



An Integrated Approach to Performance Monitoring for Autonomous Tuning

INGRES

Alexander Thiem
Kai-Uwe Sattler

Agenda

- **Introduction**
- **Concept**
- **Implementation**
- **Evaluation**
- **Outlook**



Introduction

INGRES

Motivation

- **Staffing costs take 60% of TCO**
 - **Software is getting complexer**
 - **Data volumes are getting bigger**
 - **Software and database designs are getting automated**
 - **Systems get embedded**
- Maintenance must become easier**

Why Monitoring?

- **No common answer to every problem**
 - **Each setup is different**
 - **Setups change over time**
- Constant monitoring required to keep up with demands**

Why Integrated?

- **Lower overhead**
 - Less communication
- **Accurate information**
 - High data resolution / no blind spots
- **Quicker reactions**
 - 'Realtime' monitoring
- **Specialized on one system**
 - Knows every quirk of the system

Related approaches

- **Oracle Automatic Tuning Advisor**
 - Monitoring through 1h snapshots
 - Analysis mechanisms integrated into optimizer
- **SQLCM**
 - Integrated monitoring component
 - Performs internal/automatic changes
- **DB2 Event / Snapshot Monitor**
 - Internal monitor/log facility
 - Data available over SQL

Ingres' History

1974 ← **Michael Stonebraker, Eugene Wong**
1975
1976
1977
1978
1979
1980 ← **Relational Technology, Inc.**
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990 ← **ASK Group, Inc.**
1991
1992
1993
1994 ← **Computer Associates, Inc.**
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005 ← **Ingres Corporation**
2006



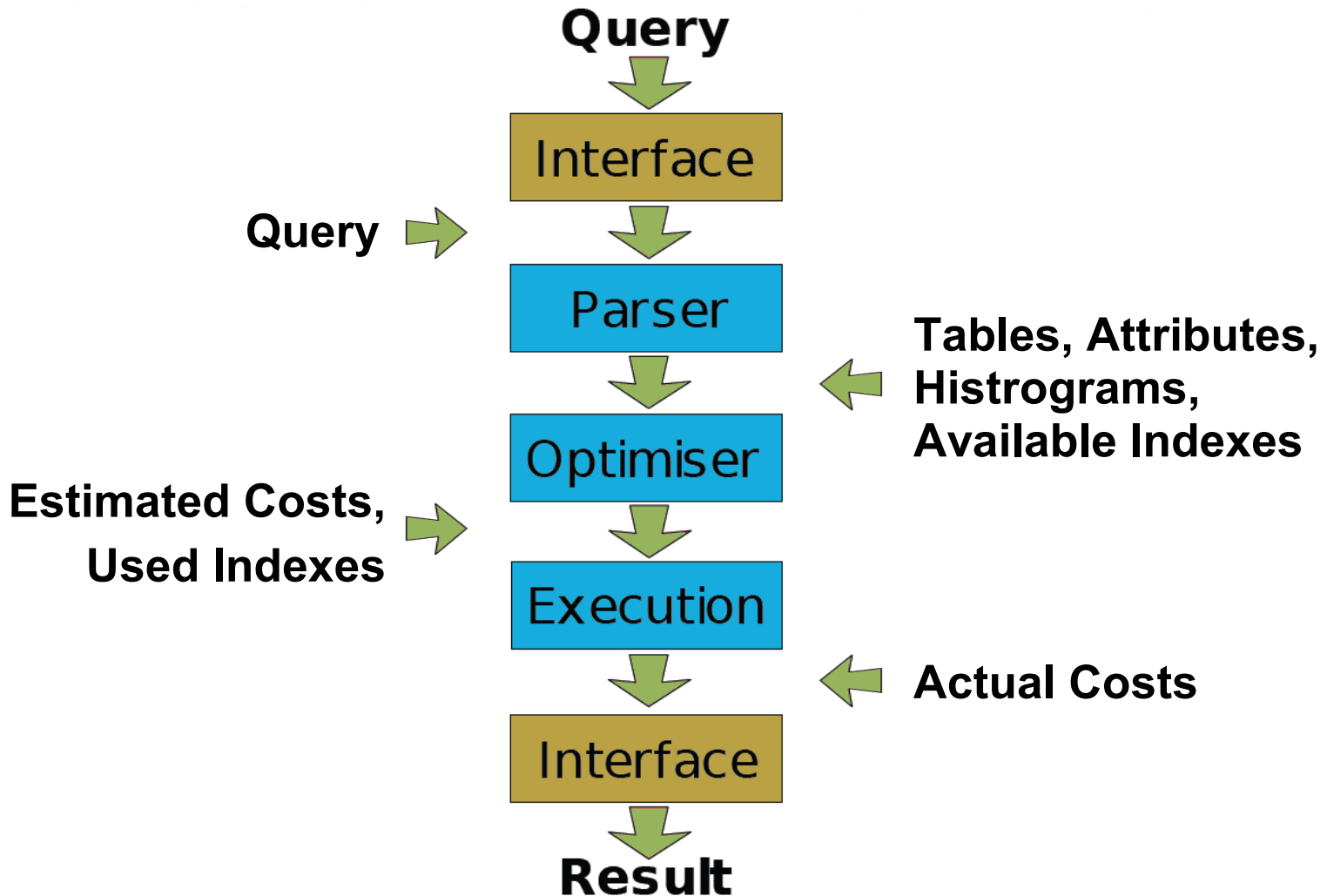
Ingres Today

- **Ingres Corp.**
 - 14 Offices + people on 5 continents
 - Headquarter in Redwood City, CA
- **Ingres Database**
 - Currently at version 9.2, 9.3 in QA
 - GPLv2 and commercial license
 - Primarily available on Linux, Windows
 - ~16 additional platforms supported



Concept

Where to Monitor?



What to Monitor?

Statements

(Database name,
Unique hash key,
Query text,
Frequency)

Workload

(Database name,
Hash key → Statements,
Optimiser CPU time,
Optimiser disk I/O,
Execution CPU time,
Execution disk I/O,
Estimated CPU time,
Estimated disk I/O,
Wallclock to execute)

What to Monitor? (2)

Tables

(Database name,
Table ID,
Table name,
Frequency,
Storage structure,
Data pages,
Overflow pages)

References

(Database name,
Hash key → Statements,
Object type,
Object ID,
Table ID → Tables)

How to Monitor?

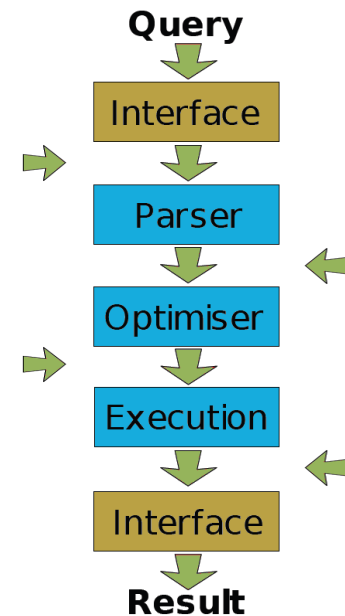
■ Ingres Management Architecture

- allows access to in-memory structures in DBMS core
- offers relational interface over standard SQL
- is used for (remote) monitoring and controlling of the DBMS

→ 'Sensors' are placed in the DBMS core

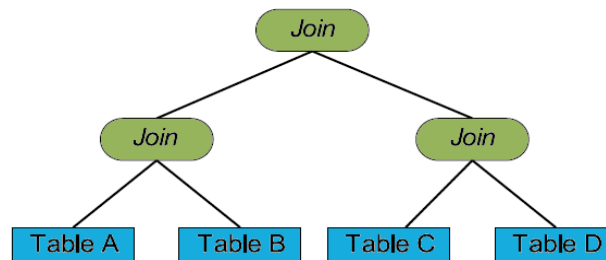
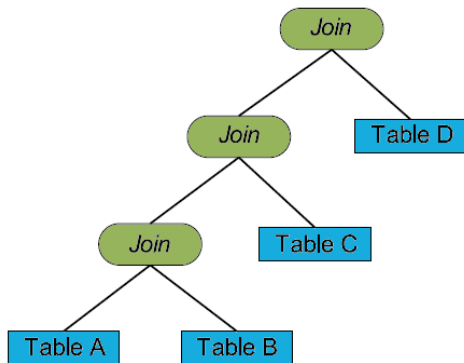
→ Structs are filled with data

→ Data can be read over IMA



Indexes in Ingres

- **Secondary indexes are tables**
 - ➔ Indexes are added to joining tables
 - ➔ Optimizer will automatically find fitting index
- **Left/Right-deep vs. bushy trees**
 - ➔ Too many indexes will slow down plan generation



➔ Need to find the *optimal* index set!

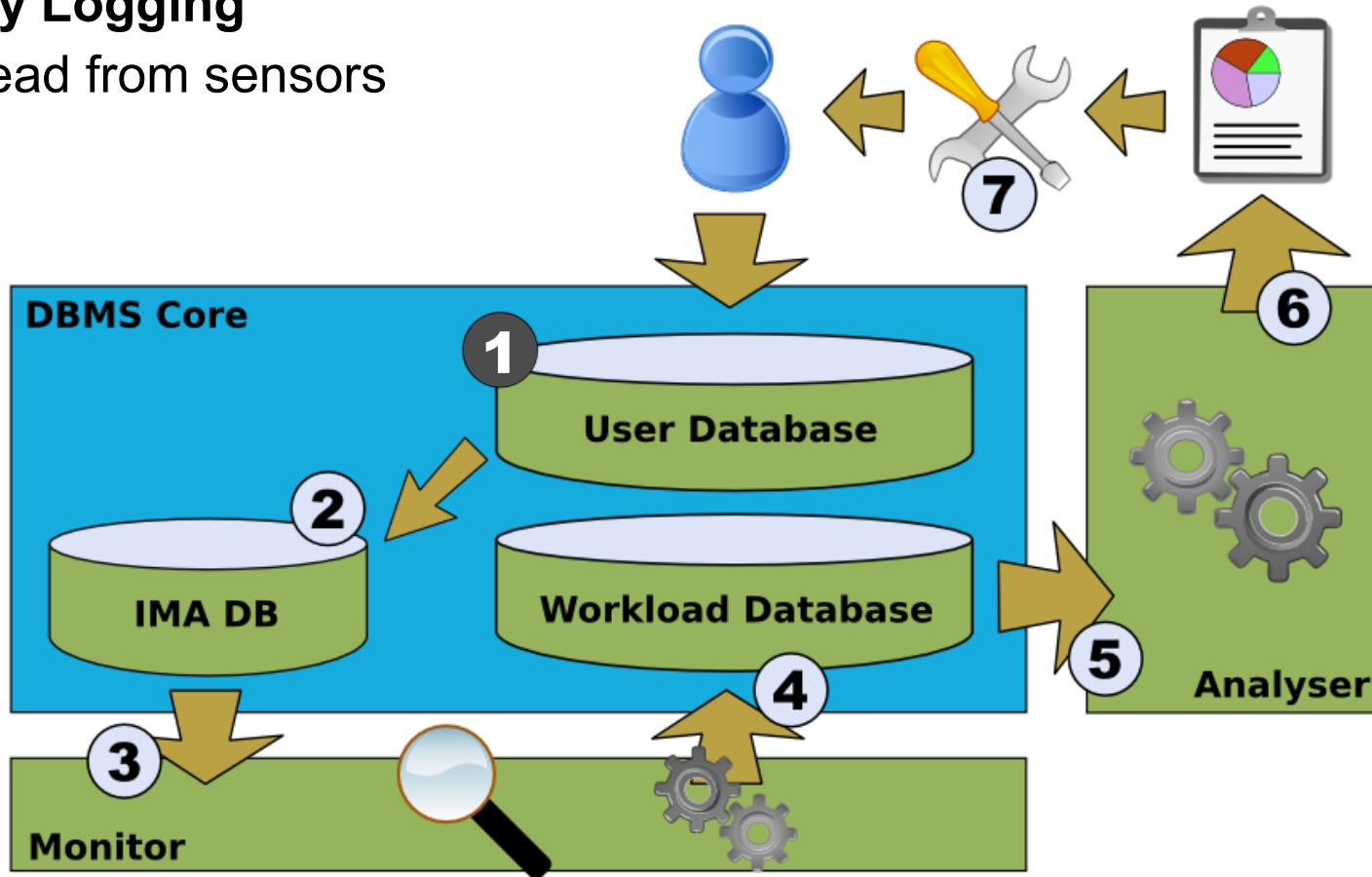


Implementation

Architecture

Query Logging

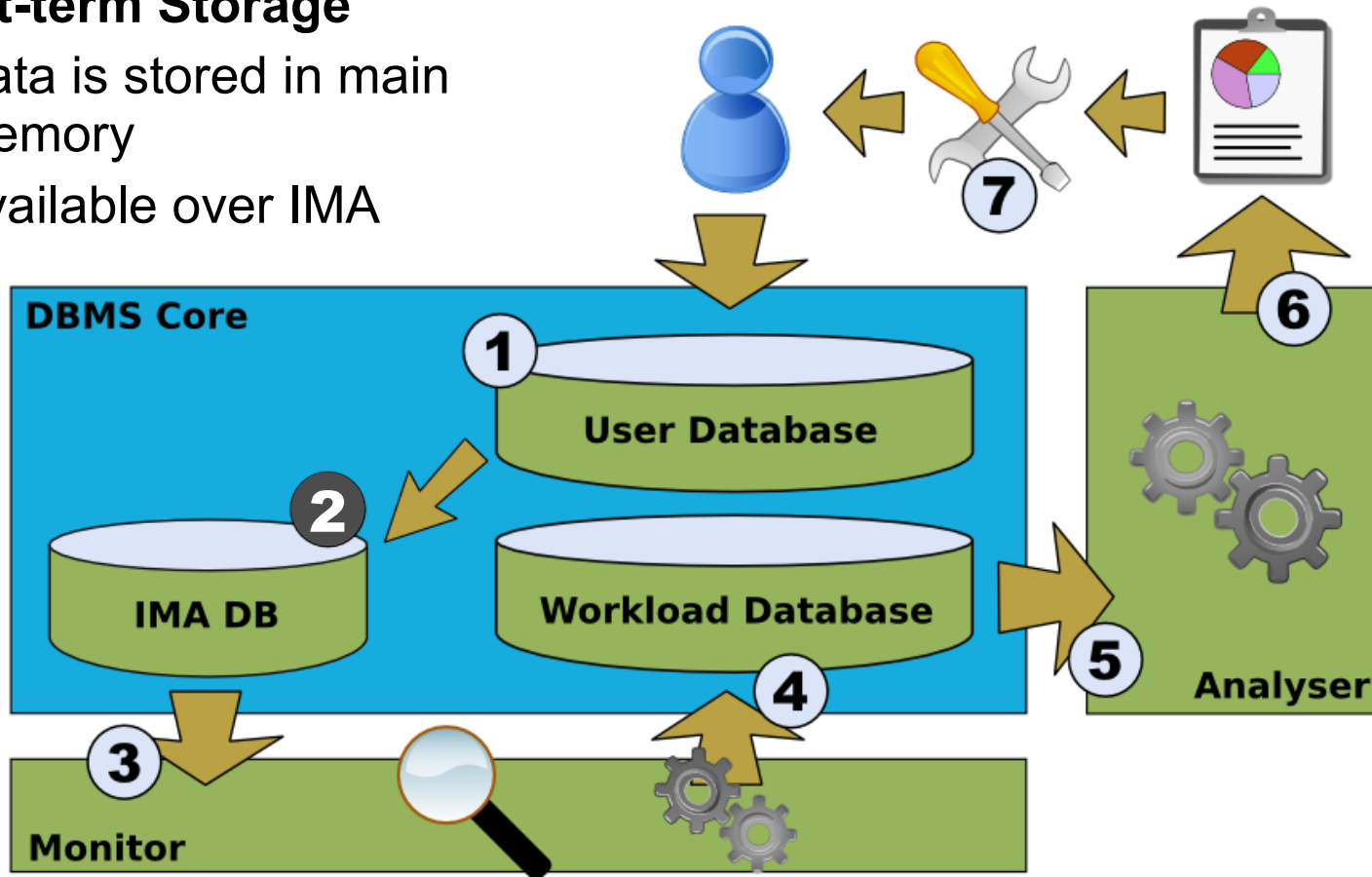
- Read from sensors



Architecture

Short-term Storage

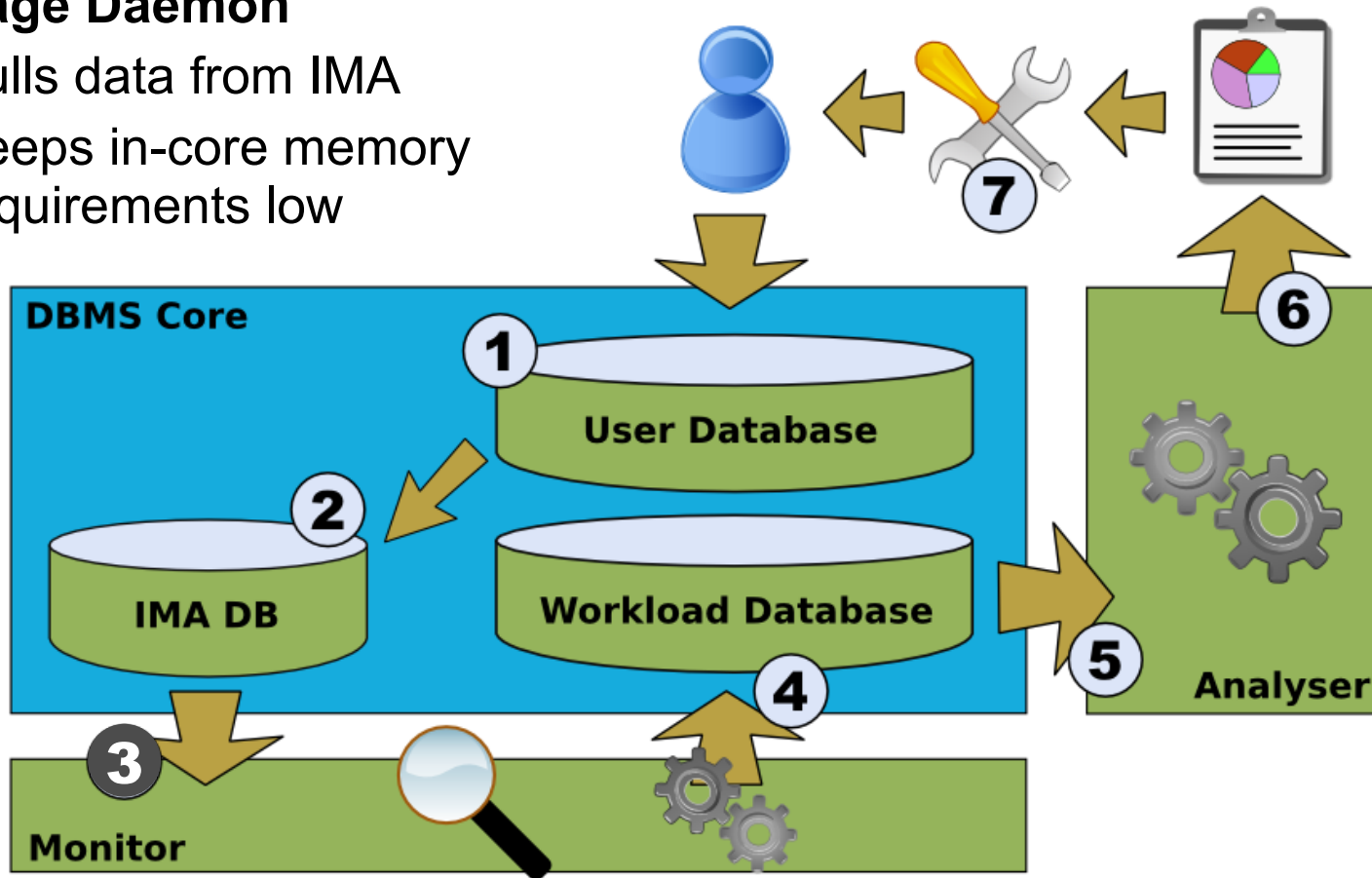
- Data is stored in main memory
- Available over IMA



Architecture

Storage Daemon

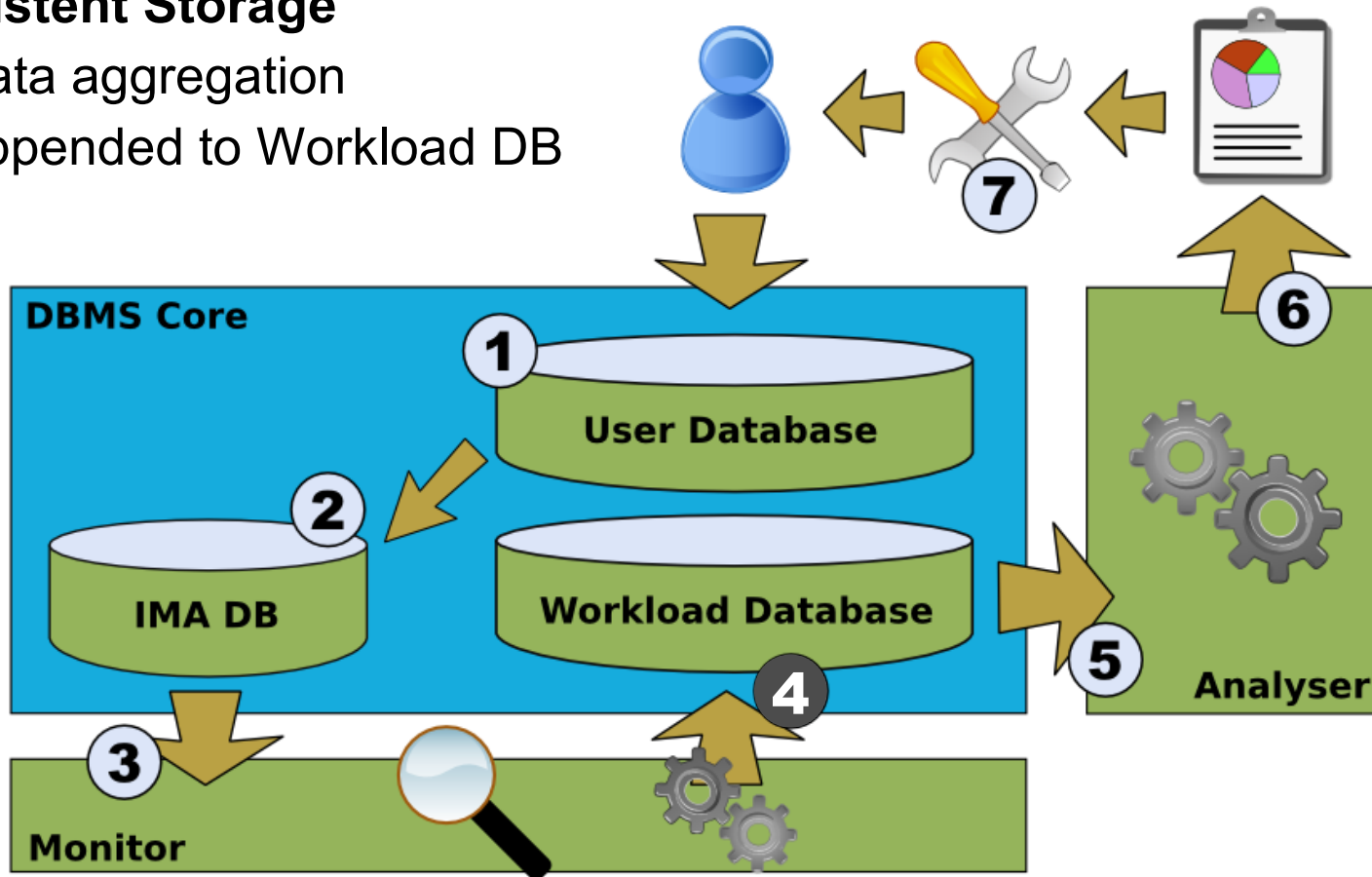
- Pulls data from IMA
- Keeps in-core memory requirements low



Architecture

Persistent Storage

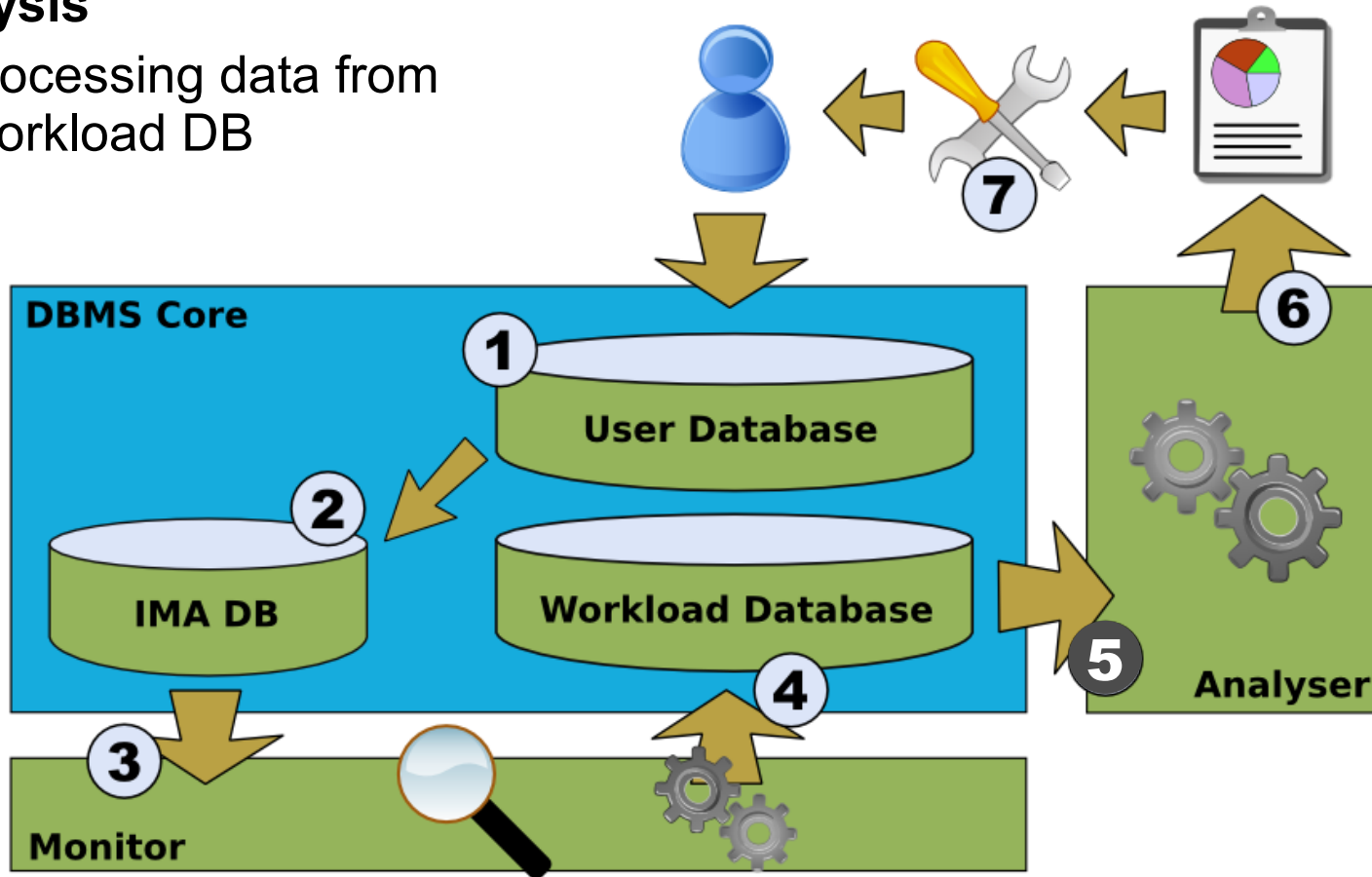
- Data aggregation
- Appended to Workload DB



Architecture

Analysis

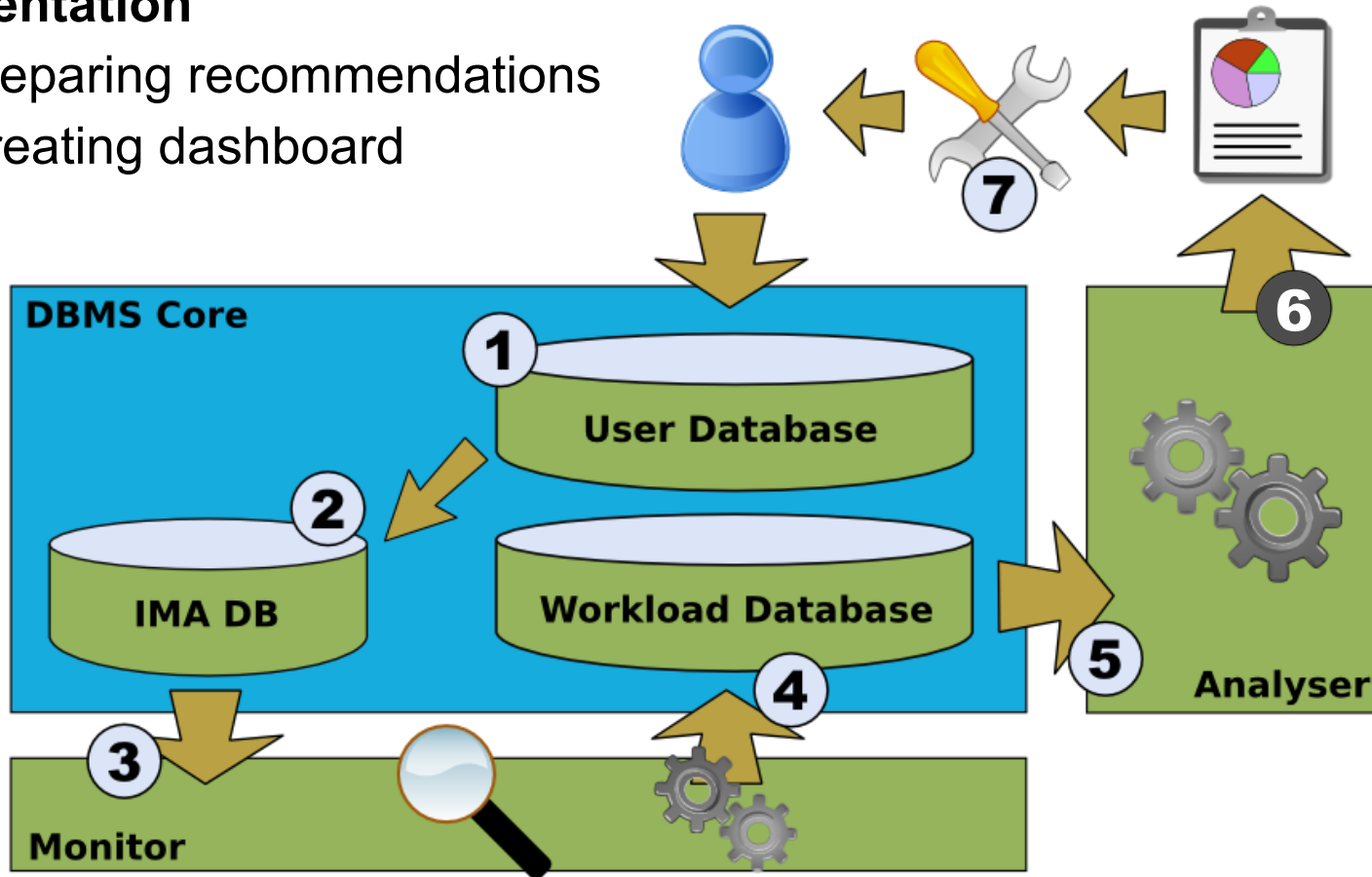
- Processing data from Workload DB



Architecture

Presentation

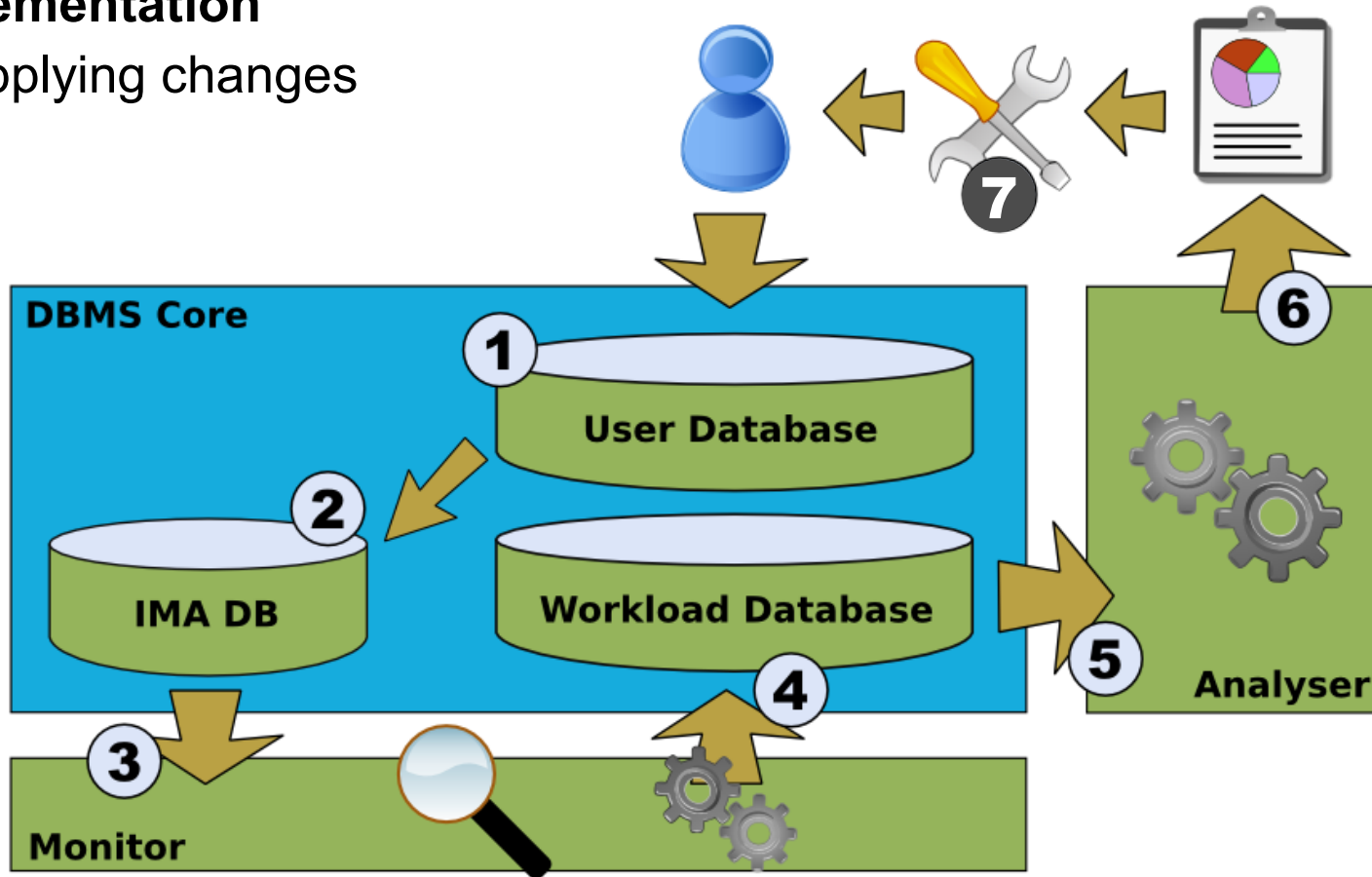
- Preparing recommendations
- Creating dashboard



Architecture

Implementation

- Applying changes





Evaluation

Monitoring Overhead

How much do we slow down the system?

- **Setups**

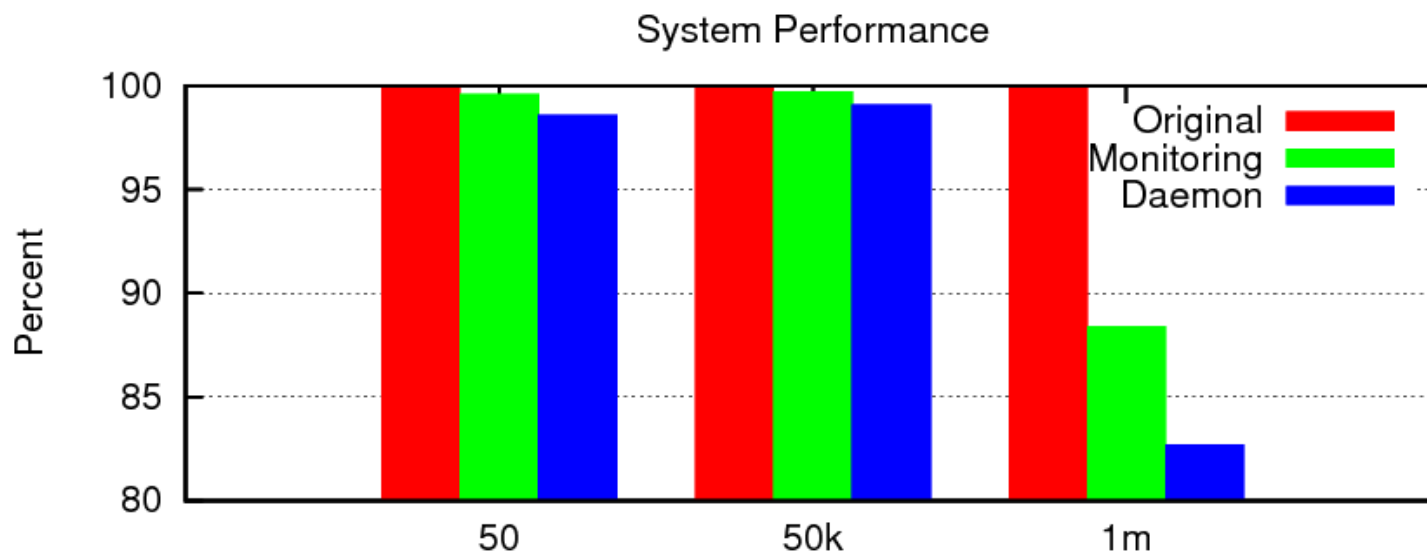
- 1) Vanilla Ingres
- 2) Monitoring Ingres
- 3) Monitoring Ingres with Storage Daemon

- **Tests**

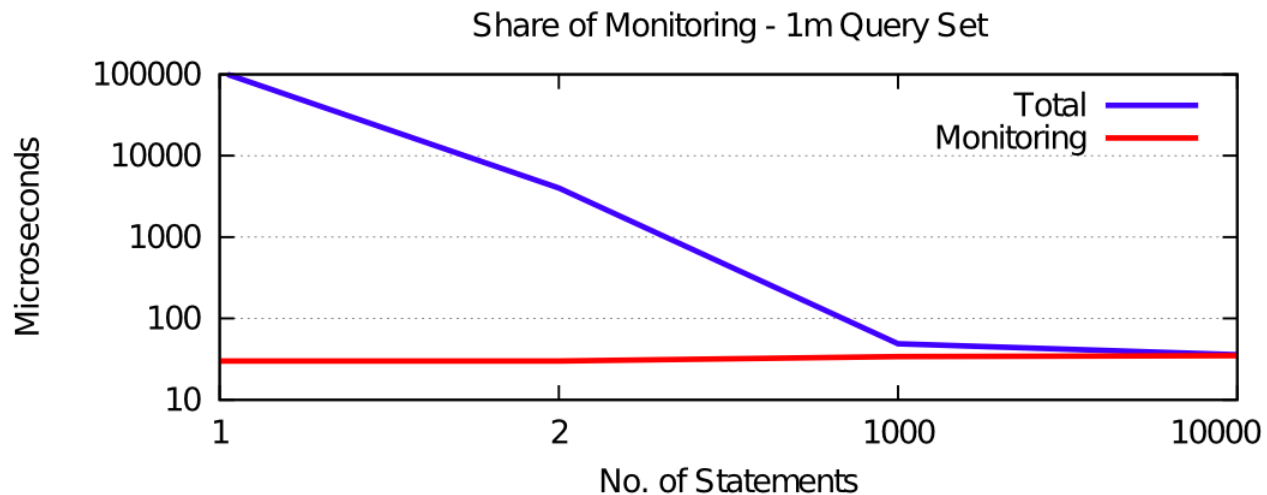
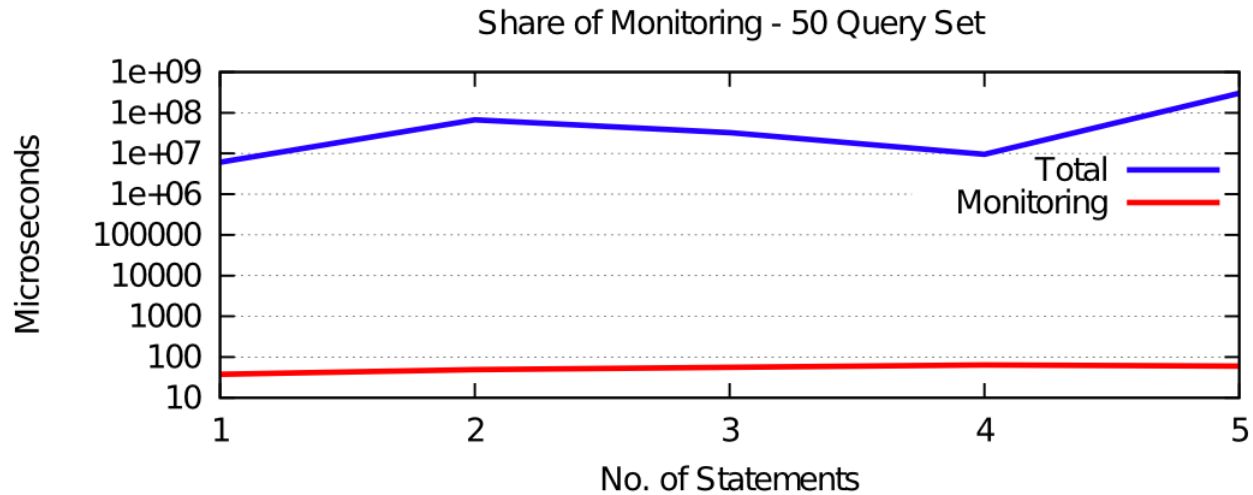
- a) 50 complex queries¹
- b) 50.000 two-table JOINS
- c) 1.000.000 one table SELECTs

¹From the Toronto Automatic Benchmarks / NREF database

Monitoring Overhead (2)

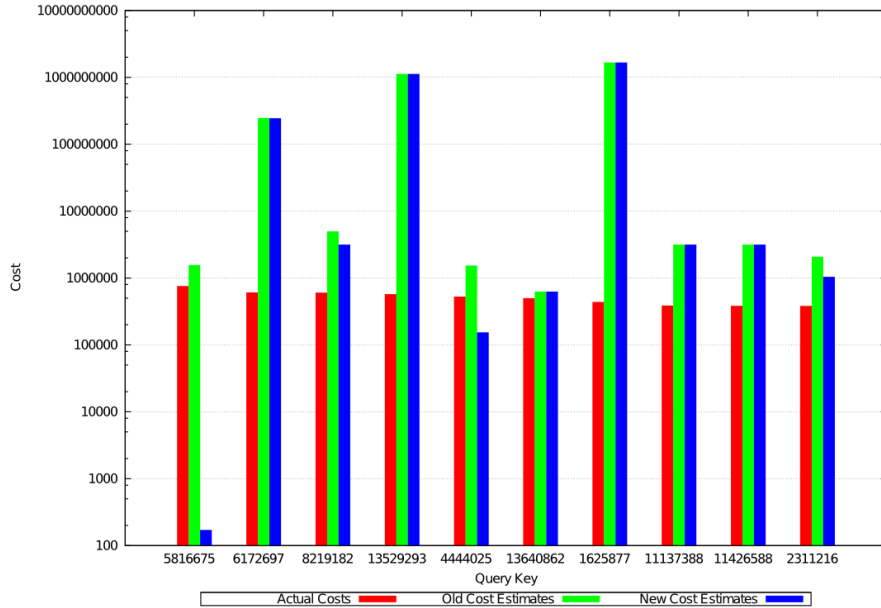


Monitoring Overhead (3)

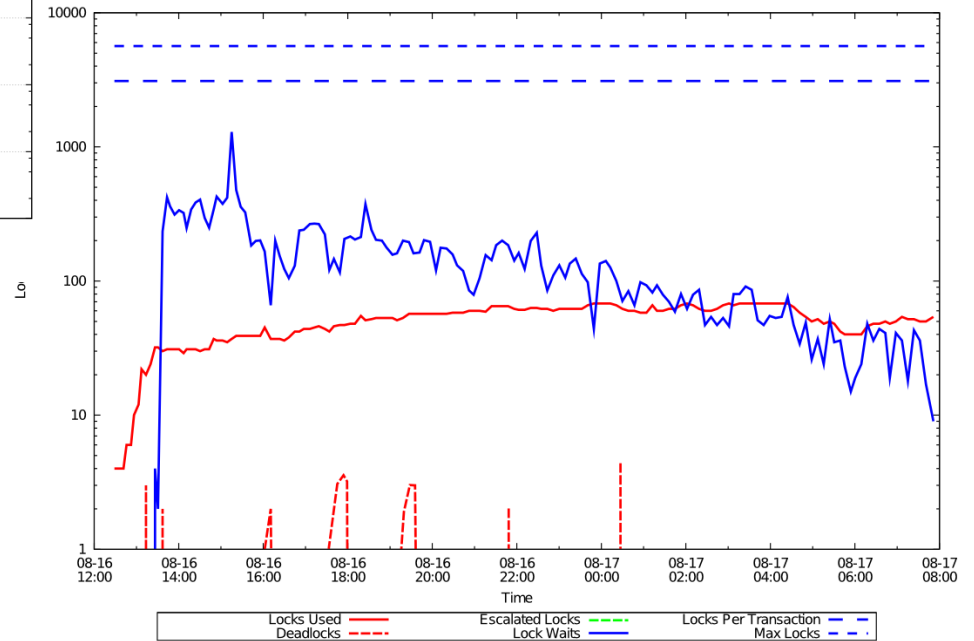


Analyzer Output

Top 10 - Cost Comparison



DBMS Locks





Outlook

INGRES

What's next?

- **Better caching for monitoring**
 - Lower overhead for cheap queries
- **Integrate storage daemon into core**
 - No need for external tool
- **Create smart analyzer**
 - Dependencies between physical structures
- **Autonomous tuning**
 - Automatic implementation of changes



Thank You

Alexander Thiem
alexander.thiem@ingres.com

Kai-Uwe Sattler
kus@tu-ilmenau.de

INGRES