Model-based Database Systems

Kasun S Perera

Research Progress Report

Supervisors

TU Dresden – Prof. Wolfgang Lehner

Aalborg University – Prof. Torben Bach Pedersen

Introduction & Motivation

Dresden Database Systems Group

Definition 1: Model

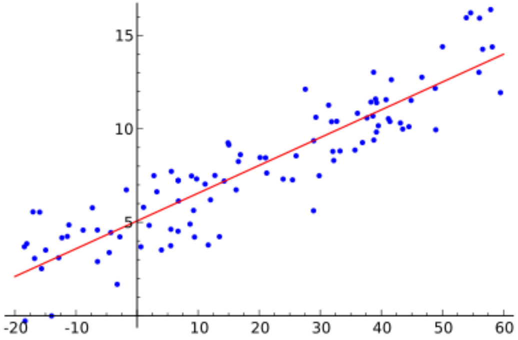
- A model is a representation, generally a simplified description, especially a mathematical one, of a system or a process to assist in calculation and predictions. ~ Oxford Dictionary

Why Models ?

- Approximate representation of underlying data
- Produce approximate results for decision making process
- Low memory footprint
- Query execution directly over model domain without regenerating data

Business Intelligence

- Querying large amount of data
- Extract information rather than querying individual data points
- Faster Approximate Results Vs Slower Exact Results



Select City,Phone,Color,AVG(Sales)
From tbl_Sales
Where City = "Barcelona" **And** Phone = "iPhone-5S" **And** Color="Black"
Date **Between** "Jan-2014" **And** Mar-2014

Exact: 1134 Units
Time : 5 mins

Approximate: 1100 Units
Time : 1 mins

Agenda

MODEL-BASED DATABASE SYSTEM

PROPOSED QUERY SYNTAX

WORKING WITH SINGLE DIMENSION TIME SERIES

MULTI-DIMENSIONAL TIME SERIES DATA

UPDATED PHD PLAN

Model-based Database System

Dresden Database Systems Group

Single Database/Data Warehouse

- Model Storage
- Data Storage

Query Results

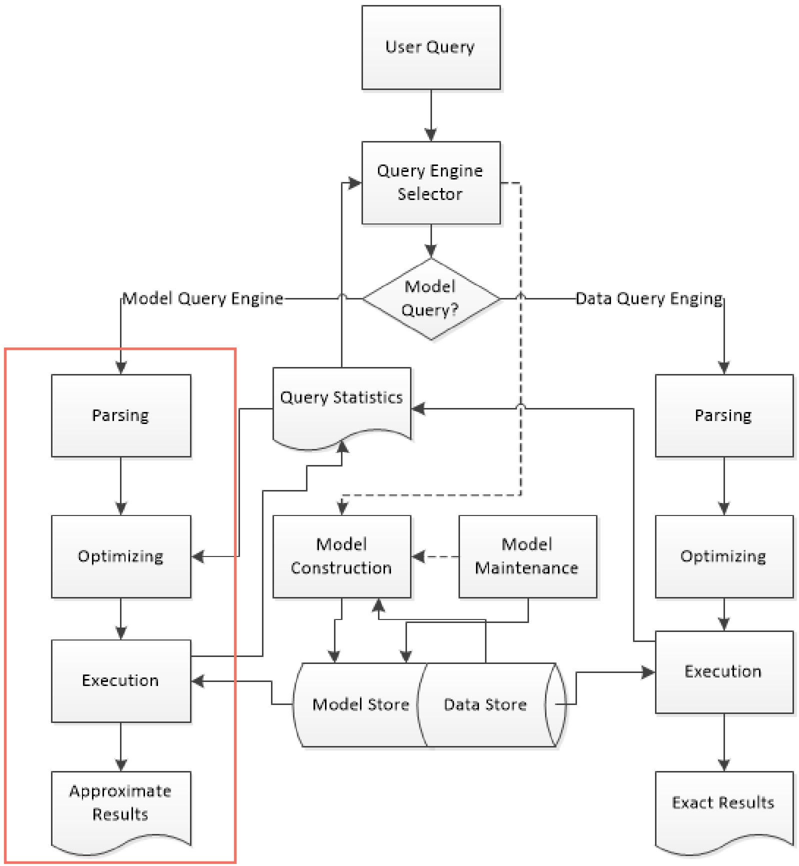
- Slow Exact Queries
- Faster Approximate Queries

Query Engines

- Traditional Query Processor
- Model Query Processor

Model Query Processor

- Parsing
- Model Selection
- Optimizing wrt Models
- Query Execution over Models



Proposed Query Syntax

SELECT CITY,PRODUCT, AVG(SALES)

FROM TBL_SALES

WHERE CITY="BARCELONA" AND PRODUCT = "IPHONE" AND DATE BETWEEN '01-01-2013' AND '31-12-2013'

USE MODEL MODELCATEGORY

ERROR WITHIN 10%

RUNTIME WITHIN 5 SECONDS

User can selects which models to use

User defines his/her expected maximum runtime for the given query

User defines his/her desired maximum error bound for the given query

Singular Time Series

TECHNISCHE UNIVERSITÄT DRESDEN

Paper 01 – Modeling Time Series Data

Kasun S. Perera, Martin Hahmann, Wolfgang Lehner, Torben Bach Pedersen, Christian Thomsen, "Modeling Large Time Series for Efficient Approximate Query Proccesing", 2nd International workshop on Big Data Management and Service (BDMS 2015) DASFAA 2015, Hanoi, Vietnam

Time Series

- $TS = (t_1, v_1)$, (t_2, v_2),.., (t_n, v_n)
- $\sum(TS) = \sigma(ts_1) + \sigma(ts_2) + .. + \sigma(ts_m)$

Model Construction

- Partitioning to preserve local trend
- Modeling partitions
- Final model is a collection of partition models

Querying over Model

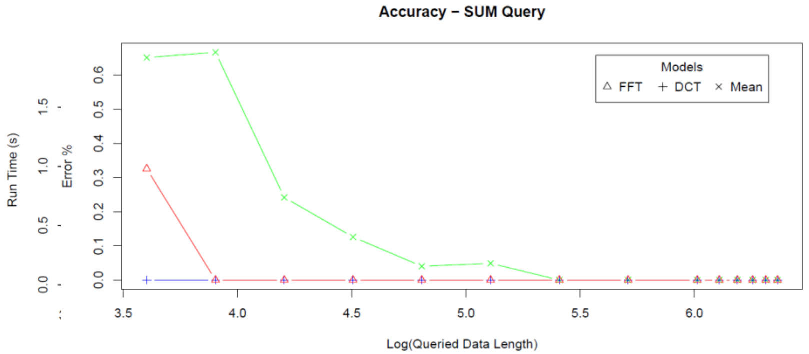
$$Q_{SUM} = \sum_{n=SP\%C}^{C} \frac{\sum_{i=0}^{C} \sigma(ts_{\lfloor SP/C \rfloor})[i] e^{-i2\pi k \frac{n}{C}}}{C} + \sum_{p=\lceil SP/C \rceil}^{\lfloor EP/C \rfloor} \sigma(ts_p)[1] + \sum_{n=1}^{EP\%C} \frac{\sum_{i=0}^{C} \sigma(ts_{\lceil EP/C \rceil})[i] e^{-i2\pi k \frac{n}{C}}}{C}$$

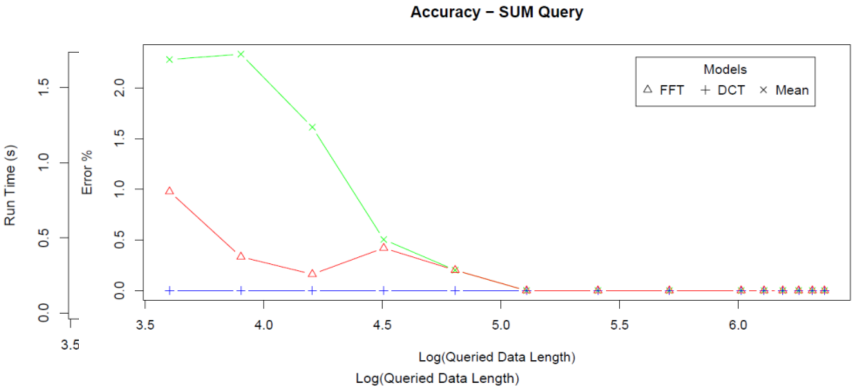
2,x3...xm]

Evaluation

RUN TIME VS QUERIED DATA LENGTH

ACCURACY VS QUERIED DATA LENGTH





Problems faced

Compression over single time series

- Local Trend Vs Global Trend
- Seasonal patterns partitioned to separate partitions

Querying multiple Time Series

- Aggregation dimensions

BI questions

- Aggregation over million points Vs analyzing local trend

Multi-dimensional Time Series

Paper 2 – Querying Multi-dimensional Time Series Data

Dresden Database Systems Group

CASE STUDY:

- Germany Consumer Information
  - BI Question : What is the average sales of 500L Refrigerators of a given brand in Saxony state during summer season.
  - Aggregation - Average Query
- IRISH Electricity Consumption Survey
  - BI Question : What is the total energy consumption of a household in Dublin with a size of $50m^2$ and having an average income of 2000 GBP
  - Aggregation – Sum Query
- Danish Wind Energy Production
  - BI Question : Which turbines of a given area shows energy production patterns different to the common acceptable pattern
  - Similarity/dissimilarity Query
- Potential
  - Produce results for these queries need aggregation over a large dataset and comparison of multiple time series, which requires sufficiently large time on RDBMS. But users willing to accept approximate results.
  - Model-based system provides faster but approximate results

Grouping

Dresden Database Systems Group

MINIMIZES DATA ACCESS

Month	Outlet	Brand	Color	Sales_Units
March-15	A	Samsung	Blue	16
March-15	B	Samsung	Black	170
April-15	A	HTC	White	12
March-15	A	HTC	Blue	6
April-15	B	Nokia	Black	80

Similarity Based Grouping

Context Based Similarity - CBS

Time series for any aggregation level

Group time series based on the distance measure calculated over the participating **dimensions**

Ex: [(A, Samsung, Blue),(A, HTC, Blue)] and [(B, Nokia, Black)]

Aggregation over the dimension values

Value Based Similarity - VBS

Time series for lowest aggregation level possible

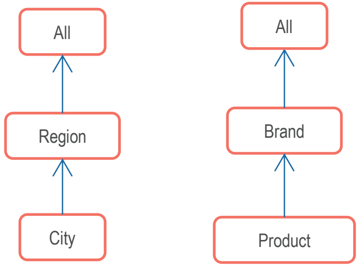
Group time series based on the distance measure calculated over the **measured values**

Reference Time Series per group + Outliers

Aggregate to build over Reference Time Series

Aggregation

Hierarchy in DB



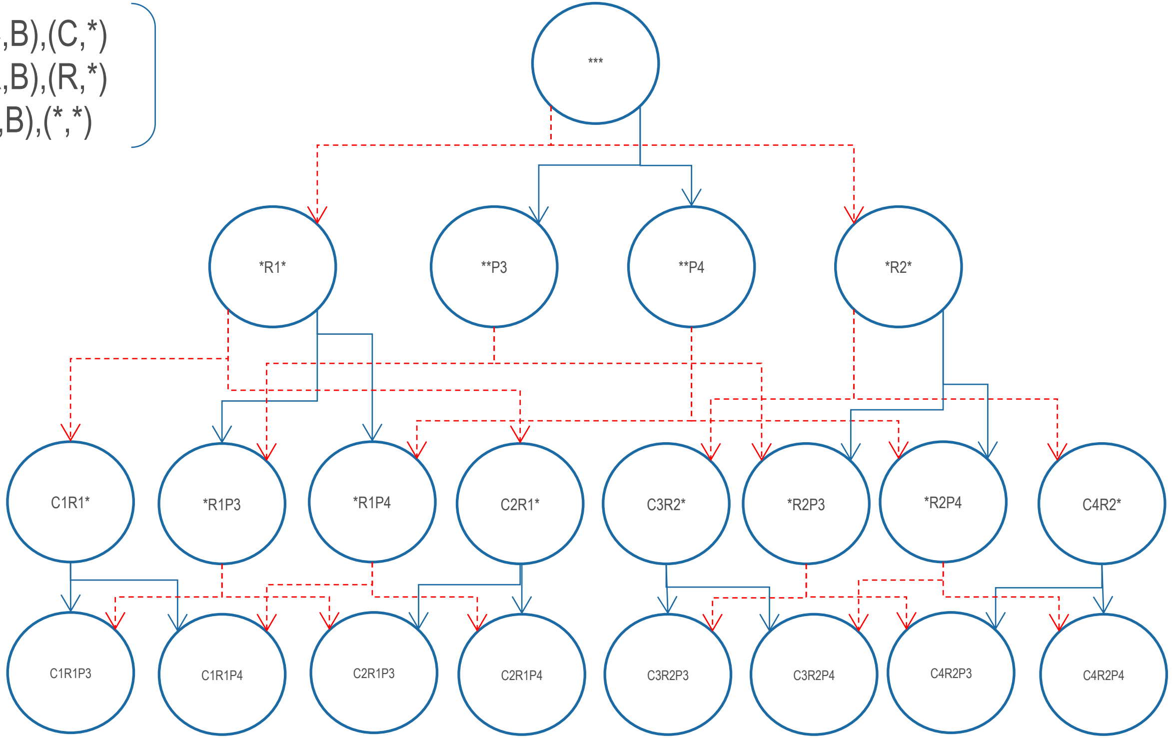
Aggregation levels

$$\left\{ \begin{matrix} (C,P),(C,B),(C,*) \\ (R,P),(R,B),(R,*) \\ (*,P),(*,B),(*,*) \end{matrix} \right\}$$

Example

- Product
  - P3,P4
- City
  - C1,C2,C3,C4
- Region
  - R1,R1
- Dependency
  - C1,C2 –> R1
  - C3,C4 –> R2

Example



Top Down Disaggregation - CBS

Dresden Database Systems Group

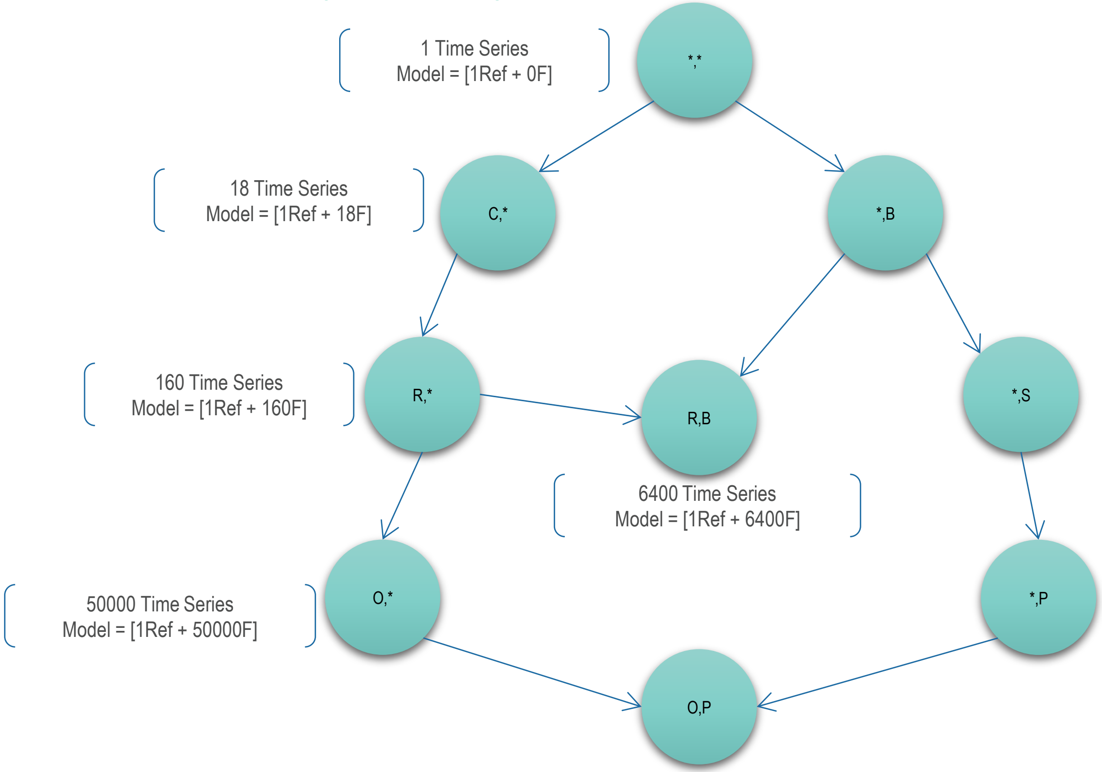
Example Data Distribution

Dimension$_1$	Dimension$_2$
D_1 – ALL (1)	D_2 – ALL (1)
C – Country (18)	B – Brand (40)
R – Region (160)	S – Series (250)
O – Outlet (50000)	P – Product (3000)
O,P – (9,000,000)	

Aggregation Levels

- 16 possibilities

Model Hierarchy Example



Top Down Disaggregation

Dresden Database Systems Group

DERIVE REFERENCE TIME SERIES ?

FACTOR CALCULATION ?

SINGLE REF_{TS} VS MULITPLE REF_{TS} ?

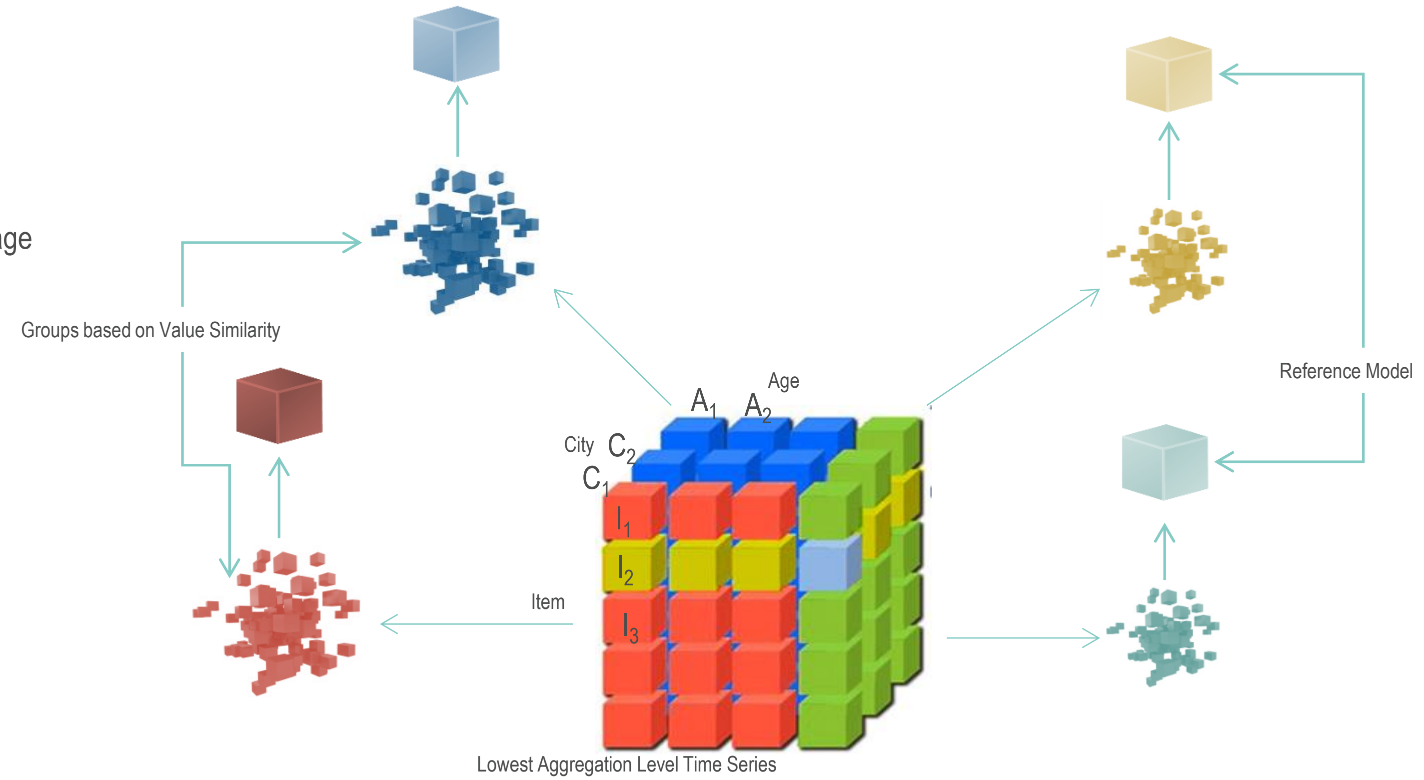
MULTIPLE FACTORS FOR A SINGLE TIME SERIES ?

DIRECT AND INDIRECT MODELS

Bottom Up Approach - VBS

QUERIES

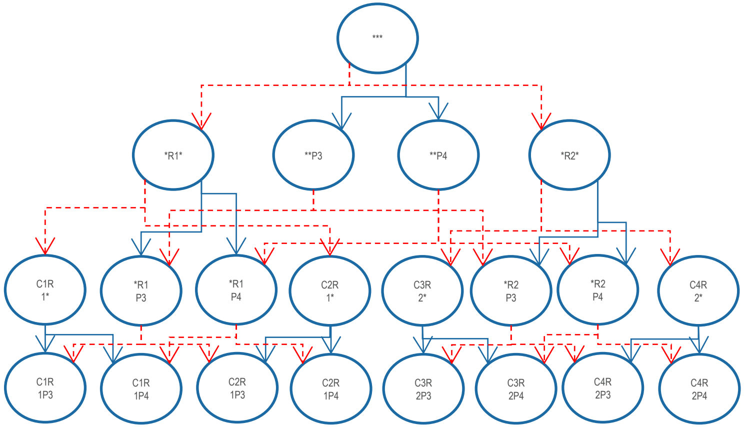
- Sales of a given item
  - I_1 (Red,Blue,Green)
- Sales of given item,city
  - I_1,C_1 (Red,Green)
- Sales of a given item,city,age
  - I_1,C_1,A_1 (Red)
- Roll-Up
  - City to Regions
  - C_1,C_2 -> R1
  - Sales of R_1,I_1

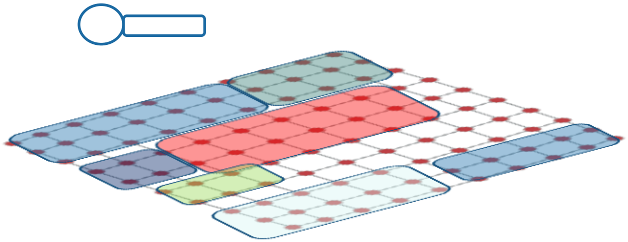


Bottom Up Approach

Lowest Aggregation Level

- Any upward aggregation is possible
- Detailed patterns
- Larger groups
- Model
  - Reference Time Series + Outliers
- Performance Gain
  - $N_{groups} <<< N_{timeseries}$
  - Space and I/O
  - Cache models in memory
- Given a Query in Higher Aggregation Level
  - $N_{groupstoread} <= N_{participatingtimeseries}$
- Objective Function
  - $M(S) = W_1[Ref_{(TS)} + Outliers_{(TS1..TSn)}] + W_2[Error\ Bound]$





Optimization in Model Domain – Future Work

Dresden Database Systems Group

Pros and Cons of Two methods

- Aggregation Upwards
- Disaggregation Downwards

Top-down and Bottom-Up combined

When to use which

Grouping in disaggregation method (CBS)

Multiple Models at a Given Aggregation

Updated Schedule

Dresden Database
Systems Group

Milestone	Description	Date
Paper 01	Efficient Approximate Query processing for large time series BDMS workshop at DASFAA 2015	15 Dec 2014
Paper 02	Querying multi-dimensional time series using representative models EDBT 2016 Conference	Sep 2015
Paper 03	Query analysis and optimization in model-based database systems TODS Journal	Dec/Jan 2015
Paper 04	ModDB : Model Based Database Management System Demo Paper	Feb 2016
Paper 05	Model indexing and maintenance in ModDB CIKM 2016 Conference	May 2016

Paper 03

Dresden Database Systems Group

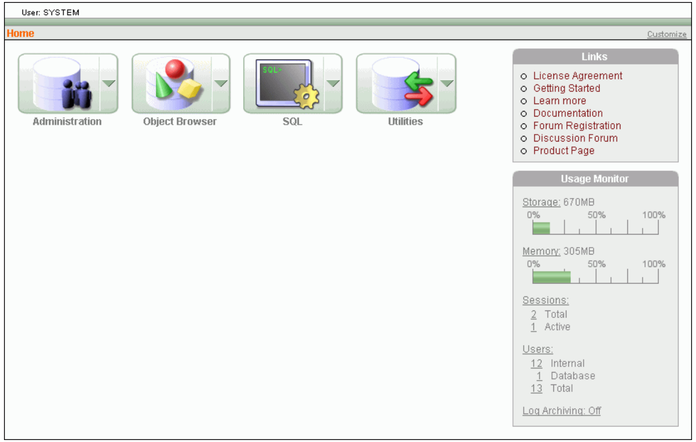
Query and Model optimization in model-based database systems

- Query parsing and analysis to derive participation models
- Selecting best possible candidate models from model pool to given user parameters for better results
- Optimize model pool by combining set of models for better performance
  - Direct models and Indirect Models
- Thorough evaluation of the system using real world use cases

Paper 04

ModDB : Model Based Database Management System

- PostgreSQL integration
- Offline Model Generation
- Model evaluation against user defined parameters
- Querying direct models
- Querying indirect models



Paper 05

Dresden Database Systems Group

Model indexing and maintenance in ModDB

- Indexing direct models
- Indexing for indirect models
- Indexing multiple models per aggregation level
- Updating models
  - Scheduled updates
  - Update on demand

Courses

Dresden Database
Systems Group

Courses	Place	ECTS	General/Project/Informal	Status
Foreign Language (German)	TUD	2.5	General	Completed Winter 2013
Transactional Information Systems	TUD	6.0	Project	Completed, Winter 2013
Database Seminar	TUD	3.0	Project	Completed, Summer 2014
European Business Intelligence Summer School	Berlin, Germany	2.0	Project	Completed, July 2014
ECML-PKDD Conference Participation	Nancy, France	1.0	Informal	Completed, 15-19 Sep 2014
Modern Analytical Database Technology	AAU	2.0	Project	Completed, Oct 2014
Patenting, commercialization and entrepreneurship	AAU	1.0	General	Completed, Fall 2014
Study Circle - Spatio-Temporal Database Systems	AAU	2.0	Project	Completed, Fall 2014
Introduction to the PhD study	AAU	1.0	General	Completed, Spring 2015
Data Science: Systems and Concepts	AAU	2.0	Project	Completed, Spring 2015
Writing and Reviewing Scientific Papers	AAU	3.75	General	Completed, Spring 2015
IT4BI-DC Doctoral Colloquium	Barcelona	3.0	Project	Enrolled, Summer 2015
Foreign Language (Danish)	AAU	2.5	General	Planned, Spring 2015
Conference attendance	To be decided	2.0	Informal	Planned, Fall 2015
General Courses		**10.75**		
Project Courses		**20.00**		
Informal Courses		**3.00**		
Total		**33.75**		
Completed		*29.25*		
Remaining		*4.5*		

Dresden Database Systems Group

Model-based Database Systems

Kasun S Perera

Research Progress Report

Related Work

BlinkDB: Queries with Bounded Errors and Bounded Response Times on Very Large Data [S. Agarwal et. al.]

Approximate Query Processing Using Wavelets [K Chakrabarti et. al.]