of the request to determine the receiver

Not all aspects are present in every case; for instance an Ethernet switch does not need address decoding. However, in traditional on-chip busses, such as IBM's CoreConnect Processor Local Bus (PLB), these aspects are found and are part of most bus transactions. For instance, distributed address decoding is found in system busses (VME, PCI, ISA, etc.) and was therefore also defined for onchip busses, where it does not offer an advantage. A network-on-chip needs to perform address decoding prior to issuing a message into the network, because otherwise the message cannot be routed.

As an example for the relative importance of 1 compared with 4, consider the Cell chip, in which 11 units are connected through the EIB. The chip measures approximately $12 \times 18.5\text{mm}^2$, hence the longest (Manhatten) distance is 40.5mm. Combined with the maximum transmission-line signal-propagation speed for this technology of 5.4ps/mm results in a maximum latency for connecting two units of about 220ps. With 1.6GHz the EIB is one of the fastest interconnect structures, the bidirectional ring imposing an average distance of three stages, or 1875ps, demonstrating the dominance of the aspects considered.

The model used in this article considers only items 4 and 5. One reason is that transport and buffering form the basis, which defines the requirements for the logic associated with arbitration and flow control. Furthermore, only these two aspects scale with wider busses.

A very simple form of a switch is the n-to-1 concentrator of n inputs to 1 output without buffering. As a circuit element it is called a multiplexer. In fact, the entire model only requires two basic elements, *multiplexers* and *(storage) registers*. As almost all switching structures involve some degree of buffering, storage elements (called registers here) are needed in the model. The reuse of storage implies some notion of sequence and time, which is represented in the model as a synchronous clock. This clock can be related to a clock in a traditional synchronous circuit design clock, but there is more freedom when implementing a data path based on a given abstract model: finer pipelining can always be introduced. It does not even have to be the same in all parts of the system as the word width visa clock speed can be adapted as required.

Two cascaded multiplexers – i.e., a circuit in which the output from one multiplexer is connected to one input of a second multiplexer – logically form a wider multiplexer. Therefore, when creating a circuit from multiplexers and registers, only one multiplexer needs to be considered between two registers.

The combination of all registers forms the state of the system at a given point in time, which can be represented by a vector. As the values are not relevant, the elements of a state vector are symbols. During operation, some items stored in the registers will be written into other registers. This transition can be represented my multiplying the vector with a matrix consisting of ones and zeroes. Taking this further, we also add another abstract variable T representing time. Therefore, the vector $(aT + b, 0)$ represents the use of two registers, where the first register contained first data word b and in the next cycle a, while the second register was not used. Typically, data exchange differs depending on the direction of the data transport and buffering. In this case, the entries of the matrix are represented by variables. When considering several cycles of operation, the entries are time-dependent functions.

As an example we consider first a five-stage shift register. Its transition matrix is $M_{SR5} := \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$.

The input can be provided by adding a vector with a new symbol in the first position. Typically, a shift register will be used, which can also hold its state in state, leading to the

transition matrix

$$M_{CSR5} := \begin{pmatrix} \eta_0 & 1-\eta_1 & 0 & 0 & 0 \\ 0 & \eta_1 & 1-\eta_2 & 0 & 0 \\ 0 & 0 & \eta_2 & 1-\eta_3 & 0 \\ 0 & 0 & 0 & \eta_3 & 1-\eta_4 \\ 0 & 0 & 0 & 0 & \eta_4 \end{pmatrix}.$$

Different control symbols are used for the individual registers, because this allows dropping inner data items. To combine input, state, and output in one matrix, the inputs $\mathbf{a}$ and the state $\mathbf{s}$, as well as the resulting state $\mathbf{s}'$ and the outputs $\mathbf{o}$ are concatenated in this order: $(\mathbf{s}', \mathbf{o}) = (\mathbf{i}, \mathbf{s})M$.

Given a model of a data path, several characteristics are of interest. First of all, the *width of the widest multiplexer (MMW)* in the system limits the clock speed. Even if the multiplexers were implemented by cascading, the resulting logic depth would not decrease. For small widths, the minimal cycle time grows at worst logarithmically with the MMW, because a tree of the best multiplexers could be used. In the current two-dimensional technologies for a large scale the cycle time will grow at best with the square root of the number of inputs, because the wiring delays will dominate. Examples for the state of the art are represented by the crossbar switch [Wu02] and the SRAM in [Zha06]. In the first example, two stages of four-input multiplexers per clock cycle at a clock speed of 1GHz in $0.25 - \mu\mathrm{m}$ technology are used.

A second relevant information from the model is the *cost*, expressed in the number and sizes of the multiplexers and registers. For this purpose, we introduce two constants, A_{FF} and A_M which represent the area of one register and one multiplexer per input. Of course, this cost is for a one-bit-wide data path; the area of wide paths is assumed to be proportional to its width. The assumption of linearly growing cost for multiplexers with respect to the number of inputs is valid for most CMOS implementations. For high-scale multiplexers such as those found in SRAMs, there also is an associated fix cost for precharge and sense amplifiers but a much lower proportional cost (one transistor per input). Therefore, when passing the tradeoff point from simpler multiplexer structures to complex ones, different factors needs to be applied.

Finally, the number of inputs driven from one register output (*fanout*) also impacts cost and speed. All three properties can be determined on the matrix representation of a datapath structure by determining the maximum number of non-zero entries per matrix row, the total number of non-zero matrix entries plus the dimension of the matrix minus the number of inputs, and the maximum number of non-zero entries per column.

To sum up all information on a model with i inputs, o outputs, r registers, and the transition matrix M, a quadruple (i, o, r, M) is used.

Computer architectures typically use a hierarchical description. Hierarchy can also be applied to the matrix representation. By representing substructures by smaller matrices, the higher levels can be represented either by using block matrices or by arithmetically combining the parts.

The two aspects of data buffering and transport can be regarded as temporal and spatial rearrangement of the input data. Whereas the spatial reordering is typically described by

permutations, or – when considering multicasts – by mappings of outputs to inputs, and is well understood, the aspect of temporal reordering not only lacks a formal model, but bounds on cost and composition of structures do not seem to be well understood.

3 Basic Building Blocks

The most important building blocks of switching structures are crossbar switches and buffers, in particular with random (SRAM) and FIFO access. A traditional crossbar $(n, m, 0, M_{CB})$ consists of n times m switches such that each input is connected with every output:

$$M_{CB} = \begin{pmatrix} \alpha_{00} & \dots & \alpha_{0m} \\ \dots & & \dots \\ \alpha_{n0} & \dots & \alpha_{nm} \end{pmatrix} \text{ with } \alpha_{ij} \in \{0, 1\}.$$

To avoid conflicts, only one entry in each column might be 1: $\forall i \sum_j \alpha_{ij} \leq 1$. Meeting this requirement is part of the control logic, which is not covered by the model. Applying the model characteristics, the MMW is n, the fanout is m, and the cost is mnA_M. To reduce the multiplexer width, pipelining as in [Wu02] can be applied, which introduces registers. For instance with two stages $\sqrt{n}$-wide multiplexers are used (assuming n is square, otherwise some minor adjustments have to be made): $(n, m, m\sqrt{n}, M_{PCB})$ with

$$M_{PCB} = \begin{pmatrix} {M'}_{CB} & 0 & \dots & 0 & \dots & 0 \\ \dots & \ddots & \dots & 0 & \dots & 0 \\ 0 & \dots & {M'}_{CB} & 0 & \dots & 0 \\ 0 & \dots & 0 & M_M & 0 & \dots \\ 0 & \dots & 0 & 0 & \ddots & 0 \\ 0 & \dots & 0 & 0 & \ddots & M_M \end{pmatrix}.$$

${M'}_{CB}$ is a $\sqrt{n} \times m$ crossbar and M_M represents a multiplexer with $\sqrt{n}$ inputs.

The cost estimation of the crossbar is too pessimistic. As has been known for a long time, a crossbar can be replaced by a multistage interconnection network, for example, the one from Waksman [Wak68]. The matrix representation of such a network is omitted here. Note that it is a decomposition into a product of matrixes. This decomposition results in a system of equations with respect to the configuration of the corresponding crossbar. The equations relate to the independent paths through the network, and to the requirement that the paths need to be disjoint. If $m = n$ is a power of two, the decomposition results in a cost of $(n \log n - n + 1)4A_M$.

Another important building block is the (static[1]) random access memory with n words capacity. As the name suggests, it allows the arbitrary read or write access to its content.

[1] Because the model assumes registers as storage elements, dynamic memory cells have to be excluded, because they keep their content only for a limited time

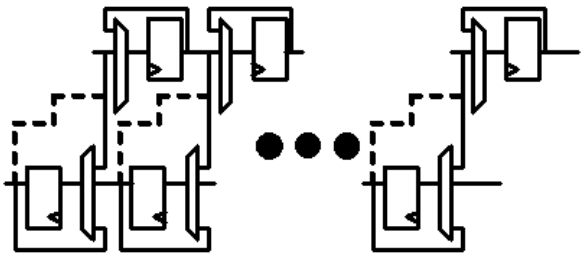

Figure 1: FIFO structure

Therefore, the MMW of an SRAM is n. Although in a typical implementation also the fanout is n it could be just 1 by implementing the write structure as a shift register. In many applications this is not practical because the control would have to keep track where each data word is located:

$$M_{RA} = \begin{pmatrix} \beta_0 & \dots & \beta_{n-1} & 0 \\ 0 & \dots & 0 & \gamma_0 \\ \dots & & \dots & \dots \\ 0 & \dots & 0 & \gamma_{n-1} \end{pmatrix}$$

$$M_{SA} = \begin{pmatrix} \kappa_0 & 1-\kappa_1 & 0 & 0 & 0 \\ 0 & \kappa_1 & 1-\kappa_2 & 0 & \gamma_0 \\ \dots & & & & \dots \\ 0 & 0 & 0 & \kappa_{n-1} & \gamma_{n-1} \end{pmatrix}.$$

A third structure is the FIFO buffer already mentioned. Whereas an SRAM can and typically is used to implement FIFO buffers, the fixed sequence of the output words allows a much faster (lower MMW) implementation, which can be found, for instance in [Dic94]Two shift registers in opposite direction with a transfer from one to the other allow an implementation with fanout and an MMW of 3, and cost of $2.5A_M + A_{FF}$ per buffer capacity. The buffer can be shared by two FIFOs through two-way communication between the two shift registers; this is shown with dotted lines in Figure 1. Sharing buffers between more than two FIFOs probably requires a higher MMW, but the optimum is unknown.

Another important structure is a ring. It has a long tradition in networks and also in on-chip busses, for instance the CoreConnect DCR-bus. The advantage of the bus is the parallel transmission on individual bus segments combined with the minimal logic depth for inserting data into a ring: (k, k, k, M_R).

$$M_R = \begin{pmatrix} \lambda_0 & 0 & \dots & 0 & \dots & 0 \\ 0 & \lambda_1 & 0 & 0 & \dots & 0 \\ 0 & \dots & \lambda_{n-1} & 0 & \dots & 0 \\ 1-\lambda_0 & 0 & 0 & 1 & \dots & 0 \\ 0 & 1-\lambda_1 & 0 & \dots & 1 & 0 \\ 0 & \dots & 1-\lambda_{n-1} & 0 & \dots & 1 \end{pmatrix}.$$

4 Analysis of Existing Systems

The EIB consists of four rings – two in each direction – and connects 11 units. For each element there is a 4-to-1 multiplexer to select the output from one of the four rings. Therefore the MMW is 4 and the total cost of the EIB is $11A_{FF} + 66A_M$. As mentioned, in 90nm SOI technology a clock frequency of 1.6 GHz is reached and 128-bit-wide data words result in a maximum bandwidth of 204.8 GByte per second. With appropriate load, this performance is actually reached; however, there are traffic pattern where the bandwidth is reduced.

As a second example consider the already mentioned crossbar switch by Wu et al. It switches 2 Gbps from each of the 256 input ports to the same number of output ports. It uses two pipeline stages as presented in Section 3. It is interesting that only two bit slices are used, using a 1 GHz clock. On the basis of transistor count, a 1-bit register with integrated 4-to-1 multiplexer costs 30 transistors. For the static multiplexer, no numbers are given, but assuming the use of a complex AND-OR-gate (such as AO2222 in the Samsung and IBM libraries in comparable technologies) it would be 16. Therefore, we can assume $A_{FF} = 14$ and $A_M = 4$. To cope with the delay implied by the high fanout of the crossbar logic, the crossbar presented is divided into four parts and another pipeline stage is added for two of these parts. The inputs are registered at least once, which results in 768 two-bit registers at the inputs and outputs. Because of the division into four 128×128 sub-arrays, the two pipeline stages are not balanced, one implements 8-to-1 multiplexers, the other 16-to-1. Surprisingly, the authors choose to put the smaller multiplexers into the earlier stage, which results in a higher latency. Therefore, the design has 4096 two-bit pipeline registers. The total cost of the pipelined design chosen is therefore $2(768 + 4096)A_{FF} + 2 \cdot 256 \cdot 256A_M = 660'480$. A non-pipelined, four-bit wide design (with registers at the inputs and outputs) would cost $4 \cdot 512A_{FF} + 4 \cdot 256 \cdot 256A_M = 1'077'248$ about a third more expensive. Their paper only gives numbers on a much smaller 64×64 crossbar chip.

5 Summary

The abstract data path model presented here is quite simple, which allows an easy design space exploration. When comparing the results of the model with technical implementations, SRAMs in particular show a significant difference because they use pass transistors

in a bidirectional way, i.e., one and the same transistor is used for writing and reading. Whereas for reading the signal flows away from the storage cell, for writing the opposite is the case. However, when studying more recent SRAM implementations, one can observe that the word lines are getting shorter, and traditional unidirectional multiplexers are used to handle the read data separately from the write data. Furthermore, the model ignores layout questions, which can be seen in the crossbar chip example, where the entire chip serves the interconnection problem.

There is a set of open questions based on this model. First of all, for many problems solutions with minimal cost, minimal MMW or a combination of them are of interest. In particular, a set of FIFO buffers which share most of the storage resources is interesting for many high-speed networking applications. Related to that is the support of virtual channels or virtual output queueing in a buffer structure with common input and output. PCI-Express [Bud05] adds several traffic classes on top of this with specific ordering rules. This results in the requirement that typically the order within each queue be maintained, and only in very rare situations do data items from the middle of a particular queue have to be extracted. Initial studies on structures indicate that the time span between the knowledge about a required data item and the actual point when it has to be provided at an output has a considerable impact on the cost and speed of a buffering and transport datapath. Furthermore, for structures with shared resources the granularity of sharing seems to have an impact on the cost. In the past, structures like these were implemented with regular SRAM, whose capabilities are not fully exploited, but which results in a speed impact due to capacity scaling caused by the capacity dependence of the MMW.

It is also interesting to relate the model to the structures present in programmable logic devices such as FPGAs. Whereas the implementation of multiplexers with the typical Look-up-Table structure is not very efficient compared with other instances of combinational circuits, many FPGAs provide extra resources such as larger memory blocks, shift registers, or carry chains which represent basic building blocks on a coarser granularity. In the other direction, the considerations of adding switched interconnection resources as complement to the typical static wiring in programmable logic devices have been discussed for some time and this is apparently unavoidable for future device scaling. Results from the model with respect to optimal structures for certain requirements could be applied in this domain.

Finally, the formal aspects on the model should be extended by mechanisms to describe sequences of data words and their manipulation. For instance, intuitively a shared buffer for two logical queues is certainly restricted compared with an SRAM buffer, because only items from different queues can overtake each other. How can this difference be formally described, can we provide a quantitative judgement?

References

[Bud05] D.; Shanley T Budruk, E.;Anderson. *PCI Express System Architecture*. Addison Wesley, 2005.

[Cla06] Ch.; Krolak D. Clark, S.; Johns. Single Port/Multiple Ring Implementation of a Data Switch. In *Patent application World International Property Organization*, number WO 2006/095838 A3, September 2006. available at www.delphion.com.

[Dic94] S. Dickey. *Systolic Combining Switch Designs*. PhD thesis, New York Univeristy, 1994.

[Gup99] N. Gupta, P.; McKeown. Design and Implementation of a Fast Crossbar Scheduler. In *IEEE Micro*, volume 19(1), pages 20–28, Jan-Feb 1999.

[Hor06] T.; Hattori A. Horie, T.; Shimizu. Single-Chip, 10-Gigabit Ethernet Switch LSI. In *Fujitsu Scientific & Technical Journal*, volume 42(2), pages 206–213, April 2006.

[Rad07] S.; Cheng K. Radhakrishnan, S.; Chinthamani. The Blackford Northbridge Chipset for the Intel 5000. In *IEEE Micro*, volume 27(2), pages 22–33, 2007.

[Tam93] H.C. Tamir, Y.; Chi. Symmetric Crossbar Arbiters for VLSI Communication Switches. In *IEEE Transactions on Parallel and Distributed Systems*, volume 4(1), pages 13–27, 1993.

[Wak68] A. Waksman. A Permutation Network. In *Journal of the ACM*, volume 15(1), page 159163, 1968.

[Wu02] C.-Y.; Hamdi M. Wu, T.; Tsui. A 2Gb/s 256*256 CMOS Crossbar Switch Fabric Core Design Using Pipelined MUX. In *IEEE International Symposium on Circuits and Systems, ISCAS*, pages 568–571, 2002.

[Zha06] U.; Chen Z. et al. Zhang, K.; Bhattacharya. A 3-GHz 70Mb SRAM in 65-nm CMOS Technology With Integrated Column-Based Dynamic Power Supply. In *IEEE Journal of Solid-State Circuits*, volume 41(1), pages 146–151, 2006.