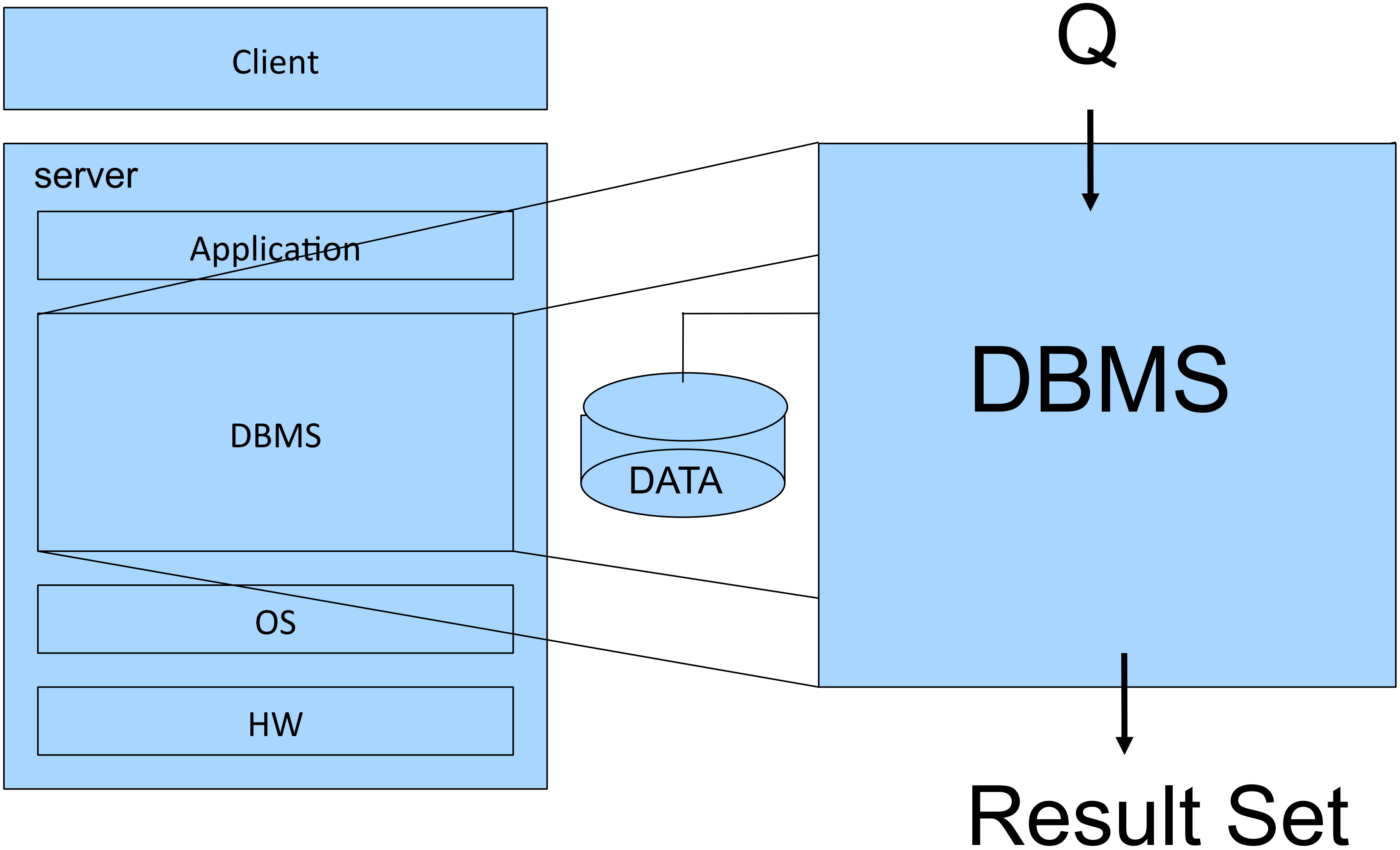
Troubleshooting techniques

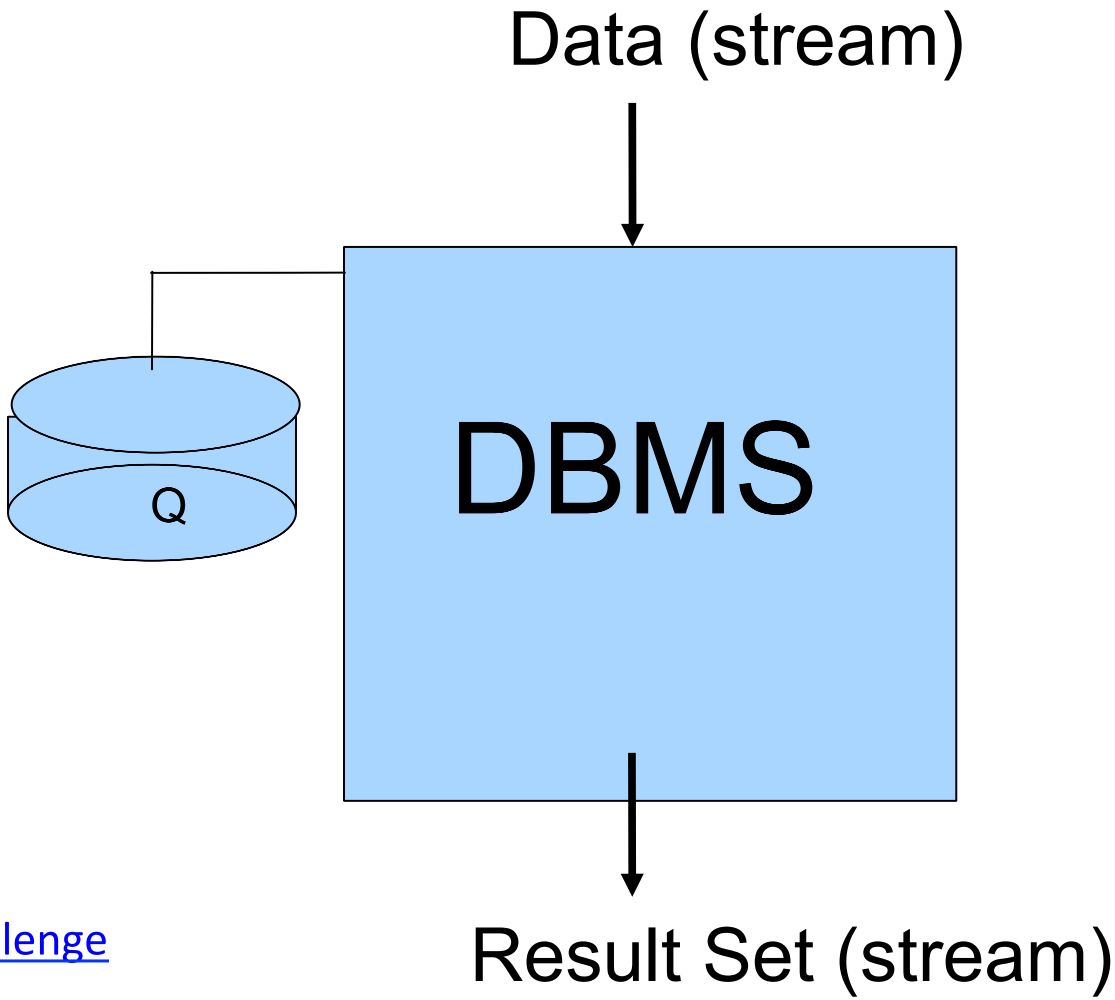
Outline

- DBMS architecture overview
  - More details as we progress through lectures
- Troubleshooting and experimentation
- Troubleshooting methodologies
  - Resource consumption model
  - Time-spent model

Traditional Architecture

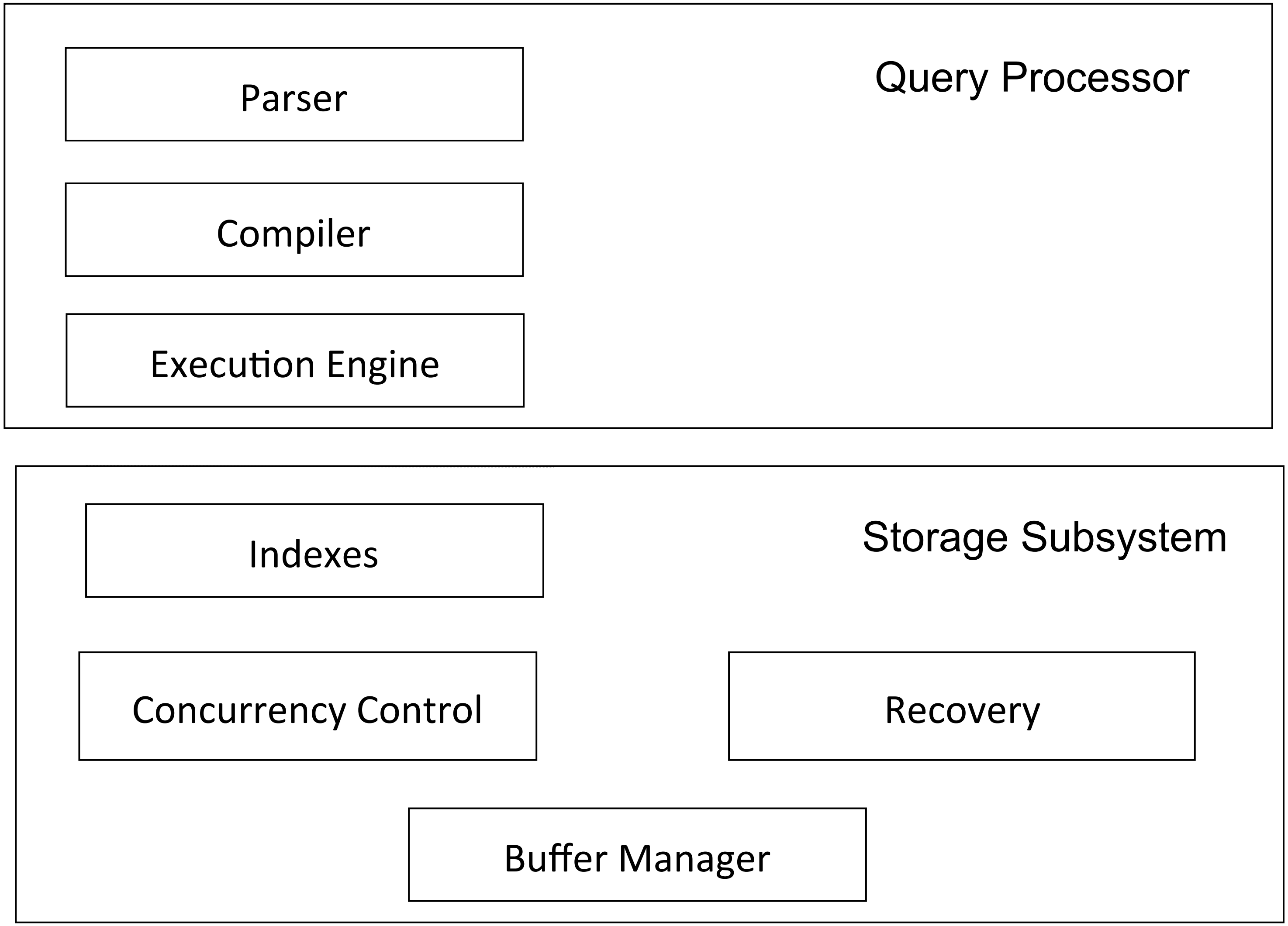


Streaming Architecture

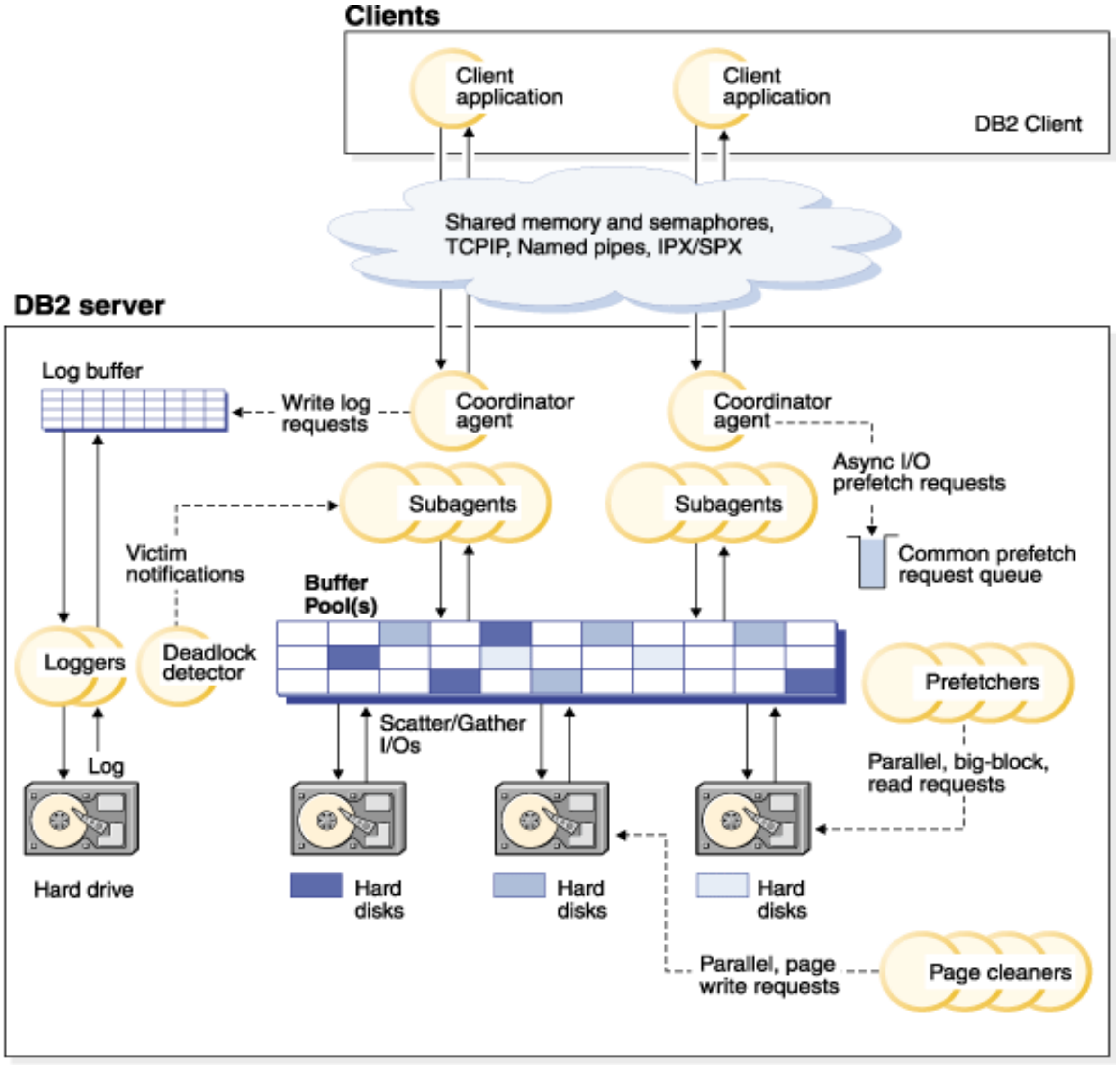


LOOK UP: DEBS2013 Grand Challenge

DBMS Components



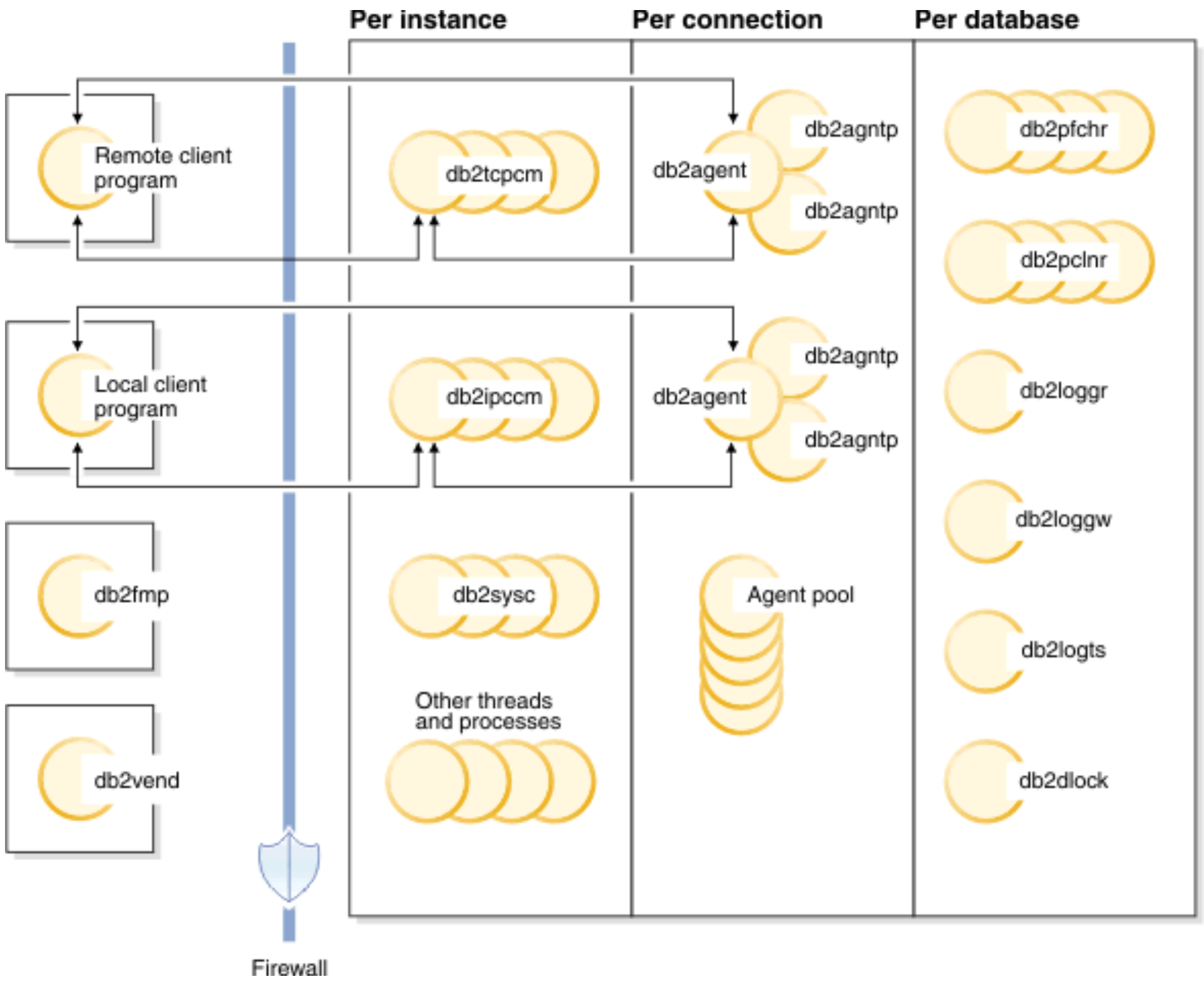
DB2 9.7 Process Architecture



@ Dennis Shasha and Philippe Bonnet, 2013

Source: http://pic.dhe.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.perf.doc/doc/00003525.gif

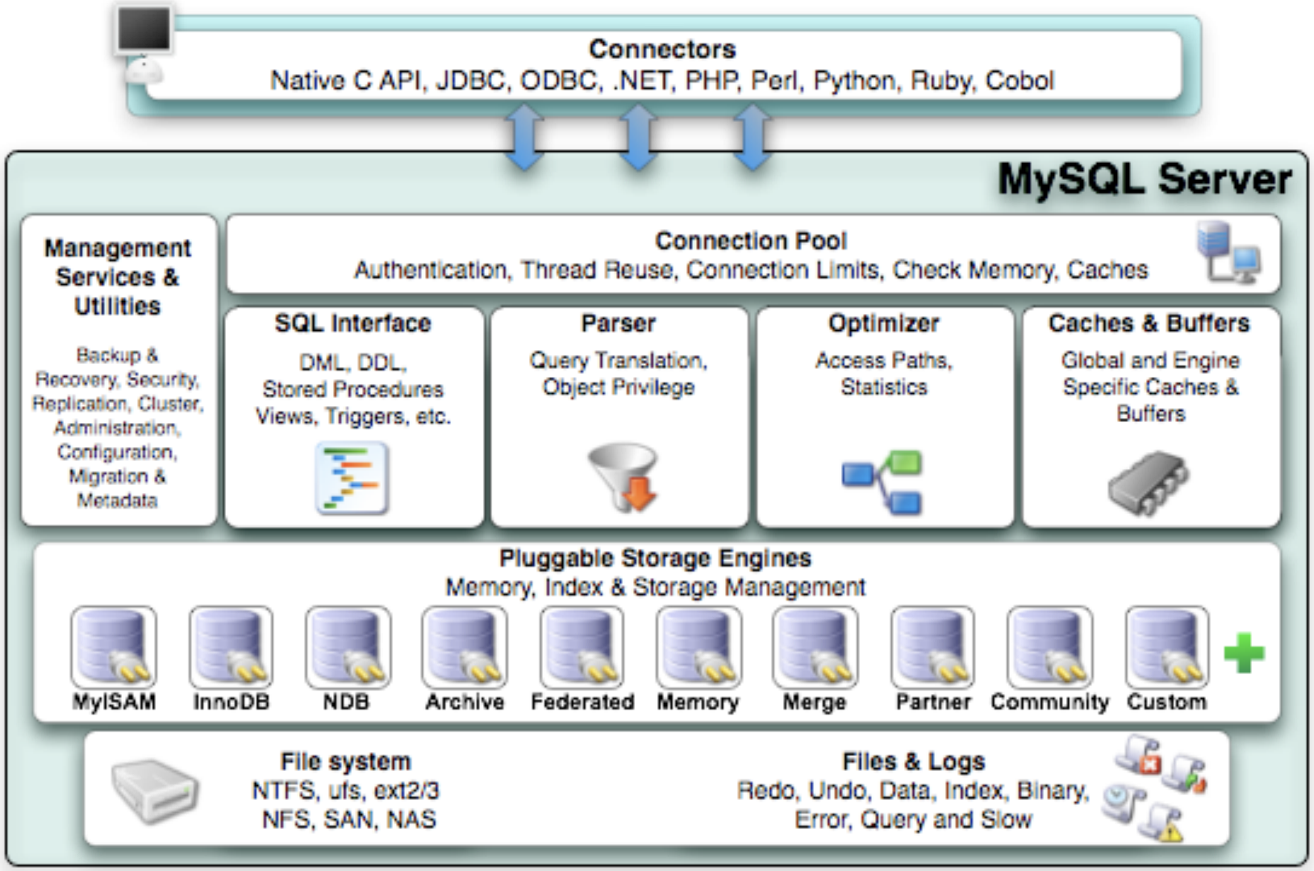
DB2 10.1 Process Architecture



@ Dennis Shasha and Philippe Bonnet, 2013

Source: http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/index.jsp?topic=%2Fcom.ibm.db2.luw.admin.perf.doc%2Fdoc%2Fc0008930.html

MySQL Architecture


Connectors
Native C API, JDBC, ODBC, .NET, PHP, Perl, Python, Ruby, Cobol
MySQL Server
Management Services & Utilities
Backup & Recovery, Security, Replication, Cluster, Administration, Configuration, Migration & Metadata
Connection Pool
Authentication, Thread Reuse, Connection Limits, Check Memory, Caches
SQL Interface
DML, DDL, Stored Procedures Views, Triggers, etc.
Parser
Query Translation, Object Privilege
Optimizer
Access Paths, Statistics
Caches & Buffers
Global and Engine Specific Caches & Buffers
Pluggable Storage Engines
Memory, Index & Storage Management
MyISAM
InnoDB
NDB
Archive
Federated
Memory
Merge
Partner
Community
Custom
File system
NTFS, ufs, ext2/3
NFS, SAN, NAS
Files & Logs
Redo, Undo, Data, Index, Binary, Error, Query and Slow

@ Dennis Shasha and Philippe Bonnet, 2013

Source: http://docs.oracle.com/cd/E19957-01/mysql-refman-5.5/storage-engines.html

Troubleshooting: Why

- **Production**
  - Users/manager complaints.
  - Needs monitoring. What is going on NOW?
  - Once identified, a problem should be represented in a synthetic form (so that others can avoid the problem, or as a request for new features from DBMS/OS)
- **Test**
  - New application / New system /
    New functionalities / New scale
  - Can system keep performance up in new settings?
  - Needs Experiments.

Troubleshooting: How

- You MUST measure system performance (black box)
  - Profiling tools
- You MUST instrument your system to get some insight about the internal processes (white box)
  - System instrumentation
- You MUST follow a systematic approach for troubleshooting / experimentation
  - Scientific Method
  - Troubleshooting methodology

Experimental Framework

1. System
   - Application + DBMS + OS + HW
   - Parameters (fixed/factors)
2. Metrics
   - Throughput / Response Time
3. Workload
   - Actual users (production), replay trace or synthetic workload (e.g., TPC benchmark)
4. Experiments
   - What factor to vary?

Exercise 2.1:
Is throughput always the inverse of response time?

Exercise 2.2:
Define an experiment to measure the write throughput of the file system on your laptop

Troubleshooting Methodologies

- Resource Consumption Model (Chapter 7)
  - Primary, DBMS system resources
  - Applications as consumers
  - Instrumentation through aggregate probes
- Time Spent Model
  - response time = execution time + wait time
  - Instrumentation to obtain session-based time series

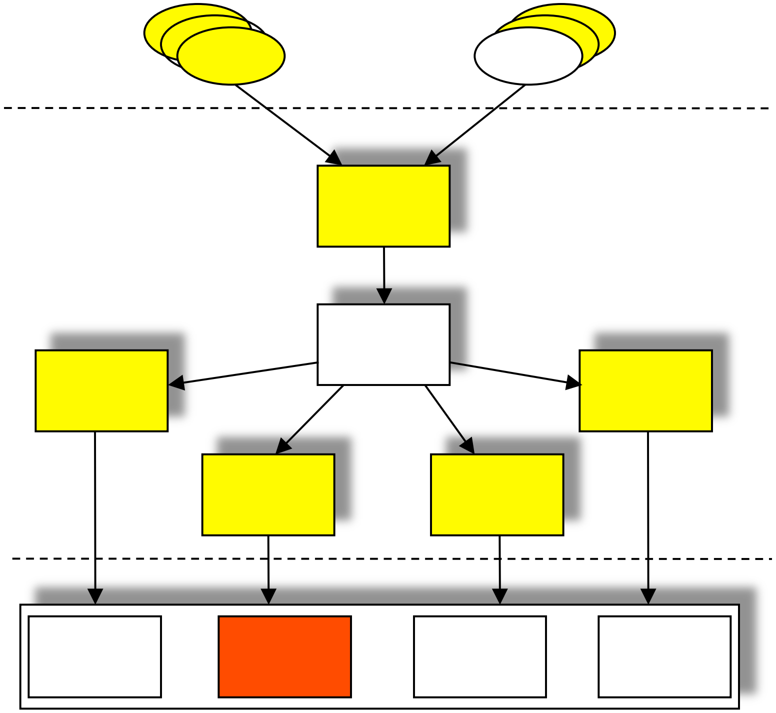
Exercise 2.3:

Does DB2 top follow the resource consumption or time spent model?

Resource Consumption Model

Effects are not always felt first where the cause is!

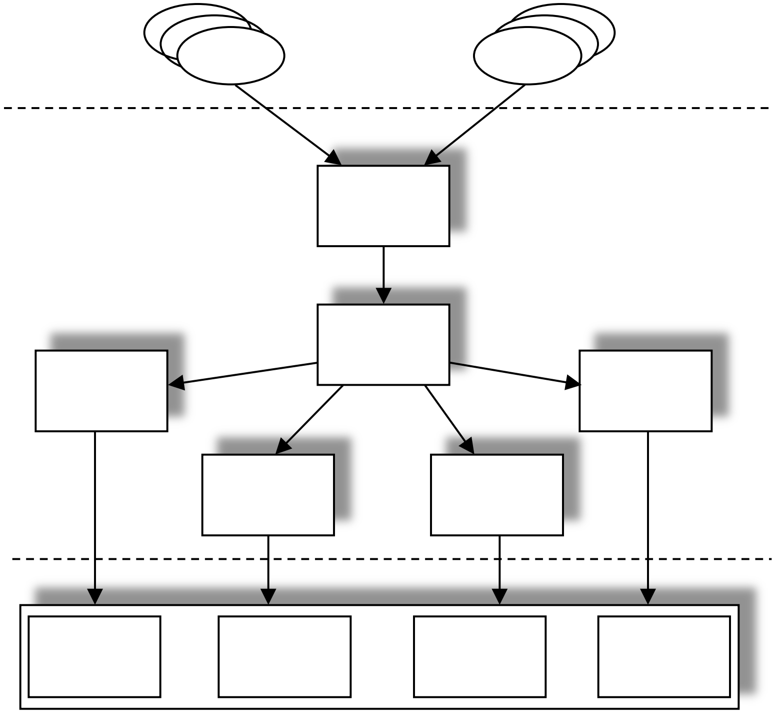
- An overloading high-level consumer
- A poorly parameterized subsystem
- An overloaded primary resource



Resource Consumption Model

Extract indicators to answer the following questions

- Question 1: Are critical queries being served in the most efficient manner?
- Question 2: Are subsystems making optimal use of resources?
- Question 3: Are there enough primary resources available?



Methodology

- Which are the critical queries?
  - Ask users.
  - Use query log to find queries that take longest.
- Resource Usage
  - Check out <u>system-wide</u> performance indicators.
  - Use rules of thumbs to check whether they are ok.

Example

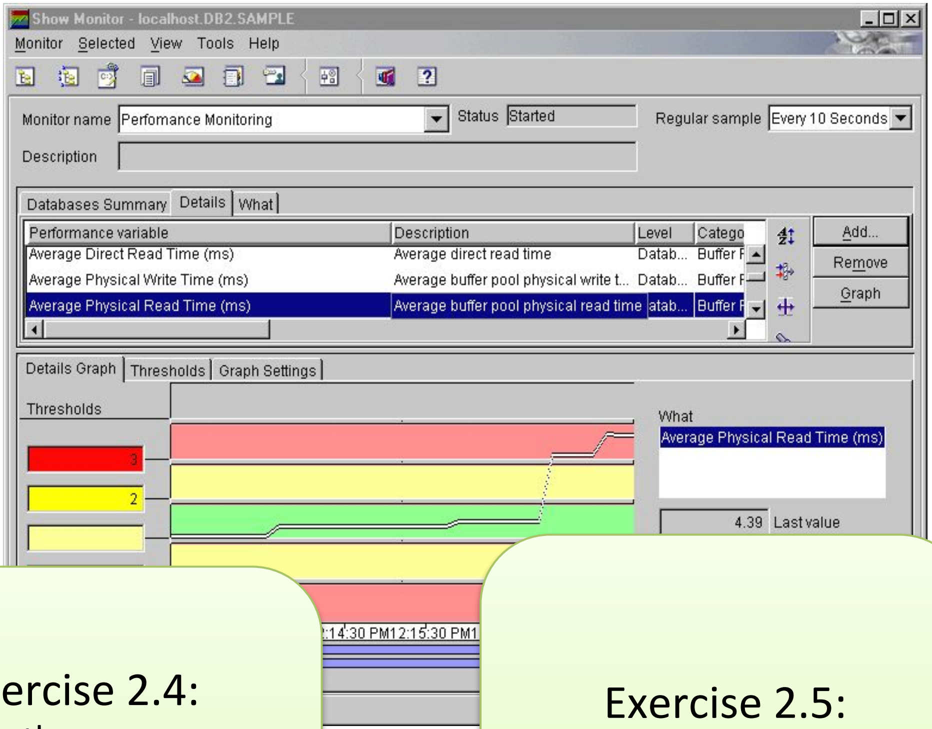
DBMS

```
Statement number: 1
select C_NAME, N_NAME
from DBA.CUSTOMER join DBA.NATION on C_NATIONKEY = N_NATIONKEY
where C_ACCTBAL > 0

Number of rows retrieved is:    136308
Number of rows sent to output is:    0
Elapsed Time is:           76.349     seconds

…
Buffer pool data logical reads              = 272618
Buffer pool data physical reads             = 131425
Buffer pool data writes                     = 0
Buffer pool index logical reads             = 273173
Buffer pool index physical reads            = 552
Buffer pool index writes                    = 0
Total buffer pool read time (ms)            = 71352
Total buffer pool write time (ms)           = 0
…
Summary of Results
==================
               Elapsed          Agent CPU       Rows      Rows
Statement #     Time (s)         Time (s)       Fetched   Printed
1               76.349           6.670          136308      0
```

OS



Monitor name: Perfomance Monitoring
Status: Started
Regular sample: Every 10 Seconds
Description:

Performance variable	Description	Level	Catego
Average Direct Read Time (ms)	Average direct read time	Datab...	Buffer F
Average Physical Write Time (ms)	Average buffer pool physical write t...	Datab...	Buffer F
Average Physical Read Time (ms)	Average buffer pool physical read time	atab...	Buffer F

What: Average Physical Read Time (ms)

4.39 Last value

Exercise 2.4:
What are the 11 system wide indicators recommended for DB2 10.1?

Exercise 2.5:
Which are the profiling tools on your laptop OS?

Fine Granularity Analysis

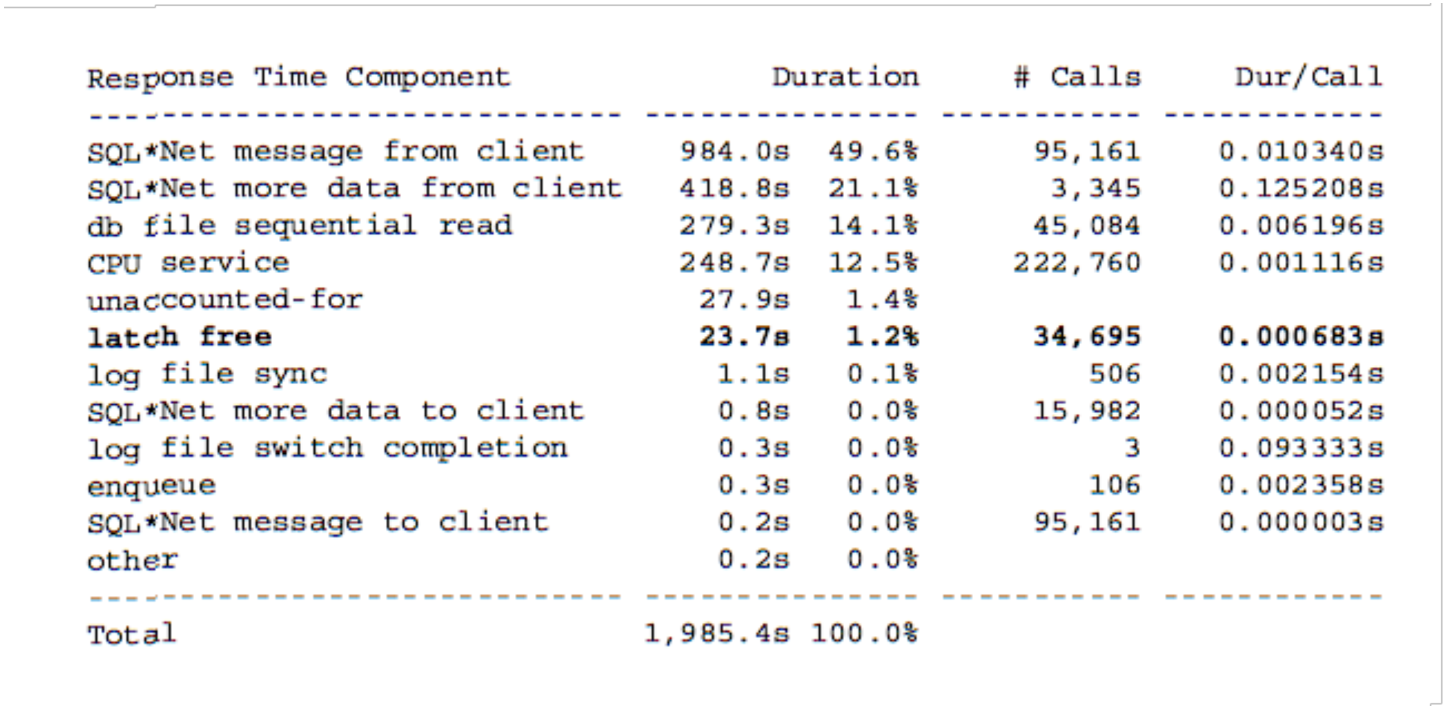
LOOK UP:

- Index usage analysis in DB2 10.1
- Event and resource monitoring in DB2 10.1

Time Spent Model

- Given a critical query / session
- Where does the time go?
  - Throughout DBMS/OS/HW components
  - Waiting / Executing
  - Find out which components cause a session to be slow.

Example: Oracle



```
Response Time Component             Duration        # Calls      Dur/Call
---------------------------- -------------- ----------- ------------
SQL*Net message from client     984.0s  49.6%      95,161    0.010340s
SQL*Net more data from client   418.8s  21.1%       3,345    0.125208s
db file sequential read         279.3s  14.1%      45,084    0.006196s
CPU service                     248.7s  12.5%     222,760    0.001116s
unaccounted-for                  27.9s   1.4%
latch free                       23.7s   1.2%      34,695    0.000683s
log file sync                     1.1s   0.1%         506    0.002154s
SQL*Net more data to client       0.8s   0.0%      15,982    0.000052s
log file switch completion        0.3s   0.0%           3    0.093333s
enqueue                           0.3s   0.0%         106    0.002358s
SQL*Net message to client         0.2s   0.0%      95,161    0.000003s
other                             0.2s   0.0%
---------------------------- -------------- ----------- ------------
Total                         1,985.4s 100.0%
```

Source: Cary Millsap: Oracle Operational Timing Presentation

Example: IBM DB2 10.1

Time-spent monitors

- Wait time
- Processing time
- Elapsed time

Scope

- Database
- Application handle
- Resource (lock, buffer, …)

LOOK UP: MON_GET_ACTIVITY_DETAILS table function

Exercise 2.6:

Assume your DBMS is DB2 10.1. How can you answer the following questions:

- How many IOs are performed while a given query is running?
- How much time is spent performing those IOs?
- What percentage of the total execution time does it take to perform those IOs?
- What portion of those IOs are sequential?

Hint, LOOK UP:

- Workloads in DB2
- Table space utilization view
- Physical reads and writes monitor elements