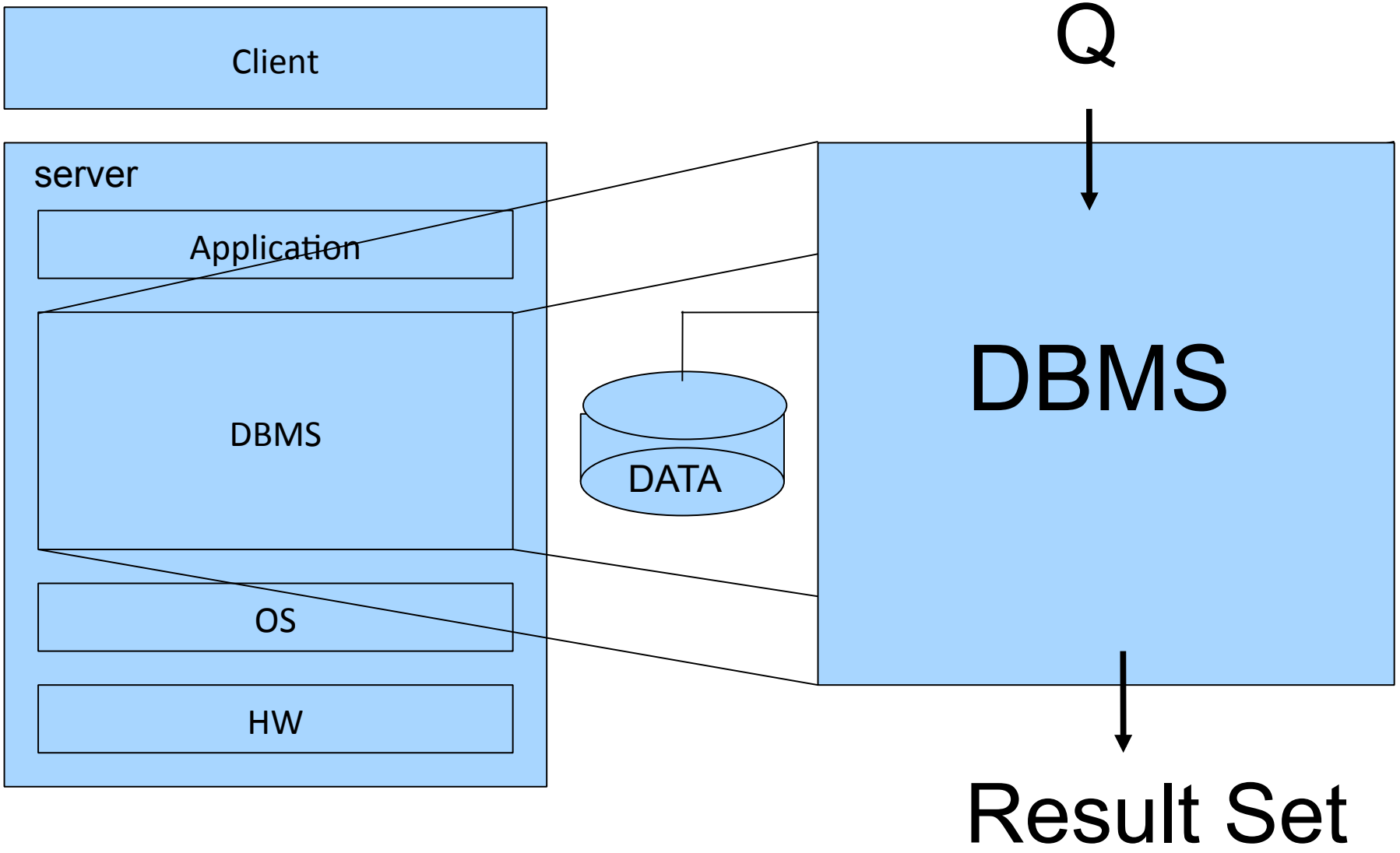
Troubleshooting techniques

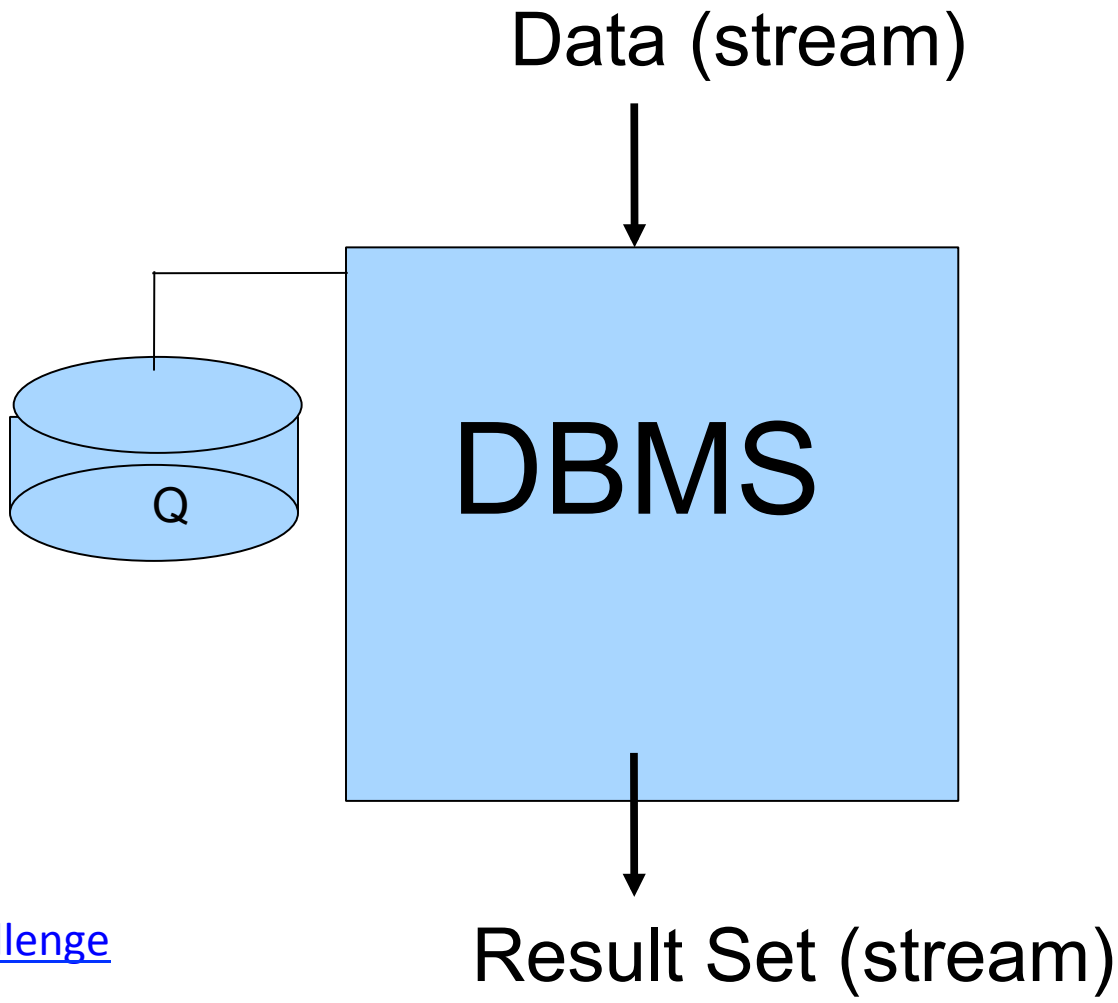
Outline

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

Traditional Architecture

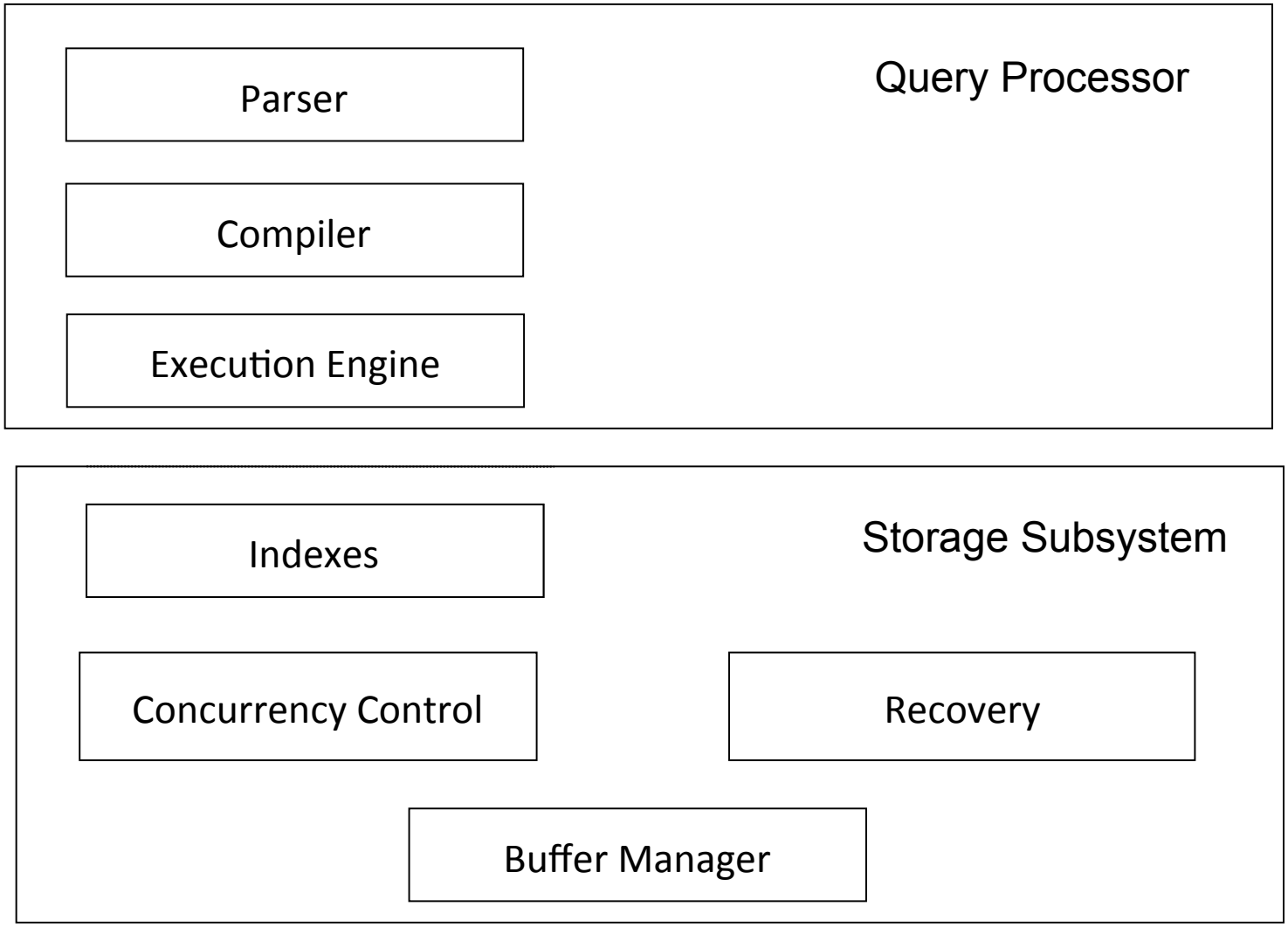


Streaming Architecture

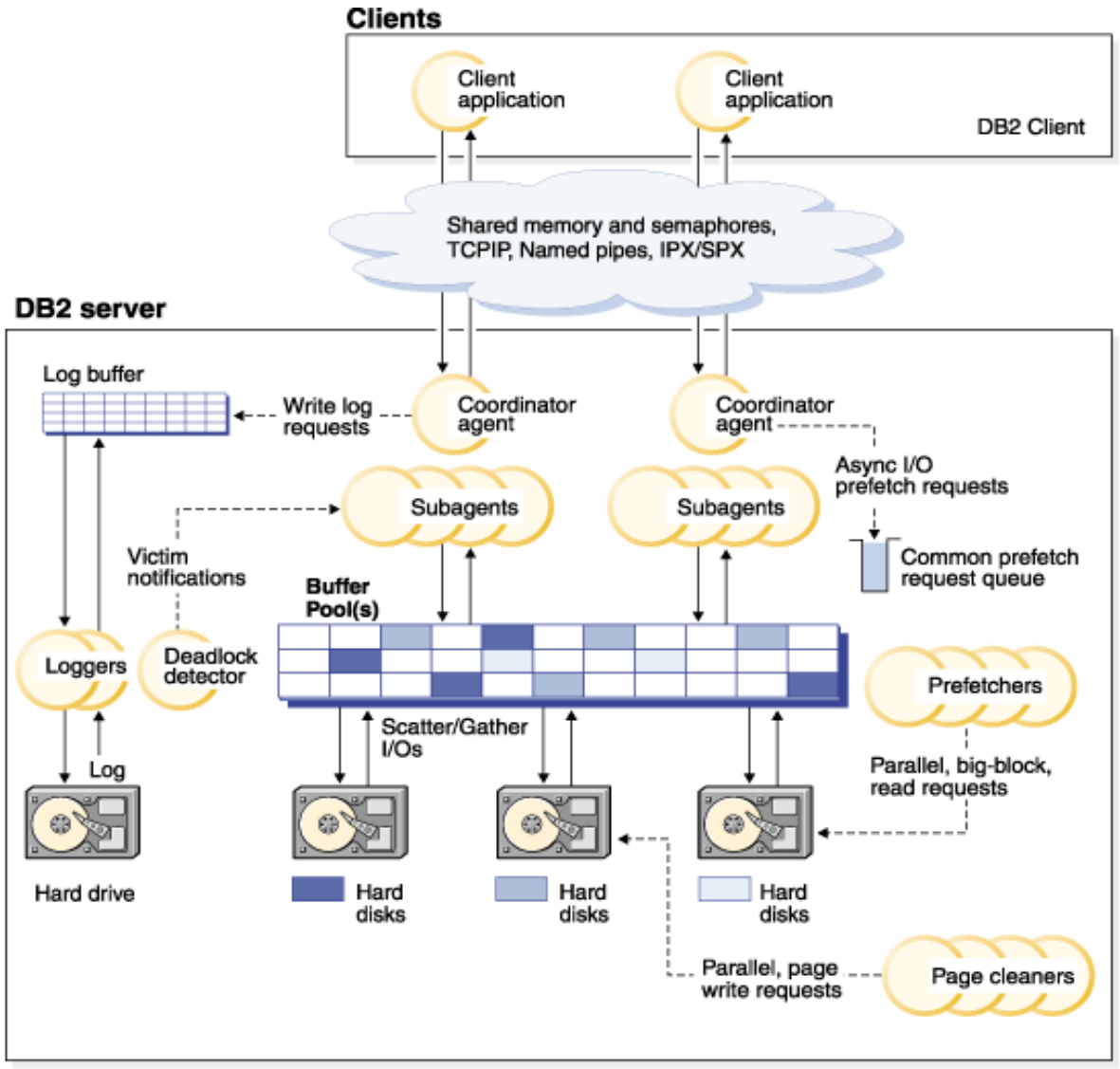


LOOK UP: DEBS2013 Grand Challenge

DBMS Components



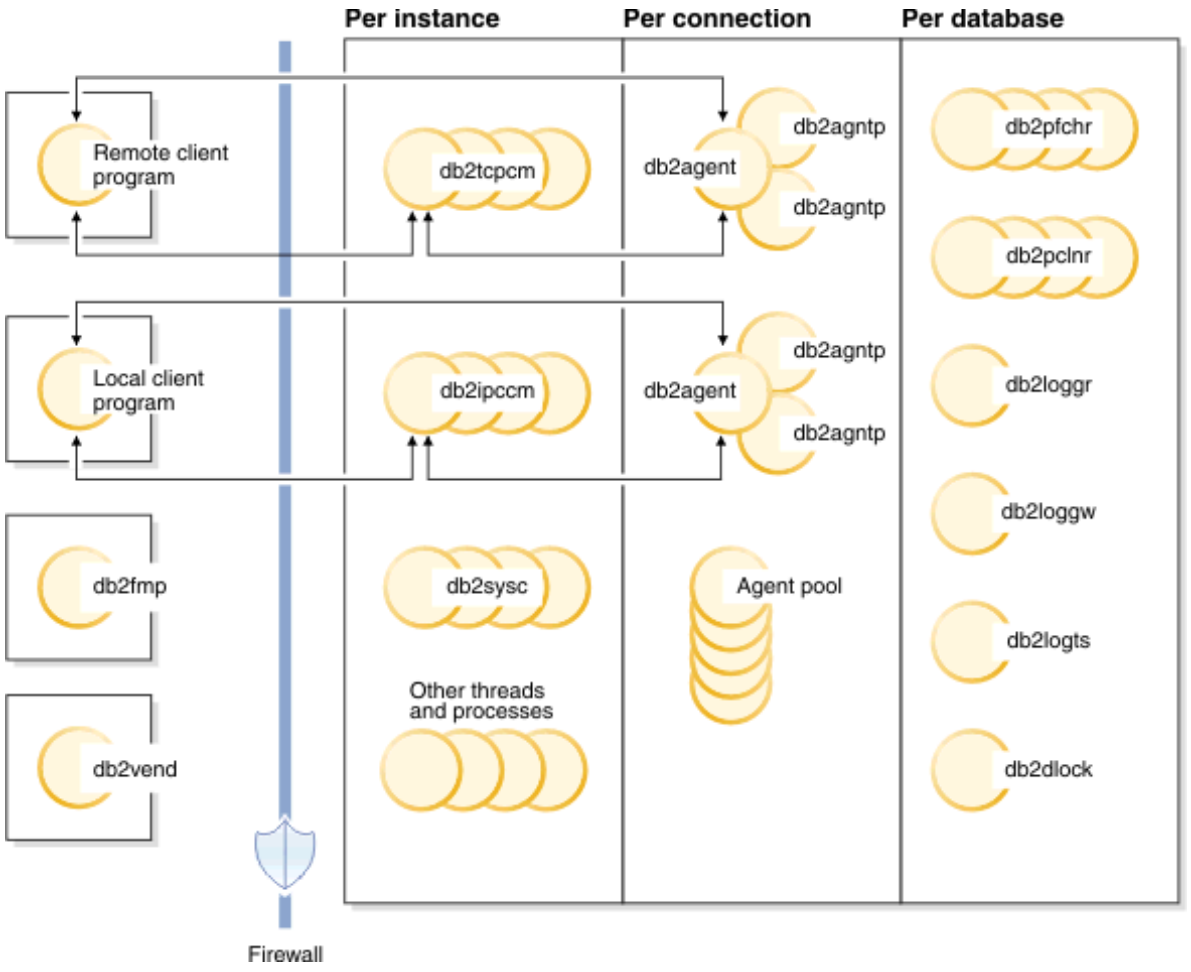
DB2 9.7 Process Architecture



Exercise 2.1:
Is intra-query parallelism possible with this process model? In other words, can a query be executed in parallel within a same instance (or partition)?

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

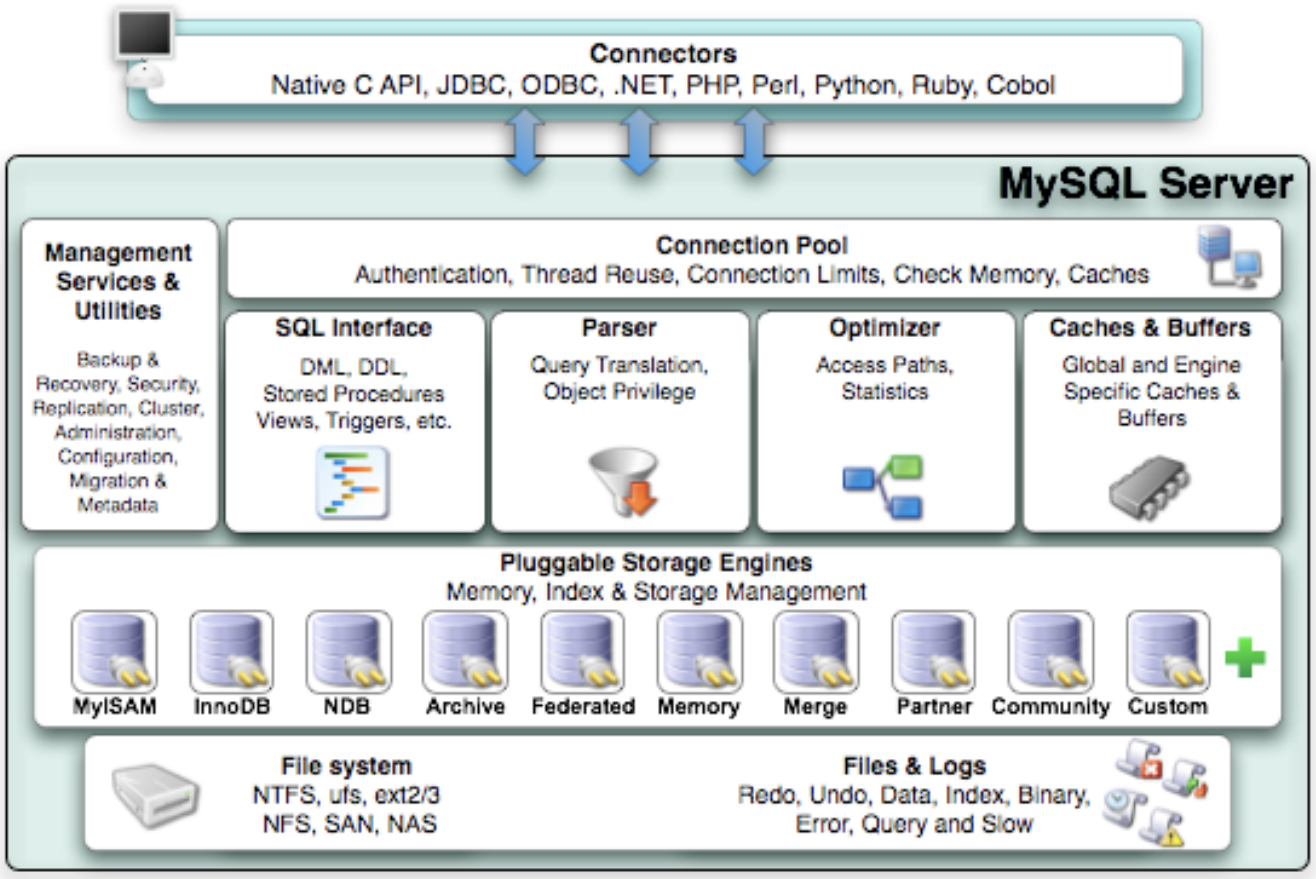
DB2 10.1 Process Architecture



No need to know much more about the process abstractions. We will cover much more on the memory abstractions (log tuning), and on the communication abstractions (tuning the application interface, tuning across instances).

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

MySQL Architecture



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
   - DBMS performance indicators
   - OS performance indicators
3. Workload
   - Actual users (production), replay trace or synthetic workload (e.g., TPC benchmark)
4. Experiments
   - What factor to vary?

Exercise 2.2:
Is throughput always the inverse of response time?

Exercise 2.3:
Define an experiment to measure the write throughput of the file system on your laptop

Performance Indicators

- Type
  - Counter
  - Watermark
  - Gauge
  - Timestamp
- Scope
  - System-wide (snapshot) vs. workload-specific (activity detail)
- Collection
  - **switched on/off** vs. triggered by a specific event
  - File dump vs. materialized view
    - Frequency of update of the materialized view
- Access
  - Table functions, views, XML files
  - Alert triggered by a specific event

Exercise 2.4:
What are system-wide indicators good for?

LOOKUP: DB2 monitor elements and interfaces for monitoring, Oracle dynamic performance views

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

Windows: Performance monitor
Linux: iostat, vmstat
More on this in tuning the guts.

Exercise 2.6:
What are the [11 system wide indicators] recommended for DB2 10.1?

Exercise 2.7:
Which are the profiling tools on your laptop OS?

Troubleshooting Methodologies

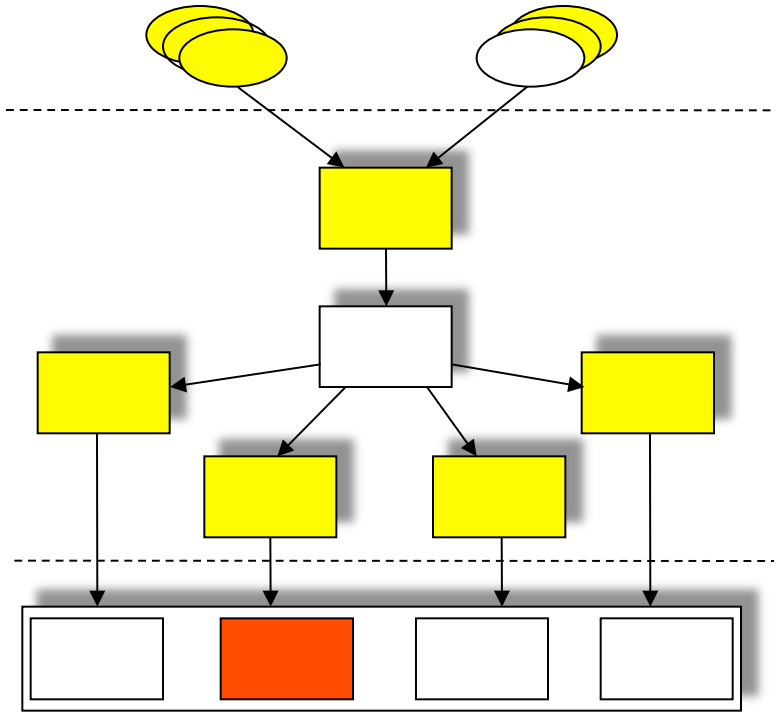
- Resource Consumption Model (Chapter 7)
  - Primary, DBMS system resources
  - Applications as consumers
  - Instrumentation through counters/ watermarks/gauge
- Time Spent Model
  - response time = execution time + wait time
  - Instrumentation through timestamps

Exercise 2.5:
Does DB2 top follow the resource consumption or time spent model?

Resource Consumption Model

Effects are not always felt first where the cause is!

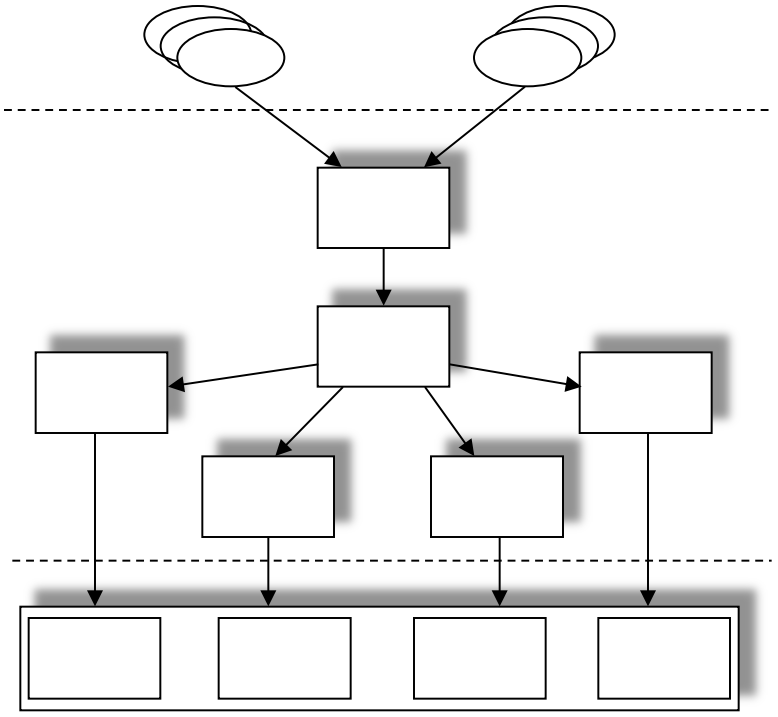
- An overloading high-level consumer
- A poorly parameterized subsystem
- An overloaded primary resource



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 system-wide performance indicators.
  - Use rules of thumbs to check whether they are ok.

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.

Issues:
- How to instrument ONLY a given critical query/sesssion
- What timestamps indicators are available?

LOOK UP: Cary Millsap's book: Oracle Operational Timing Presentation

Example: IBM DB2 10.1

Time-spent monitors

- Wait time
- Processing time
- Elapsed time

Scope

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

LOOK UP: DB2 10.1 GET SNAPSHOT, MON_GET_ACTIVITY_DETAILS table function, work identification by origin with workloads

Exercise 2.8:

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

- How many IOs are performed from the command line for the execution of a given query?
- How many pages are actually read? How much space is used in the buffer pool?
- 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?
- How much overhead is there in obtaining the answer to those questions?

Exercise 2.8: mon_IO.sql

```
SELECT A.AGENT_ID,
        B.ROWS_RETURNED,
        B.ROWS_READ,
        (B.POOL_TEMP_DATA_L_READS + B.POOL_INDEX_L_READS +
                B.POOL_TEMP_DATA_L_READS + B.POOL_TEMP_INDEX_L_READS) as BUFFER_POOL_READS,
        (B.POOL_DATA_P_READS + B.POOL_DATA_P_READS +
                B.POOL_TEMP_DATA_P_READS + B.POOL_TEMP_INDEX_P_READS) as IO_TOTAL,
        B.POOL_READ_TIME as IO_TIME,
        B.CLIENT_IDLE_WAIT_TIME as CLIENT_WAIT_TIME,
        B.TOTAL_RQST_TIME as DB2_SPENT_TIME,
                        B.TOTAL_WAIT_TIME as DB2_WAIT_TIME,
                        B.TOTAL_COMPILE_TIME as DB2_COMPILE_TIME,
                        B.TOTAL_SECTION_PROC_TIME as DB2_SECTION_TIME,
                        B.TOTAL_COMMIT_PROC_TIME as DB2_COMMIT_TIME,
                        B.TOTAL_ROLLBACK_PROC_TIME as DB2_ROLLBACK_TIME,
                        B.TOTAL_RUNSTATS_PROC_TIME as DB2_RUNSTATS_TIME,
                        B.TOTAL_REORG_PROC_TIME as DB2_REORG_TIME,
                        B.TOTAL_LOAD_PROC_TIME as DB2_LOAD_TIME
FROM    SYSIBMADM.APPLICATIONS A,
        TABLE(MON_GET_CONNECTION(cast(NULL as bigint), -1)) B
WHERE A.DB_NAME = 'TUNING'
   AND SUBSTR(A.APPL_NAME,1,5) = 'db2bp'
   AND A.AGENT_ID = B.APPLICATION_HANDLE
;
```

Exercise 2.8:

Let us assume you are done with the GettingStarted exercise, then the aircraft table has been populated inside the tuning database on the db2inst1 instance

```
db2inst1@student-VirtualBox:~$ db2 connect to tuning
db2inst1@student-VirtualBox:~$ db2 –f mon_IO.sql
db2inst1@student-VirtualBox:~$ db2 –f mon_IO.sql
db2inst1@student-VirtualBox:~$ db2 “select * from aircraft”
db2inst1@student-VirtualBox:~$ db2 –f mon_IO.sql
```