IBM Chat with Lab for Greater China Group

- **Executive introduction** (audio)

  Sal Vella, Vice President, Development, Distributed Data Servers and Data Warehousing

- **Presentation: Monitoring and Tuning DB2 System Performance**

  Guiyun Cao, IBM DB2 performance analyst

- **Host:** Frank Ning, Manager, DB2 LUW Install and Up/Running Development

Executive Introduction

Sal Vella

Vice President, Development, Distributed Data Servers and Data Warehousing

IBM Software Group

IBM Software Group - Information Management

Monitoring and Tuning DB2 System Performance

- Chat with the Toronto Lab for the Greater China Group

Guiyun Cao, IBM DB2 performance analyst
and the DB2 Toronto Lab team

Agenda

- Determining the system bottleneck
- Addressing the bottleneck
  - Types of bottlenecks
  - Refining the diagnosis
  - Responding to the problem
- Some best practices for DB2 performance

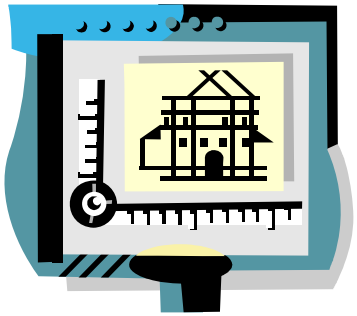
Why have this chat on DB2 Performance?

- Performance issue can occur almost anywhere in a system.
- What to do?

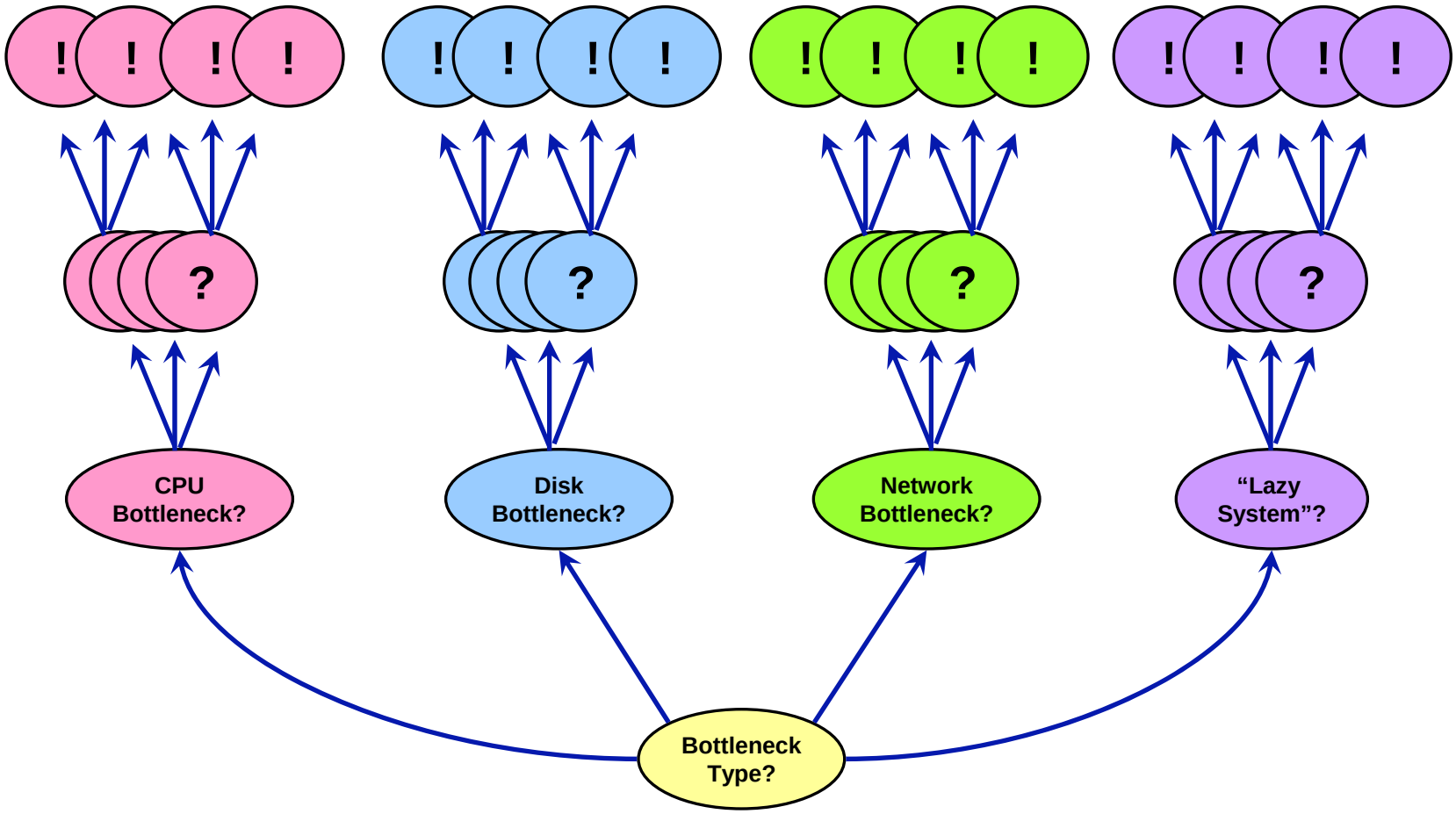
1. Panic
2. Buy more hardware
3. Blame DB2…
4. Or ….?

What Data Might You Need?

- Configuration data
  - DB & DBM configuration parameters
  - DB2 registry variables (db2set –all)
  - Schema definition with db2look
- Runtime data
  - Db2 get snapshot for all on <dbname>
  - Event monitor for statements and deadlocks
  - Statement plans (explain tables or db2expln)
  - Operating system data
    - vmstat, iostat, sar, perfmon, truss, strace, …
  - Application throughput / response time



What Bottleneck in the system?



Basic Bottlenecks

Resource Bottleneck	Basic Symptoms
Disk	▪ I/O wait seen in vmstat (wa)/ iostat / perfmon / top ▪ Disk > 80% busy seen in iostat or perfmon ▪ Long device queues seen in perfmon
CPU	▪ Total CPU utilization near 100% seen in vmstat / perfmon ▪ One process/thread steadily consuming one CPU/core
Memory	▪ Low free memory seen in vmstat (fre) / perfmon ▪ Swapping reported in vmstat (pi/po) / perfmon
Network	▪ See following pages …

vmstat sample

```
r  b avm      fre     re pi po fr  sr cy in    sy    cs   us sy id wa
2 0 1459870 35479 0  0  0   0  0  0   1587 16040 4106 24  7  15  54
1 0 1461205 34141 0  0  0   0  0  0   1670 12105 3964 20  3  22  55
```

Basic Bottlenecks

Non-Resource Bottleneck	Basic Symptoms
Locking	▪ High total lock wait time seen in DB2 snapshot ▪ Many processes in lock wait state seen in DB2 snapshot ▪ Low CPU consumption seen in vmstat
External Factors	▪ E.g. copy large files to different systems, batch jobs, etc

Disk Bottleneck - Mapping Hot Devices to DB2 Entities

- Mapping DB2 entities to filesystems (or devices)
  - Logs?

```
$ db2 get db cfg for <dbname> | grep "Path to log files"
```

  - Containers?

```
$ db2pd -d <dbname> -tablespaces

Containers:
  Address    TspId ContainNum Type    TotalPgs   UseablePgs StripeSet  Container
  0x03107BE0 0     0          File    16384      16380      0          C:\DB2\NODE0000\SAMPLE\T0000000\C0000000.CAT
  0x03108360 1     0          Path    1          1          0          C:\DB2\NODE0000\SAMPLE\T0000001\C0000000.TMP
  0x0310CB70 2     0          File    8192       8160       0          C:\DB2\NODE0000\SAMPLE\T0000002\C0000000.LRG
  0x0310D340 3     0          File    8192       8160       0          C:\DB2\NODE0000\SAMPLE\T0000003\C0000000.LRG
```

  - Utility files?
    - Load input data, load message file, backup image

- Mapping devices to filesystems

AIX example – suppose iostat shows hdisk43 is busy

```
/usr/sbin/lspv -l hdisk43
     # … shows physical volume hdisk43 is part of logical volume homelv
     # mounted on /home
```

Disk Bottleneck #1 – a Data Tablespace

- What dynamic SQL statement(s) are most active from the snapshot (then find out which tables)?

```
$ db2 select * from
 table(snapshot_dyn_sql('<dbname>',-1)) as t
 where translate(cast(substr(stmt_text,1,1024) as varchar(1024)))
 like '%<tbname>%' order by ...
```

  - Use ORDER BY clause to pick up hot statements, for example - rows_read, total_sys_cpu_time, total_usr_cpu_time, total_exec_time

- Expensive statements -> Hot table(s) -> Hot tablespace(s)

Disk Bottleneck #1 – a Data Tablespace

- What static SQL statement(s) are most active on the hot table?

```
$ db2 create event monitor e for statements write to table ….

 Statement Event ...
  Appl Handle: 7
  Appl Id: *LOCAL.DB2.100809192825
  Appl Seq number: 00002

  Record is the result of a flush: FALSE
  ------------------------------------------
  Type     : Static
  Operation: Execute
  Section  : 5
  Creator  : KCHEN
  Package  : P5125365
  Consistency Token  : ZB3MPJIa
  Package Version ID  :
  Cursor   :
  Cursor was blocking: FALSE
  ------------------------------------------
```

  - Db2expln –d <dbName> -c schemaname –p P5125365

Disk Bottleneck #1 – a Data Tablespace

- From tablespace snapshot

```
Read_time=Total buffer pool read time / (Buffer pool temporary index physical
    reads+ Buffer pool temporary data physical reads+ Buffer pool index physical
    reads+ Buffer pool data physical reads)
Write_time=Total buffer pool write time/ (Buffer pool data writes+ Buffer pool index
    writes)
```

- High Read Time
  - Snapshot rows_read >> # of executions
  - Are you getting bad plan?
    - Tablescan ? Confirm plan with db2expln
    - Are statistics out of date?
    - Better indexes ? Use DB2 Design Advisor
    - `time waited for prefetch?` increase NUM_IOSERVERS

- High Write Time
  - Are long varchars or LOBs present in the “write-hot” statement or table?
    - SMS or DMS File containers with filesystem cache for LOB
  - Need more disks?

Disk Bottleneck #2 – an Index Tablespace

- large number of index physical reads?
  - Increase index bufferpool size
- large index scans
  - large logical reads on indexes from sql snapshot

    Buffer pool index logical reads = 9050799
  - Better index ?
  - Fine tune the input parameters?

Disk Bottleneck #3 – a Temp Tablespace

- Are many sort overflows to disk?

```
Total sorts                          = 61533388
Total sort time (ms)                   = 22291323
Sort overflows                       = 41946
Buffer pool data logical reads           = 61169040778
Buffer pool data physical reads          = 1818914045
Buffer pool temporary data logical reads   = 636557325
Buffer pool temporary data physical reads  = 10556130
```

  - Tune SORTHEAP or SHEAPTHRES

- Sorts and intermediate result sets *may* be caused by bad plans
  - e.g. index ordering, group by (without the proper order in the index)

Disk Bottleneck on a Tablespace - Configuration?

- No utility is running, no hot statements
- If current disk storage is evenly / optimally configured
  - Additional capacity (more disks, etc.) may be required
- Basic rule of thumb for disk configuration
  - 16 to 20 disks per CPU (e.g. p595 16 disks/core)

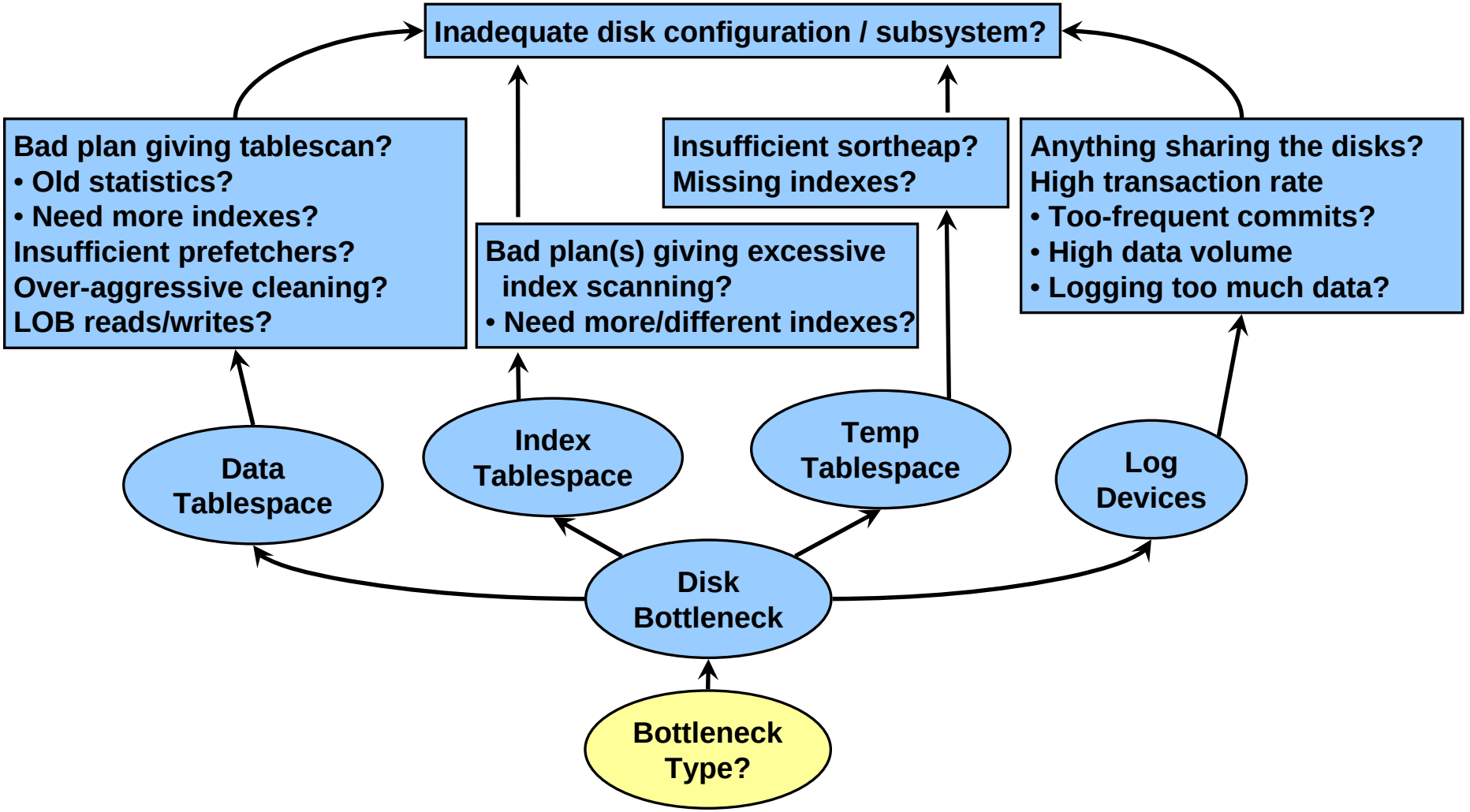
Disk Bottleneck #4 – Transaction Logs

- High Logger Write Time

```
Log pages written                = 110227630
Log write time (sec.ns)          = 16498.000000004
Number write log IOs             = 18846794
average logger write time = Log write time / Number write log IOs
Average logger size = Log pages written / Number write log IOs
```

- High transaction rate?
  - iostat shows high tps with small (e.g. 4k) avg size
  - Reduce commit frequency?
- High data volume?
  - iostat shows huge Kbps
  - Reduce amount logged?
    - Possibly use ‘not logged’ LOBS
    - Batch job use NOT LOGGED INITIALLY
- Best Practice for Log disk configuration
  - Consider RAID-10 striping multiple disks with 8k stripe size
  - Fast disks with write caching on for controller
  - Dedicated disks for OLTP
  - Avg. log write time < 1.5 ms

Disk Bottleneck Diagnosis – 10,000 feet



CPU Bottleneck – High User Time

- Is a utility saturating the system?
  - **db2 list utilities** to show what utilities are running
  - Are DB2 utility support processes consuming lots of CPU?

Utility	UNIX Processes
LOAD	**db2lmr, db2lfrm, db2lrid, db2lbm**
BACKUP	**db2bm, db2med**
RUNSTATS	**db2agent**
All	**db2agent, db2agntp** (SMP only)

There are mechanisms available to throttle DB2 utilities & free up resources for applications

- UTIL_IMPACT_PRIORTY keyword on RUNSTATS, BACKUP
- CPU_PARALLELISM keyword on LOAD

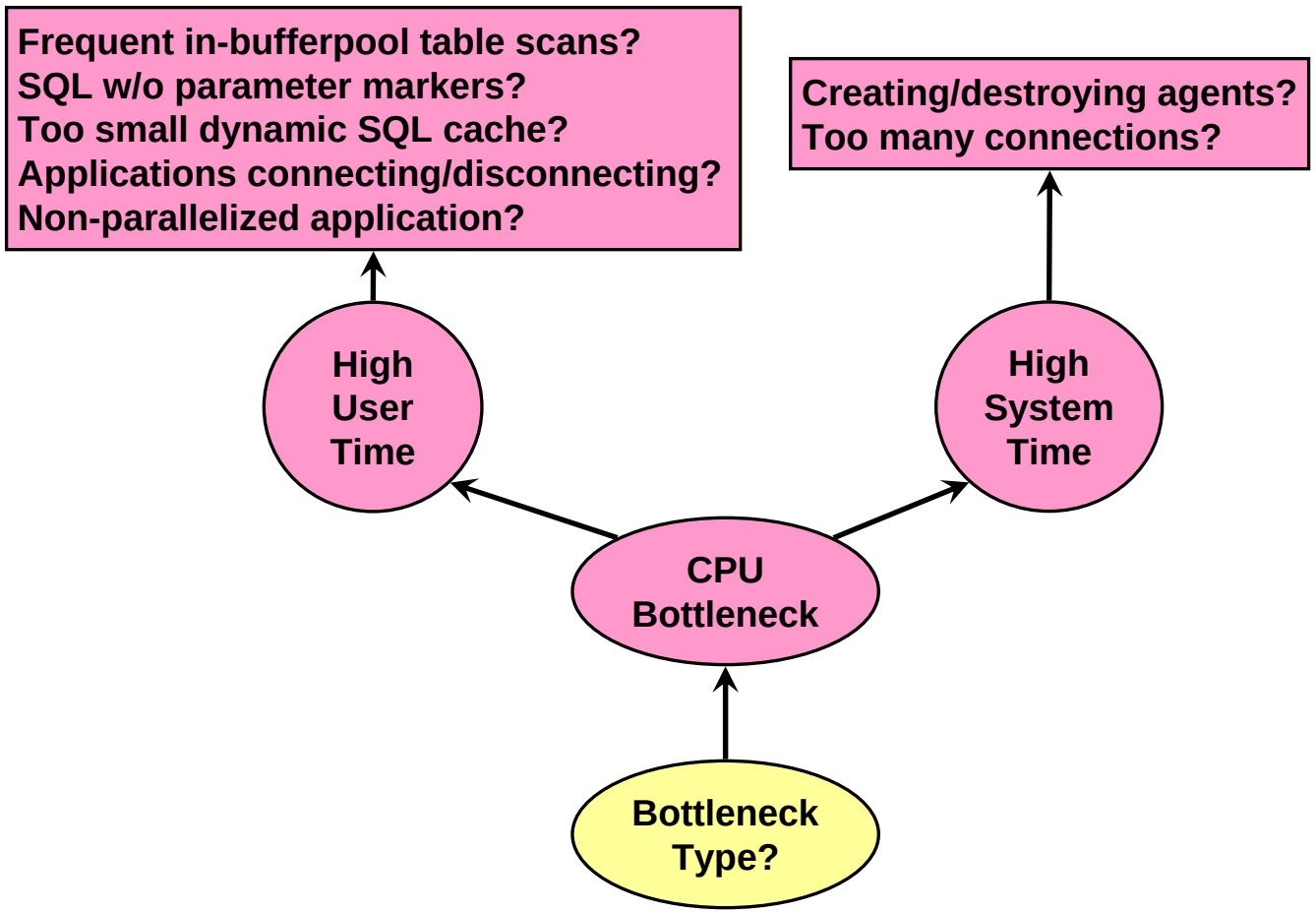
CPU Bottleneck – High User Time

- Are dynamic SQL statements being re-prepared unnecessarily?
  - dynamic SQL snapshot shows many compilations for some statements
  - Are there package cache inserts occurring?
    - Consider increasing package cache size
  - Prepared once, save the statement handle, and re-execute with new data.
- Tracking down the expensive sqls as in tablespace section

CPU Bottleneck – High System Time

- High activity in OS due to memory management
  - swap activity shown in vmstat / performance monitor?
    - Reduce memory consumption, e.g. bufferpools, sort heap
  - High Kernel time for large memory systems
    - Large page support can cut this dramatically
- Does the system have a high number of context switches?
  - CS column in vmstat > 75k or 100k / second
  - Short transactions, very frequent commits?
- Are DB2 connections coming & going frequently?
  - Possibly increase NUM_POOL_AGENTS closer to MAX_AGENTS
  - Connection pools

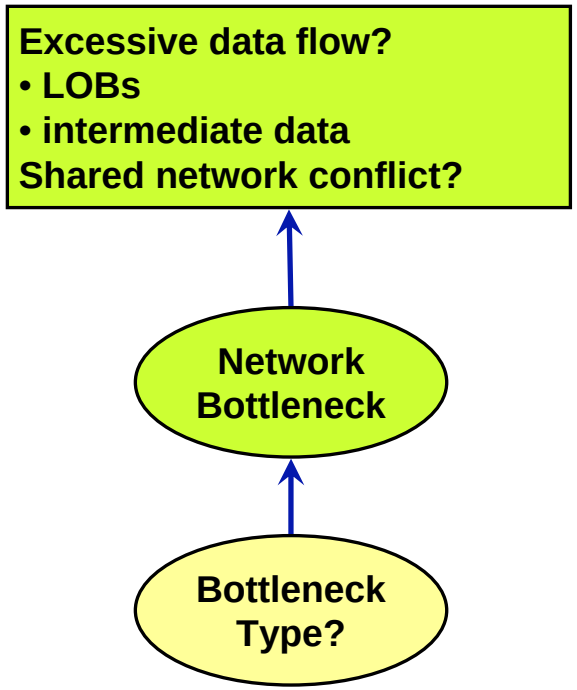
CPU Bottleneck Diagnosis – 10,000 feet



Network Bottleneck

- Not very common, but occasionally due to -
  - Very high client-server network traffic
    - Client Load utility, bulk data extraction, LOB manipulation, very high rate OLTP, …
  - Configuration issues, e.g. mismatched Ethernet transmission rates
  - External factors such as other activity on shared LAN
- High Network time?
  - Predicates to filter result set at the server

Network Bottleneck Diagnosis – 10,000 ft



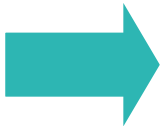
“Lazy System” Bottlenecks

- “Lazy system” – slow, but no obvious external symptoms
  - None of disk, CPU or network seems to be saturated
- Common culprit #1 – lock contention
  - Is there significant lock contention activity shown in snapshots?
    - Lock wait time?
    - Number of escalations?
    - Number of agents waiting on locks?
  - Inadequate LOCKLIST / MAXLOCKS can cause escalations

“Lazy System” Bottlenecks - Locks

- Having a convention on order of data access can help reduce deadlocks / lock waits
  - e.g., all apps access customer table first, then stock, then …
- Push selection criteria down into DB2 to minimize the number of ‘excess’ rows locked/fetched/returned ?

```
exec sql declare curs for
  select c1,c2 from t
  where c1 not null;
exec sql open curs;
do {
  exec sql fetch curs
    into :c1, :c2;
} while( P(c1) != someVar );
```



```
exec sql declare curs for
  select c1,c2 from t
  where c1 not null
  and myUdfP(c1) = :someVar;
exec sql open curs;
exec sql fetch curs
    into :c1, :c2;
```

“Lazy System” Bottlenecks – Agent I/O

- Consider increasing NUM_IOCLEANERS
  - IF the system getting large number of
  - …dirty page steal triggers?
  - …synchronous writes?

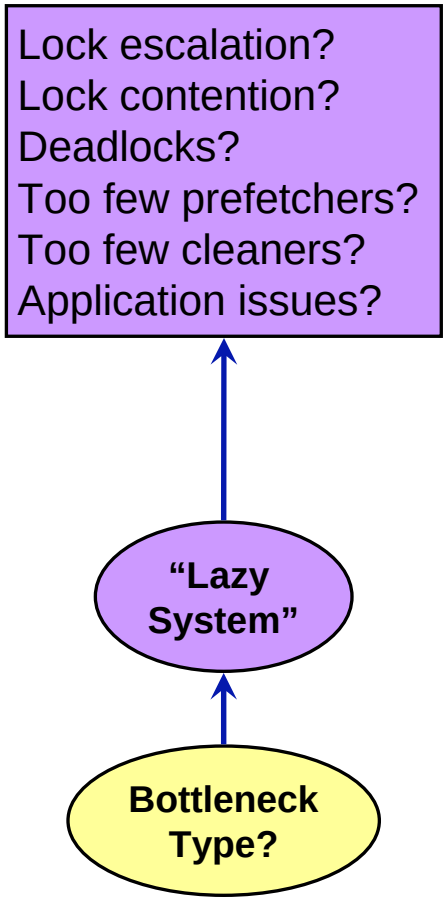
- Consider increasing NUM_IOSERVERS
  - IF there significant **`time waited for prefetch`** in database snapshot
  - IF **`buffer pool data physical reads`** in bufferpool snapshot noticeably larger than **`asynchronous pool data reads`**

Prefetchers and pagecleaners are much more efficient for I/O than DB2 agents

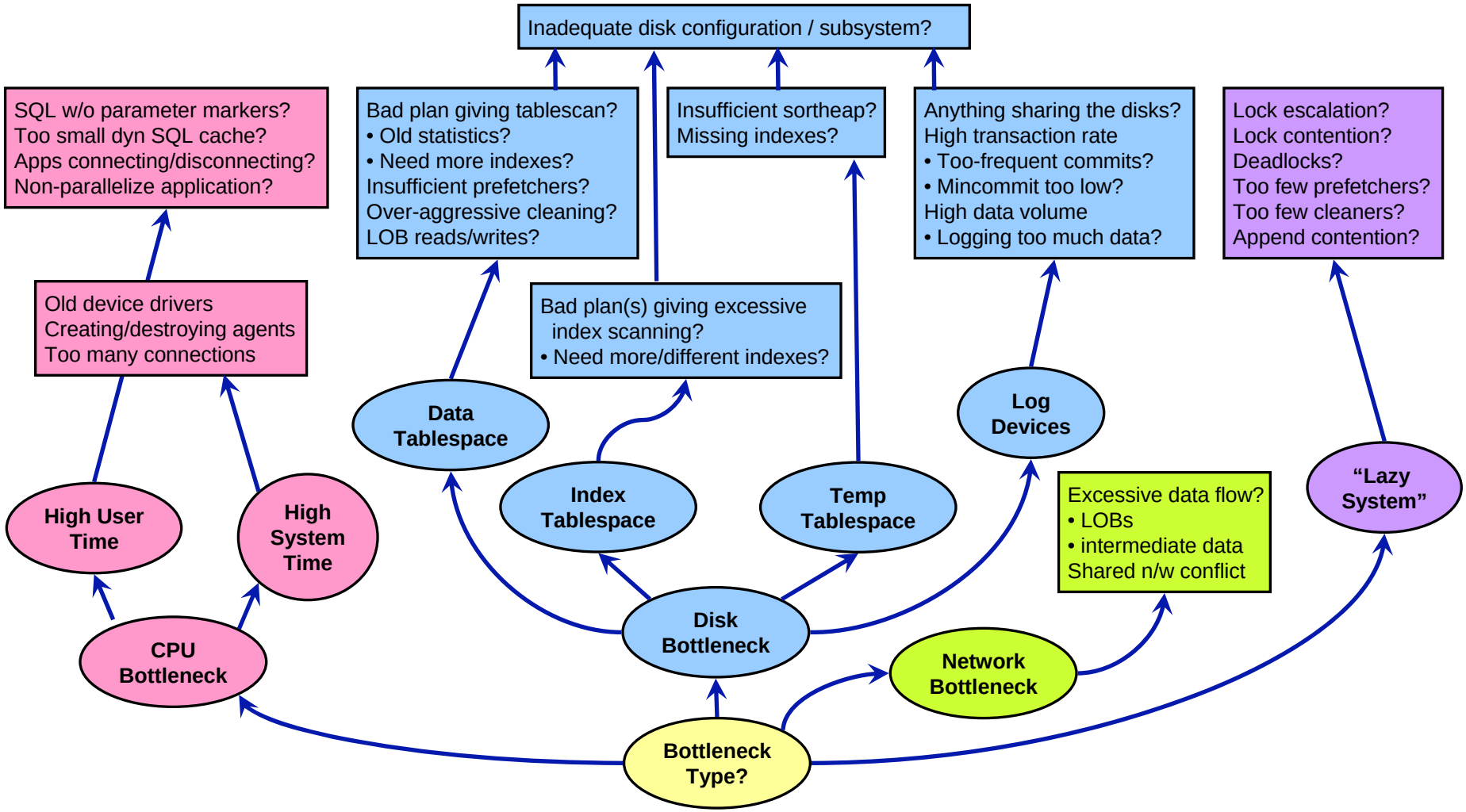
“Lazy System” Bottlenecks - Application Issues

- Is the application driving the database ‘hard enough’?
  - Does the application snapshot show that many/most DB2 agents are waiting for work (status ‘UOW waiting’)?
  - Does the event monitor show that more time is being spent on the application side than when the system was ‘healthy’?
- Examine application & client side for bottlenecks
  - Possibly increase application parallelism
    - more connections, more work in parallel

“Lazy System” Bottleneck Diagnosis – 10,000 ft



And the Whole Thing …



Best Practices

- Monitoring
  - Collect basic operational monitoring data regularly, so that background information is available in case of a problem
  - Monitor non-DB2 metrics, such as CPU utilization (nmon on AIX, iostat and vmstat, perfmon on windows) and application-level response time
  - Keep track of changes in configuration and environment settings.

Best Practices

- Troubleshooting
  - Start with the highest-level symptoms—such as a CPU, disk, or memory bottleneck— vmstat
  - Then track down hot statements from snapshot
  - Drill down into possible causes, refining with each step; for example, inefficient statement S -> table T -> Tablespace container C
  - Tune the system step by step —change only one thing at a time, and observe the result carefully

Questions?

Appendix

- **DB2 Information**
  - **Best Practices: Tuning and Monitoring Database System Performance**
    http://www.ibm.com/developerworks/data/bestpractices/systemperformance/
  - **Best Practices: Writing and Tuning Queries for Optimal Performance**
    http://www.ibm.com/developerworks/data/bestpractices/querytuning/
- **Feedback**
  - Presentation format and contents
  - Additional DB2 topics you are interested
  - Follow on questions for the presentation
- **Contact: fning@ca.ibm.com (Frank Ning)**