



# Wintersemester 2011/2012

## Seminare

**Bachelor Informatik CS 3702**  
**„Datenbanken und Anfrageverarbeitung“**

**Master Informatik**  
**„Advanced Topics of Database Systems“**

**CS 5840 - Fachübergreifende Kompetenzen = englischsprachiges Seminar**

**CS 5480 - für den Bereich Software Systems Engineering = Seminar  
Software Systems Engineering**

Dr. rer. nat. Sven Groppe  
[groppe@ifis.uni-luebeck.de](mailto:groppe@ifis.uni-luebeck.de)  
Dipl.-Inf. Stefan Werner  
[werner@ifis.uni-luebeck.de](mailto:werner@ifis.uni-luebeck.de)

# Students' Duties

- Preparation of Slides
- Preparation of Handout
  - 2 to 3 pages, to be delivered to all participants and to the supervisors directly before the presentation
- Presentation
  - Approx. 1 hour (inclusive discussion)
- Attending presentations of all other students
  - Contributions to a lively discussion after each presentation

# Timeline

participating in all other presentations  
and contributing to lively discussions



Assign-  
ment  
of topics

discussion  
with  
supervisor  
about  
presentation

electronic  
submission  
of pdfs of  
presentation  
and handout  
via Email  
(final from  
student's view)

improving  
presentation  
and handout  
according to  
supervisor's  
comments  
and remarks

presentation,  
electronic  
submission  
of source files  
and pdfs of  
slides and  
handout to  
the supervisor  
via Email

# Topics Bachelor/Master Seminars

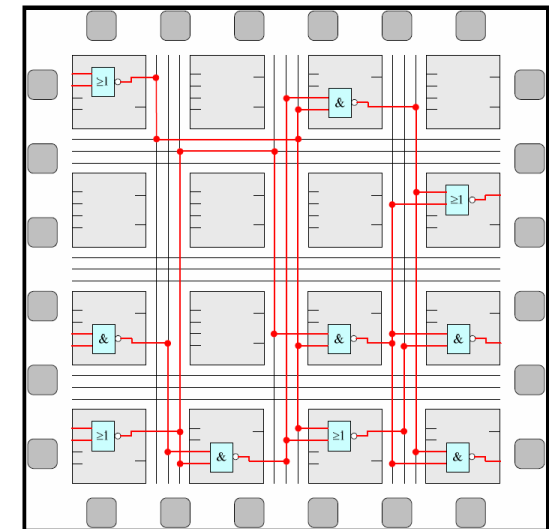
- Green topics -> seminar for bachelor students
- Blue topics -> seminar for master students
- Topics may be exchanged to be discussed with the supervisors
  - Bachelor students choosing blue topics
  - Master students choosing green topics

# Overview Topics

- **FPGA**
- Relational Databases
- Cloud Computing
- Graph Databases
- Semantic Web
- Key-Value Stores
- Join Processing
- **Main-Memory Databases**
- **Parallel Processing**
- **Datalog/Prolog**

# FPGA (**F**ield **P**rogrammable **G**ate **A**rrays)

- integrated circuit (IC)
- configuration after manufacturing
- hardware description language (HDL)
- complex functional blocks,  
arranged in periodic structure
- interconnection network



# FPGA (Field Programmable Gate Arrays)

- Advantages towards General CPUs
  - inherent parallelism
  - heavy throughput at low clock rate
  - high energy efficiency
- Reconfigurable computing = offload „expensive“ tasks from software to FPGA

# FPGA (**F**ield **P**rogrammable **G**ate **A**rrays)

## - Topics -

- **Data Processing on FPGAs**
- **Multi-Core Architecture on FPGA for Large Dictionary String Matching**
- **High Throughput and Large Capacity Pipelined Dynamic Search Tree on FPGA**
- **Massively Parallel XML Twig Filtering Using Dynamic Programming on FPGAs**
- **FPGASort: A High Performance Sorting Architecture Exploiting Run-time Reconfiguration on FPGAs for Large Problem Sorting**
- **How Soccer Players Would do Stream Joins**



# Relational Database

- Most widely used type of database
- Data model
  - Relation (table)

<b>Name</b>	<b>ID</b>	<b>Adress</b>
student1	1	HL
student2	2	HH

<b>ID</b>	<b>Lecture-ID</b>
1	DB1
2	MVDB2

- Query language
  - SQL

# Relational Database - Topics

- **Automatic Discovery of Attributes in Relational Databases**
  - data oriented solution that use statistical measures to identify strong relationships between the values of a set of columns
- **Spreadsheet As a Relational Database Engine**
  - virtually any spreadsheet software (like Microsoft Excel) is a relational database engine
- **Workload-Aware Database Monitoring and Consolidation**
  - aims to minimize the number of servers and balance load, while achieving near-zero performance degradation
  - a factor of 12 higher throughput compared to virtual machines (VM) running database servers
- **Oracle Database Filesystem**
  - consolidated solution that unifies state-of-the-art network filesystem features with relational database management ones
  - novel shared-storage network filesystem developed in the RDBMS kernel that allows content management applications to transparently store and organize files using standard filesystem interfaces, in the same database that stores associated relational content

# Cloud Computing

- typically, cloud provider offers online-services
- multiple server-based computational resources via a digital network like internet
- applications are provided and managed by the cloud server
- data is stored remotely in the cloud configuration



## Cloud Computing

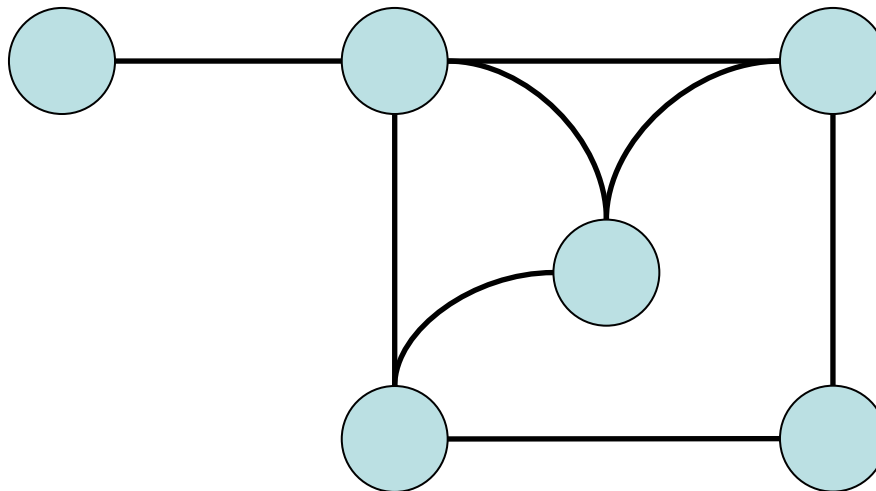
*Having secure access to all your applications and data from any network device*

# Cloud computing - Topics

- **Schedule Optimization for Data Processing Flows on the Cloud**
  - scheduling of dataflows in the context of three different problems:
    1. minimize completion time given a fixed budget,
    2. minimize monetary cost given a deadline, and
    3. find trade-offs between completion time and monetary cost
- **Zephyr: Live Migration in Shared Nothing Databases for Elastic Cloud Platforms**
  - Live migration: technique to migrate tenants with minimal service interruption and no downtime (here migration of a live database)
- **An Evaluation of Alternative Architectures for Transaction Processing in the Cloud**
  - alternative architectures to effect cloud computing for database applications and reports on the results of a comprehensive evaluation of existing commercial cloud services that have adopted these architectures
- **Apache Hadoop Goes Realtime at Facebook**
  - This paper describes the reasons why Facebook chose Hadoop and HBase over other systems such as Apache Cassandra and Voldemort and
  - discusses the application's requirements for consistency, availability, partition tolerance, data model and scalability

# Graph Databases

- powerful tool for
  - storing, accessing and visualizing graphs
  - graph-like queries, for example computing the shortest path between two nodes



# Graph Databases - Topics

- **TEDI: Efficient Shortest Path Query Answering on Graphs**
  - indexing and query processing scheme for the shortest path query answering
- **Computing Label-Constraint Reachability in Graph Databases**
  - *Can vertex  $u$  reach vertex  $v$  through a path whose edge labels are constrained by a set of labels?*

# Semantic Web

- Idea
  - "web of data" that enables machines to understand the semantics, or meaning, of information on the World Wide Web.
  - extends the network of hyperlinked human-readable web pages by inserting machine-readable metadata about pages and how they are related to each other
  - enables automated agents to access the Web more intelligently and perform tasks on behalf of users
- Semantic Web databases can be seen as graph databases for labelled and directed graphs

# Semantic Web - Topics

- **Apples and Oranges: A Comparison of RDF Benchmarks and Real RDF Datasets**
  - comparison of data generated with existing RDF benchmarks and data found in real RDF datasets
- **Efficient Query Answering in Probabilistic RDF Graphs**
  - model RDF data by probabilistic graphs
  - RDF query is equivalent to a search over subgraphs of probabilistic graphs that have high probabilities to match with a given query graph



# Key-Value Stores

- **One of the NoSQL database types**
- **Stores and retrieves key-value-pairs**

Name       -> „Harry“  
City        -> „HL“  
Street     -> „Castle Street“

...

- **SkimpyStash: RAM Space Skimpy Key-Value Store on Flash-based Storage**
  - low RAM footprint at about 1 ( $\pm 0.5$ ) byte per key-value pair

# Join Processing

R	
first	lang
„hallo“	de
„hello“	en
„hi“	en
„salut“	fr
„hola“	sp
„ciao“	it
„nihao“	ch

 $\bowtie$ 

S	
lang	second
de	„Student“
en	„collegian“
en	„student“
fr	„edutiant“
it	„universitario“
it	„studente“
ch	daxuesheng

=

R $\bowtie$ S		
lang	first	second
de	„hallo“	„Student“
en	„hello“	„collegian“
en	„hello“	„student“
en	„hi“	„student“
en	„hi“	„collegian“
fr	„salut“	„edutiant“
it	„ciao“	„universitario“
it	„ciao“	„studente“
ch	„nihao“	„daxuesheng“

**Join: set of all combinations of tuples in two relations that are equal on their common attribute names**

# Join Processing - Topics

- **Design and Evaluation of Main Memory Hash Join Algorithms for Multi-core CPUs**
  - (hash) join algorithms in main memory for multi-core CPUs
- **Query Optimization Techniques for Partitioned Tables**
  - efficient plans for SQL queries involving joins over partitioned tables

# Main-Memory Databases

- Main memory is cheap today
  - => Server with large main memory
  - => Indices of databases fit into main memory for typical datasets used in companies
  - => faster query processing/updating
- Usage of disks only to make data persistent

# Main-Memory Databases - Topics

- **Low Overhead Concurrency Control for Partitioned Main Memory Databases**
  - Comparison of two low overhead concurrency control schemes that allow partitions to work on other transactions during network stalls, yet have little cost in the common case when concurrency is not needed

# Parallel Processing

- **Modern computers are multi-core computers and parallel computations can be done in GPUs**  
=> parallel computers  
=> use parallel computers to speed up processing
- **Fast Sort on CPUs and GPUs: A Case for Bandwidth Oblivious SIMD Sort**

# Datalog/Prolog

- **query and rule language for deductive databases**
- **subset of Prolog**
- **Example:**
  - Facts:**           parent(bill,mary).  
                      parent(mary,john).
  - Rules:**            ancestor(X,Y) :- parent(X,Y).  
                      ancestor(X,Y) :- parent(X,Z), ancestor(Z,Y).
  - Query:**           ?- ancestor(bill,X).
- **More Efficient Datalog Queries: Subsumptive Tabling Beats Magic Sets**
  - top-down evaluation method with more reuse of answers than the dominant tabling strategy

- **FPGA**
  - Data Processing on FPGAs
  - Multi-Core Architecture on FPGA for Large Dictionary String Matching
  - High Throughput and Large Capacity Pipelined Dynamic Search Tree on FPGA
  - Massively Parallel XML Twig Filtering Using Dynamic Programming on FPGAs
  - FPGASort: A High Performance Sorting Architecture Exploiting Run-time Reconfiguration on FPGAs for Large Problem Sorting
  - How Soccer Players Would do Stream Joins
- **Relational Databases**
  - Automatic Discovery of Attributes in Relational Databases
  - Spreadsheet As a Relational Database Engine
  - Workload-Aware Database Monitoring and Consolidation
  - Oracle Database Filesystem
- **Cloud Computing**
  - Schedule Optimization for Data Processing Flows on the Cloud
  - Zephyr: Live Migration in Shared Nothing Databases for Elastic Cloud Platforms
  - An Evaluation of Alternative Architectures for Transaction Processing in the Cloud
  - Apache Hadoop Goes Realtime at Facebook
- **Graph Databases**
  - TEDI: Efficient Shortest Path Query Answering on Graphs
  - Computing Label-Constraint Reachability in Graph Databases
- **Semantic Web**
  - Apples and Oranges: A Comparison of RDF Benchmarks and Real RDF Datasets
  - Efficient Query Answering in Probabilistic RDF Graphs
- **Key-Value Stores**
  - SkimpyStash: RAM Space Skimpy Key-Value Store on Flash-based Storage
- **Join Processing**
  - Design and Evaluation of Main Memory Hash Join Algorithms for Multi-core CPUs
  - Query Optimization Techniques for Partitioned Tables
- **Main-Memory Databases**
  - Low Overhead Concurrency Control for Partitioned Main Memory Databases
- **Parallel Processing**
  - Fast Sort on CPUs and GPUs: A Case for Bandwidth Oblivious SIMD Sort
- **Datalog/Prolog**
  - More Efficient Datalog Queries: Subsumptive Tabling Beats Magic Sets