Wait-Time Based Performance Management

Janis Griffin
Senior DBA, Confio Software
Who Am I?

- Senior DBA/Engineer for Confio Software
  - JanisGriffin@confio.com
- DBA – 20+ Years, Oracle, Sybase, SqlServer
- Former - 15 Years in Telecom Industry, 5 Years as Database Design / Implementation Consultant
- Specialize in performance tuning using Wait Events
- Review performance of Oracle databases at many customers sites
Agenda

- Methodology
- Case Study One: Hot Block Issue
- Case Study Two: Full Table Scans
- Case Study Three: Inefficient Indexes
- Q&A
Before we start, do you know…

- The most important problem in your database?
- Did the vendor really fix the problem in that patch?
- Which bottlenecks in your database directly impacted end-user service?
Conventional Tools Measure Database Health

Unclear view of performance leads to finger pointing.

Wait Time Tuning vs. Ratios
Database processing includes hundreds of steps

- Identify Wait Time at every step
- Rank bottlenecks by impact on end user
- Cashier is the CPU
- Customer being checked out is “running”
- Customers waiting in line are “runnable”
- Customer 1 Requires Price Check
  - Customer 1 “waits” on “Price Check”
  - Customer 2 is checked out, i.e. “running”
  - Customer 3 is “runnable”
- Price Check is Completed
  - Customer 1 goes to “runnable”
Execution Model

CPU 1
SPID 60 – Running

CPU 1 Queue
SPID 51 – Runnable
SPID 61 – Runnable

Waiter List
SPID 52 -- Sql*Net more data from client
SPID 53 -- latch: cache buffers chains
SPID 54 -- db file scattered read
SPID 57 -- enq: TX row lock contention
SPID 59 -- log file sync
Four Key Principles of PI

1. **SQL View:** All statistics and information at SQL statement level

2. **Time View:** Measure **Time**, not number of times a resource is used.

3. **Full View:** Measure every **wait** individually to isolate source of problems

4. **Historical View:** Store data long term to spot trends, anomalies, relationships and easier analytics
### V$SESSION_WAIT (X$KSUSECST)
- SID
- EVENT
- P1, P1RAW, P2, P2RAW, P3, P3RAW
- STATE

- Oracle 10g added this info to V$SESSION

### V$SESSION (X$ksuse)
- SID
- USERNAME
- SQL_ID
- PROGRAM
- MODULE
- ACTION
- PLAN_HASH_VALUE

### V$SQLAREA
- SQL_ID
- EXECUTIONS
- PARSE_CALLS
- BUFFER_GETS
- DISK_READS

### V$SQL_PLAN
- SQL_ID
- PLAN_HASH_VALUE

### DBA_OBJECTS
- OBJECT_ID
- OBJECT_NAME
- OBJECT_TYPE
Oracle Wait Events

- **V$SESSION_WAIT (X$KSUSECST)**
  - SID (join to v$session)
  - EVENT
  - P1, P1RAW, P2, P2RAW, P3, P3RAW
    (Definition of parameters in V$EVENT_NAME)
  - STATE = ‘WAITING’ – currently waiting on event
  - STATE = ‘WAITED…’ – currently on CPU (or in queue)

- Oracle 10g added this info to V$SESSION
V$SESSION (X$KSUSE)

- SID
- USERNAME
- SQL_ID – parent id
  - Join to V$SQL
- PROGRAM
- MODULE / ACTION
  - DBMS_APPLICATION_INFO
- PLAN_HASH_VALUE
  - Join to V$SQL_PLAN
SELECT
    sid, username, program, module, action,
    machine, osuser, ...
    sql_id, plan_hash_value,
    decode(state, 'WAITING', event, 'CPU') event,
    p1, p1raw, p2, ...
    SYSDATE
FROM V$SESSION s
WHERE s.status = 'ACTIVE'
AND event NOT IN (<idle wait events>);
-- (AND wait_class != 'Idle')
SQL Text & Other Information

• **V$SESSION**
  - `service_name`, `machine`, `client_info`
  - `row_wait_obj#`, `blocking_session`
  - `prev_sql_id`

• **Go back later to get**
  - `Sql_text` from `v$sql`
  - SQL stats from `v$sqlarea`
  - Execution plan from `v$sql_plan`
  - Object Name & Object Type from `DBA_objects`
### V$ACTIVE_SESSION_HISTORY
- Data warehouse for session statistics
- Oracle 10g and higher
- Data is sampled every second
- Holds at least one hour of history
- Never bigger than:
  - 2% of SGA_TARGET
  - 5% of SHARED_POOL (if automatic sga sizing is turned off)

### WRH$ACTIVE_SESSION_HISTORY
- Above table gets flushed to this table
Compliant Tool Types

- **Two Primary Types of Tools**

- **Session Specific Tools**
  - Tools that focus on one session at a time often by tracing the process
  - Examples: OraSRP Profiler (open source), Hotsos Profiler, tkprof

- **Continuous DB Wide Monitoring Tools**
  - Tools that focus on all sessions by sampling Oracle
  - Examples: Confio Ignite, Precise, OEM

- Both tools have a place in the organization
Tracing with waits complies

- High Overhead
- Point in time data only

Use cautiously due to session statistics skew

- 95 of 100 sessions are running well
- 5 out of 100 have spent 99% of time waiting for locked rows
- If you trace one of the “95” sessions, it appears as if you have no locking issues (and spend time trying to tune other items that may not be important)
- If you trace one of the “5” sessions, it appears as if you could fix the locking problems and reduce your wait time by 99%
- Very precise - may be only way to get some statistics
- Variable information is available
- Can provide detailed analysis even deeper than just waits
- Ideal if a known problem is going to occur in the future
- Difficult to see trends over time
Continuous Monitoring Tools

- 24/7 sampling provides real-time and historical perspective
- Allows DBA to go back in time
  - I had a problem at 3:00 pm yesterday
- Not the level of detail provided by tracing
- Most of these tools have trend reports that allow communication with other groups
  - What is starting to perform poorly?
  - What progress have we made while tuning?
950 + wait events in 11.1
13 wait classes

850 + wait events in 10.2
12 wait classes

400 + wait events in 9.2
n/a
Top Wait Time (52 Customers)

- db file sequential read: 28%
- db file scattered read: 27%
- CPU: 12%
- direct path read / write: 11%
- buffer busy waits: 5%
- log file sync: 3%
- library cache lock: 2%
- log buffer space: 2%
Case Study I
Hot Block Issue
Problem Observed

- Critical situation: application performance unsatisfactory
  - All email coming into and going out of the company was tracked in order to find:
    - Viruses
    - Espionage
    - Legal reasons
  - However, email was getting behind
  - Email not getting to end-users for several hours
  - Declared top priority in company
Wait Events During Problem

Top SQLs causing user wait time (DOUBLE-CLICK BARS FOR DETAILS)

- Query that is doing “real” work
- Buffer busy waits
- Log file waits

Seconds of waiting

<table>
<thead>
<tr>
<th>SQL Hash Values</th>
<th>Seconds of waiting</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDL or commits</td>
<td>600000</td>
</tr>
<tr>
<td>Email Status</td>
<td>500000</td>
</tr>
<tr>
<td>Email Sent</td>
<td>400000</td>
</tr>
<tr>
<td>OEM Used Space</td>
<td>300000</td>
</tr>
<tr>
<td>OEM Free Space</td>
<td>200000</td>
</tr>
<tr>
<td>10537955750</td>
<td>100000</td>
</tr>
</tbody>
</table>

Wait Events:
- db file sequential read
- log file switch completion
- buffer busy waits
- enqueue
- direct path read
- session allocation
- log file sync
- library cache

Wait Types:
- free buffer waits
- undo segment extension
- async disk I/O
- db file scattered read
- redo allocation
- enqueue hash chains
- row cache objects

Other Wait Events:
- write complete waits
- cache buffers chains
- direct path write
- bdbms ipc reply
- CPU
- cache buffers lru chain
- shared pool
- control file sequential read
<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Which SQL:</td>
<td>DDL or Commits SQL hash_value=0</td>
</tr>
<tr>
<td>Which Resource:</td>
<td>buffer busy waits log file waits</td>
</tr>
<tr>
<td>How much time:</td>
<td>163 Hours of wait time per day</td>
</tr>
</tbody>
</table>
Buffer is being read into cache by another session and this session is waiting for that process to complete.

- In Oracle 10g buffer busy waits are further refined and this becomes “read by other session”

Buffer is already in the cache but in an incompatible mode, i.e. another session is changing it.
P1 – file number information
P2 – block number information

SELECT owner, segment_name, segment_type
FROM dba_extents
WHERE file_id = &P1
AND &P2 BETWEEN block_id AND block_id + blocks -1

Gives information about the object being waited for
“buffer busy waits” Analysis
Results

- Found hot block problem
  - “buffer busy waits” was waiting for **Block #2** in the file “...staging01.dbf”
  - The email processing code was creating a series of staging tables, every time it executed

- Solutions
  - Started using temporary tables vs. create/drop distinct tables each time the process ran
Case Study II

DB File Scattered Reads
Problem Observed

- Problem: Login taking 12 minutes for each user when they started their day
  - High wait accumulation from 6:30 – 8:30 am
  - 240 Users X 12 Minutes = 48 Hours Every Day
  - 6 employees wasted time per day
  - $400,000+ wasted per year

- Applied PI methods for problem identification
  - Identify Wait Time, offending SQL, offending Resource
Wait Events During Problem

HQPADDP1_CPAMHQ_ORC01
Top 15 Waits by Total Daily Wait
September 7, 2005 - September 29, 2005

- db file scattered read
- buffer busy waits
- db file sequential read
Investigation

Top SQLs causing user wait time
(DOUBLE CLICK BARS FOR DETAILS)

Seconds of waiting

SQL Hash Values

- LoginLookup
- UpdateInventory
- uNMotionViewers
- DDL or commits
- 150306737
- 74603140
- 3308153218
- 819321068

Seconds of waiting

- async disk I/O
- control file sequential read
- control file single write
- db file sequential read
- row cache objects
- cache buffers chains
- db file scattered-read
- buffer busy waits
- multiblock read objects
- control file parallel write
- log file sync

34
What do we know?

- **Which SQL:**
  - LoginLookup
  - UpdateInventory

- **Which Resource:**
  - scattered read
  - buffer busy waits

- **How much time:**
  - 48+ Hours
  - Every Day
Hypotheses: Oracle Interpretations

Key Questions:

1. Is full table scan necessary? (7–10%)
2. What causes a full table scan for this SQL Statement?

Two Alternative paths for optimization:

I. Eliminate Full Table Scan
   1. Add Index
   2. Update Statistics
   3. Utilize Query Hints

II. Full Table Scan Required - Improve response time
    1. Parallelized Reads
    2. Optimize I/O Subsystem
    3. Optimize Application
Solutions:

1. Add / Modify index(es) on the table
2. Update table and/or index statistics if proper index not being used
3. Add hint to use existing index
Solutions:

1. Use Parallel Reads
2. Set Database Parameters
3. Improve I/O Speed
4. Optimize the application
- Added indexes to underlying tables
- Added Materialized View
Case Study III
DB File Sequential Reads
Problem Observed

- Data Warehouse loads were taking too long
- Noticed high wait times on “db file sequential read” wait event
- DBAs were confused – why are data loads “reading” data
- Applied PI Method for problem identification
  - Identify Wait Time, offending SQL, offending Resource
Investigation for an INSERT SQL

Sequential read time by object for SQL
What does PI tell us?

- **Which SQL:** Load Process
- **Which Resource:** DB File Sequential Read
- **How much time:** 5 hour+
  - 90% of wait time
Investigating db file sequential reads

- Often considered a “good” read
- DB file sequential reads normally occur during index lookups
- Often a single-block read although it may retrieve more than one block.
  - P1 – file id
  - P2 – block id
  - Join to DBA_EXTENTS (see buffer busy waits)
Causes of excessive wait times:

I. Reading too many index leaf blocks
II. Low cardinality first column index
III. Not finding block in buffer cache forces disk read
IV. Slow disk reads
V. Contention for certain blocks
VI. High Read time on INSERT statements
Many sessions were loading data and all were updating low cardinality indexes

Modified index and noticed a 50% performance improvement

Customer is also analyzing global vs. local indexes

Reviewing usage of bitmap indexes

Removed unused indexes

Enhanced the disk subsystem
Conventional Tuning focuses on “system health” and can lead to finger-pointing and confusion.

Wait event tuning implemented according to PI Methods is the best way to tune:

- Continuous DB-wide monitoring tool
- 4 Key Principles
  - sql, time, resource (wait event), historical views

Questions & Answers
Confio Software

- Developer of Performance Tools
- Igniter Suite
  - Ignite for SQL Server, Oracle, DB2, Sybase
- Packaged, easy-to-use implementation of Performance Intelligence (PI)
- Based in Colorado, worldwide customers

- Free trial at www.confio.com