CONFIO SOFTWARE

Ignite IT Performance™

Tuna Helper
Proven Process for SQL Tuning

Dean Richards
Senior DBA, Confio Software

Tuna Helper – Proven Process for SQL Tuning

Give a man a fish and you feed him for a day.
Teach a man to fish and you feed him for a lifetime.
Chinese Proverb

Who Am I?

CONFIO SOFTWARE
Ignite IT Performance™

- Senior DBA for Confio Software
  - DeanRichards@confio.com
- Current – 20+ Years in Oracle, SQL Server
- Former – 15+ Years in Oracle Consulting
- Specialize in Performance Tuning
- Review Performance of 100's of Databases for Customers and Prospects
- Common Thread – Paralyzed by Tuning

Agenda

CONFIO SOFTWARE
Ignite IT Performance™

- Introduction
- Challenges
- Identify - Which SQL and Why
- Gather – Details about SQL
- Tune – Case Study
- Monitor – Make sure it stays tuned
- SQL Tuning Myths

Introduction

CONFIO SOFTWARE
Ignite IT Performance™

- SQL Tuning is Hard
- This Presentation is an Introduction
  - 3-5 day detailed classes are typical
- Providing a Framework
  - Helps develop your own processes
  - There is no magic tool
  - Tools cannot reliably tune SQL statements
  - Tuning requires the involvement of you and other technical and functional members of team

Challenges

CONFIO SOFTWARE
Ignite IT Performance™

- SQL Tuning is Hard – did I mention that?
- Requires Expertise in Many Areas
  - Technical – Plan, Data Access, SQL Design
  - Business – What is the Purpose of SQL?
- Tuning Takes Time
  - Large Number of SQL Statements
  - Each Statement is Different
- Low Priority in Some Companies
  - Vendor Applications
  - Focus on Hardware or System Issues
- Never Ending

Identify – Which SQL

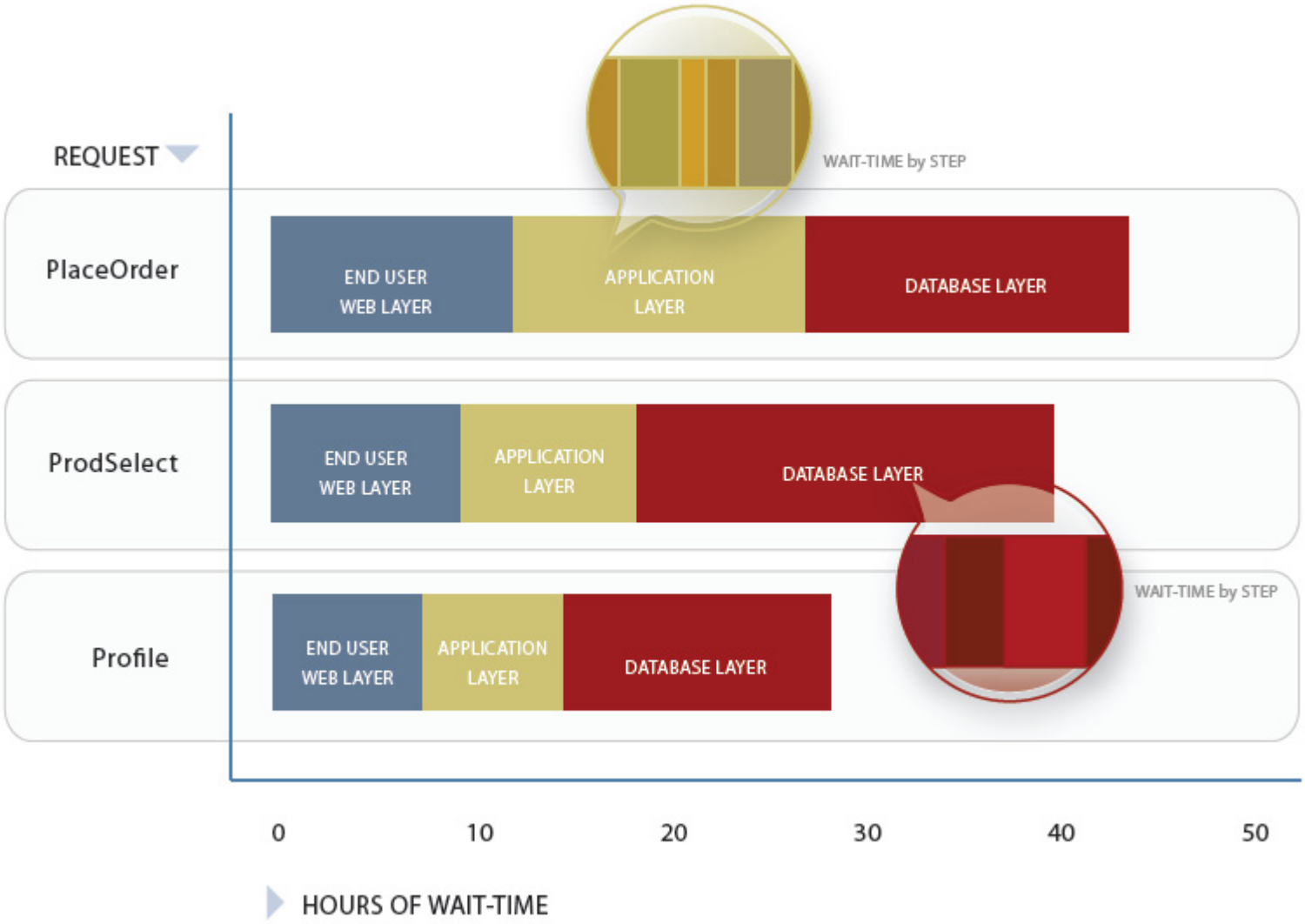
- Highest Wait Times (Ignite, AWR, etc)
- Tracing a Session / Process
- User / Batch Job Complaints
- Highest I/O (LIO, PIO)
- SQL Performing Full Table Scans
- Known Poorly Performing SQL

Identify – End-to-End

CONFIO SOFTWARE
Ignite IT Performance™

- Business Aspects
  - Who registered yesterday for SQL Tuning
  - Why does the business need to know this
  - How often is the information needed
  - Who uses this information
- Technical Information
  - Review ERD
  - Understand tables and the data (at a high level)
- End-to-End Process
  - Understand application architecture
  - What portion of the total time is database
  - Where is it called from in the application

Identify – End-to-End Time



Wait Event Information

CONFIO SOFTWARE
Ignite IT Performance™

V$SESSION

SID
USERNAME
SQL_ID
PROGRAM
MODULE
ACTION
PLAN_HASH_VALUE

V$SESSION_WAIT

SID
EVENT
P1, P1RAW, P2, P2RAW, P3, P3RAW
STATE (WAITING, WAITED…)

• Oracle 10g added this info to V$SESSION

V$SQL

SQL_ID
SQL_FULLTEXT

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

Wait Time Scenario

CONFIO SOFTWARE
Ignite IT Performance™

- Which scenario is worse?
- SQL Statement 1
  - Executed 1000 times
  - Caused 10 minutes of wait time for end user
  - Waited 99% of time on "db file sequential read"
- SQL Statement 2
  - Executed 1 time
  - Caused 10 minutes of wait time for end user
  - Waited 99% on "enq: TX – row lock contention"

Identify – Simplification

- Break Down SQL Into Simplest Forms
  - Complex SQL becomes multiple SQL
  - Sub-Queries Should be Tuned Separately
  - UNION'ed SQL Tuned Separately
  - Get the definition of views
  - Are synonyms being used

Identify – Summary

CONFIO SOFTWARE
Ignite IT Performance™

- Determine the SQL
- Understand End-to-End
- Measure Wait Time
- Simplify Statement

Gather - Metrics

CONFIO SOFTWARE
Ignite IT Performance™

- Get baseline metrics
  - How long does it take now
  - What is acceptable (10 sec, 2 min, 1 hour)
- Collect Wait Time Metrics – How Long
  - Locking / Blocking
  - I/O problem
  - Latch contention
  - Network slowdown
  - May be multiple issues
  - All have different resolutions
- Document everything in simple language

Gather – Execution Plan

- EXPLAIN PLAN
  - Estimated execution plan - can be wrong for many reasons
- V$SQL_PLAN (Oracle 9i+)
  - Real execution plan
  - Use DBMS_XPLAN for display
- Tracing (all versions)
  - Get all sorts of good information
  - Works when you know a problem will occur
- Historical – AWR, Confio Ignite

Plans Not Created Equal

CONFIO SOFTWARE
Ignite IT Performance™

```
SELECT company, attribute
FROM data_out WHERE segment = :B1
```

- Wait Time – 100% on "db file scattered read"
- Plan from EXPLAIN PLAN

```
SELECT STATEMENT Optimizer=ALL_ROWS (Cost=1 Card=1 Bytes=117)
   TABLE ACCESS (BY INDEX ROWID) OF 'DATA_OUT' (TABLE) (Cost=1 Card=1 Bytes=117)
     INDEX (UNIQUE SCAN) OF 'IX1_DATA_OUT' (INDEX (UNIQUE)) (Cost=1 Card=1)
```

- Plan from V$SQL_PLAN using DBMS_XPLAN

```
select * from table(dbms_xplan.display_cursor('az7r9s3wpqg7n',0));

-----------------------------------------------------------------------------------
| Id  | Operation          | Name     | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |          |       |       |   370 (100)|          |
|*  1 |  TABLE ACCESS FULL| DATA_OUT |     1 |   117 |   370   (4)| 00:00:05 |
-----------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter(TO_BINARY_DOUBLE("SEGMENT")=:B1)
```

Gather – Bind Values

- V$SQL_BIND_CAPTURE
  - STATISTICS_LEVEL = TYPICAL or ALL
  - Collected at 15 minute intervals

```
SELECT name, position, datatype_string, value_string
FROM   v$sql_bind_capture
WHERE  sql_id = '15uughacxfh13';

NAME    POSITION DATATYPE_STRING VALUE_STRING
----- ---------- --------------- ------------
:B1             1 BINARY_DOUBLE
```

- Bind Values also provided by tracing
  - Level 4 – bind values
  - Level 8 – wait information
  - Level 12 – bind values and wait information

Gather – Table / Index Stats

CONFIO SOFTWARE
Ignite IT Performance™

- Provides data on objects in execution plans.
  - Table sizes
  - Existing indexes
  - Cardinality of columns
  - Segment sizes
  - Histograms and Data Skew
  - Many things the CBO uses
- Use TuningStats.sql
  - http://support.confio.com/kb/1534
- Run it for expensive data access targets

Example SQL Statement

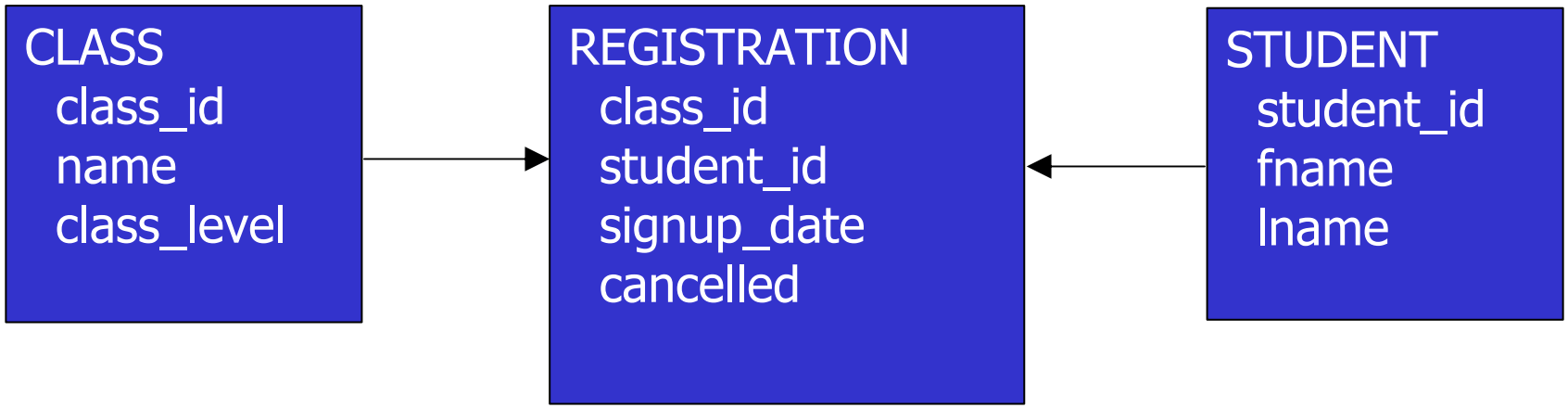
CONFIO SOFTWARE
Ignite IT Performance™

- Who registered yesterday for SQL Tuning

```
SELECT s.fname, s.lname, r.signup_date
FROM   student s, registration r, class c
WHERE  s.student_id = r.student_id
AND    r.class_id     = c.class_id
AND    UPPER(c.name)  = 'SQL TUNING'
AND    c.class_level  = 101
AND    r.signup_date BETWEEN
          TRUNC(SYSDATE-1) AND TRUNC(SYSDATE)
AND    r.cancelled = 'N`
```

- Execution Time – 12:38
- Wait Time – 95% on “db file scattered read”

Relationship



Gather – Summary

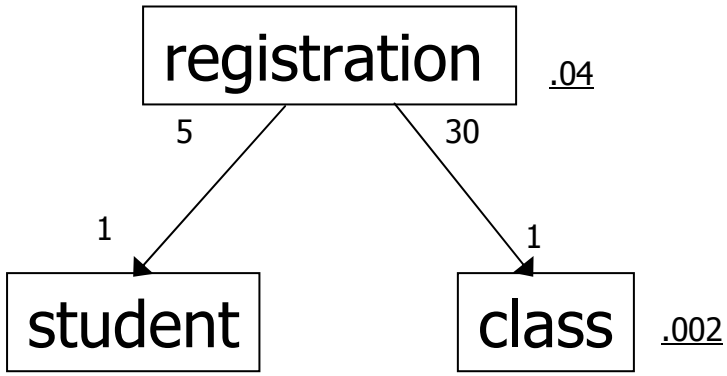
CONFIO SOFTWARE
Ignite IT Performance™

- Execution Plan
  - V$SQL_PLAN
  - Do not use EXPLAIN PLAN
  - DBMS_XPLAN
- Bind Values
  - V$SQL_BIND_CAPTURE
  - Tracing
- Table and Index Statistics
- ERD

Tune – Create SQL Diagram

CONFIO SOFTWARE
Ignite IT Performance™

- SQL Tuning – Dan Tow
  - Great book that teaches SQL Diagramming
  - http://www.singingsql.com




```
select count(1) from registration where cancelled = 'N'
and signup_date between trunc(sysdate-1) and trunc(sysdate)

3562 / 80000 = .0445

select count(1) from class where name = 'SQL TUNING'

2 / 1000 = .002
```

Execution Plan

CONFIO SOFTWARE
Ignite IT Performance™

```
-------------------------------------------------------------------------------
| Id  | Operation                     | Name         | Rows  | Bytes | Cost  |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |              |       |       |    95 |
|   1 |  NESTED LOOPS                 |              |     1 |   167 |    95 |
|   2 |   NESTED LOOPS                |              |     1 |   138 |    94 |
|   3 |    NESTED LOOPS               |              |     7 |   357 |    87 |
|   4 |     VIEW                      | VW_SQ_1      |   201 |  7035 |    87 |
|*  5 |      FILTER                   |              |       |       |       |
|   6 |       HASH GROUP BY           |              |   201 |  3417 |    87 |
|*  7 |        FILTER                 |              |       |       |       |
|*  8 |         TABLE ACCESS FULL     | REGISTRATION | 80000 | 1328K |    76 |
|*  9 |     INDEX UNIQUE SCAN         | SYS_C0036920 |     1 |    16 |     0 |
|* 10 |    TABLE ACCESS BY INDEX ROWID| CLASS        |     1 |    87 |     1 |
|* 11 |     INDEX UNIQUE SCAN         | SYS_C0036919 |     1 |       |     0 |
|  12 |   TABLE ACCESS BY INDEX ROWID | STUDENT      |     1 |    29 |     1 |
|* 13 |    INDEX UNIQUE SCAN          | SYS_C0036918 |     1 |       |     0 |
-------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   5 - filter((MAX("SIGNUP_DATE")>=TRUNC(SYSDATE@!-1) AND
              MAX("SIGNUP_DATE")<=TRUNC(SYSDATE@!)))
   7 - filter(TRUNC(SYSDATE@!-1)<=TRUNC(SYSDATE@!))
   8 - filter("CANCELLED"='N')
   9 - access("R1"."STUDENT_ID"="STUDENT_ID" AND "R1"."CLASS_ID"="CLASS_ID" AND
              "SIGNUP_DATE"="VW_COL_1")
       filter(("SIGNUP_DATE">=TRUNC(SYSDATE@!-1) AND "SIGNUP_DATE"<=TRUNC(SYSDATE@!)))
  10 - filter(("C"."CLASS_LEVEL"=101 AND UPPER("C"."NAME")='SQL TUNING'))
  11 - access("CLASS_ID"="C"."CLASS_ID")
  13 - access("S"."STUDENT_ID"="STUDENT_ID")
```

REGISTRATION Table Data

CONFIO SOFTWARE
Ignite IT Performance™

```
Name                           Null?    Type
------------------------------ -------- ----------------
STUDENT_ID                     NOT NULL NUMBER
CLASS_ID                       NOT NULL NUMBER
SIGNUP_DATE                    NOT NULL DATE
CANCELLED                               CHAR(1)


INDEX_NAME                     UNIQUENES COLUMN_NAME     COLUMN_POSITION
------------------------------ --------- --------------- ---------------
SYS_C0020876                   UNIQUE    STUDENT_ID                    1
SYS_C0020876                   UNIQUE    CLASS_ID                      2
SYS_C0020876                   UNIQUE    SIGNUP_DATE                   3


COLUMN_NAME                  NUM_DISTINCT  NUM_NULLS    DENSITY SAMPLE_SIZE
---------------------------- ------------ ---------- ---------- -----------
CANCELLED                               2          0         .5        5443
CLASS_ID                              998          0 .001002004        5443
SIGNUP_DATE                         32817          0 .000030472       79983
STUDENT_ID                           9999          0  .00010001       79983
```

New Plan

CONFIO SOFTWARE
Ignite IT Performance™

create index cl_unm on class(upper(name))

```
-------------------------------------------------------------------------------
| Id  | Operation                          | Name          | Rows  | Bytes | Cost |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                   |               |       |       |   10|
|*  1 |  FILTER                            |               |       |       |     |
|   2 |   NESTED LOOPS                     |               |     1 |   132 |    7|
|   3 |    NESTED LOOPS                    |               |     1 |   103 |    6|
|*  4 |     TABLE ACCESS BY INDEX ROWID    | CLASS         |     1 |    87 |    5|
|*  5 |      INDEX RANGE SCAN              | CL_UNAME      |     4 |       |    1|
|*  6 |     INDEX RANGE SCAN               | REG_ALT       |     1 |    16 |    1|
|   7 |      SORT AGGREGATE                |               |     1 |    17 |     |
|*  8 |       TABLE ACCESS BY INDEX ROWID| REGISTRATION  |     1 |    17 |    3|
|*  9 |        INDEX RANGE SCAN            | REG_ALT       |     1 |       |    2|
|  10 |    TABLE ACCESS BY INDEX ROWID     | STUDENT       |     1 |    29 |    1|
|* 11 |     INDEX UNIQUE SCAN              | SYS_C0036918  |     1 |       |    0|
-------------------------------------------------------------------------------
```

Query 2

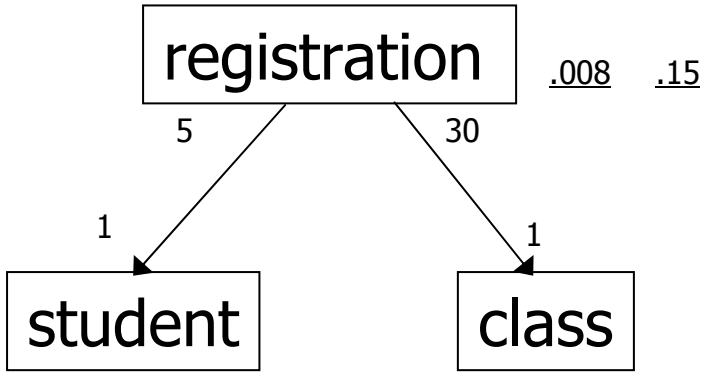
CONFIO SOFTWARE
Ignite IT Performance™

- Who cancelled classes within the week

```
SELECT s.lname, c.name, r.signup_date cancel_date
FROM   registration r, student s, class c
where  r.signup_date between sysdate and sysdate-7
AND    r.cancelled  = 'Y'
AND    r.student_id = s.student_id
AND    r.class_id   = c.class_id
```

- 30% of rows are dated within last week
- No index on CANCELLED column = FTS
- Will an index on CANCELLED column help?
  - Why or why not?

Tune – Create SQL Diagram




```
select count(1) from registration where cancelled = 'Y'

638 / 80000 = .0079

select count(1) from registration
where signup_date between trunc(sysdate-1) and trunc(sysdate)

11598 / 80000 = .1449

select count(1) from registration where cancelled = 'Y'
and signup_date between trunc(sysdate-1) and trunc(sysdate)

622 / 80000 = .0077
```

Query 2 Column Stats

CONFIO SOFTWARE
Ignite IT Performance™

```
create index reg_can on registration(cancelled);

select cancelled, count(1)
from registration group by cancelled;

C   COUNT(1)
- ----------
Y        638
N      79345
```

- Oracle will not use an index on this column
  - Unless it has more information
  - CBO assumes an even data distribution
- Histograms give more information to Oracle
  - Based on skewed data, CBO realizes an index would be beneficial
  - Works best with literal values
  - Bind Variables – Oracle peeks first time only

Query 2 - Histogram

```
dbms_stats.gather_table_stats(
    ownname    => 'STDMGMT',
    tabname    => 'REGISTRATION',
    method_opt=>'FOR COLUMNS cancelled SIZE AUTO')
```

```
-------------------------------------------------------------------------------
| Id  | Operation                    | Name         | Rows  | Bytes | Cost |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |              |       |       |     7|
|*  1 |  FILTER                      |              |       |       |      |
|*  2 |   TABLE ACCESS BY INDEX ROWID| REGISTRATION |     1 |    17 |     7|
|*  3 |    INDEX RANGE SCAN          | REG_CAN      |   754 |       |     2|
-------------------------------------------------------------------------------
```

Monitor

CONFIO SOFTWARE
Ignite IT Performance™

- Monitor the improvement
  - Be able to prove that tuning made a difference
  - Take new metrics measurements
  - Compare them to initial readings
  - Brag about the improvements – no one else will
- Monitor for next tuning opportunity
  - Tuning is iterative
  - There are always room for improvements
  - Make sure you tune things that make a difference
- Shameless Product Pitch - Ignite

Ignite for SQL Server

CONFIO SOFTWARE
Ignite IT Performance™

ignite performance intelligence

Main | Real Time | Problems | Options | Reports | Log | Logoff newdemo

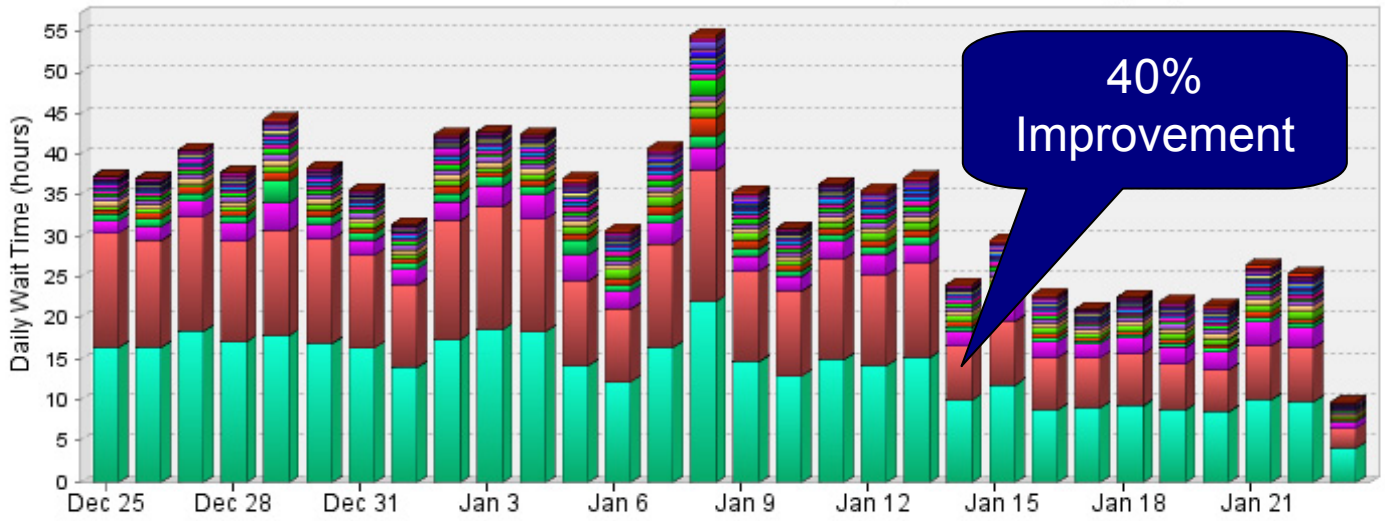
Day: All Days

Show Tips

Home > SQL Server > SQLPROD02(SQL Server) > **Wait Analysis**

SQL | Waits | Programs | Databases | Machines | DB Users

Top 20 SQL Statements by Total Daily Wait | SQLPROD02
December 25, 2008 to January 23, 2009



- Search Results - recompile
- tblCdbPeople
- Retrieve Trace Data
- Recent Activity
- Select Member List
- Basic Search
- Client Stats Update
- Validate Server Name
- Return
- 2335193507
- 5279743320
- 3872423950
- 4748133063
- 3835360828
- 6056390
- DDL or c
- 4822915
- 5813249
- 3046233
- 5796049

Change View: Total Wait | Average Wait | Typical Day

Email Chart

View Historical Charts for SQL:

Show Full SQL Text

Summary

- Identify
  - What is the Bottleneck
  - End-to-End view of performance
  - Simplify
- Gather
  - Metrics – Current Performance
  - Wait Time
  - Execution Plan
  - Object Definitions and Statistics
- Tune
  - SQL Diagrams
- Monitor
  - New Metrics, Wait Time Profile, Execution Plan

Confio Software - Monitor

CONFIO SOFTWARE
Ignite IT Performance™

- Developer of Wait-Based Performance Tools
- Igniter Suite
  - Ignite for SQL Server, Oracle, DB2, Sybase
- Provides Help With
  - Identify
  - Gather
  - Monitor
- Based in Colorado, worldwide customers
- Free trial at www.confio.com

Myth 1 – Option 1

- Use outer joins vs. NOT IN / NOT EXISTS
- Which class is currently empty?

```
select class_id, name
from class
where class_id not in (
   select class_id from registration)
```

```
-------------------------------------------------------------------
| Id  | Operation          | Name         | Rows  | Bytes | Cost |
-------------------------------------------------------------------
|   0 | SELECT STATEMENT   |              |       |       |    88|
|*  1 |  HASH JOIN ANTI    |              |     1 |    68 |    88|
|   2 |   TABLE ACCESS FULL| CLASS        |  1000 | 64000 |    14|
|   3 |   TABLE ACCESS FULL| REGISTRATION | 80056 |   312K|    72|
-------------------------------------------------------------------
```

Myth 1 – Option 2

CONFIO SOFTWARE
Ignite IT Performance™

- Try NOT EXISTS vs. NOT IN

```
select class_id, name
from class c
where not exists (
   select 1 from registration r
   where c.class_id = r.class_id)
```

```
-------------------------------------------------------------------
| Id  | Operation          | Name         | Rows  | Bytes | Cost |
-------------------------------------------------------------------
|   0 | SELECT STATEMENT   |              |       |       |    88|
|*  1 |  HASH JOIN ANTI    |              |     1 |    68 |    88|
|   2 |   TABLE ACCESS FULL| CLASS        |  1000 | 64000 |    14|
|   3 |   TABLE ACCESS FULL| REGISTRATION | 80056 |   312K|    72|
-------------------------------------------------------------------
```

Myth 1 – Option 3

CONFIO SOFTWARE
Ignite IT Performance™

- Try OUTER JOIN
- No Differences with 3 options

```
select c.class_id, c.name
from class c, registration r
where c.class_id = r.class_id (+)
and r.class_id is null
```

```
-------------------------------------------------------------------
| Id  | Operation          | Name         | Rows  | Bytes | Cost |
-------------------------------------------------------------------
|   0 | SELECT STATEMENT   |              |       |       |    88|
|*  1 |  HASH JOIN ANTI    |              |     1 |    68 |    88|
|   2 |   TABLE ACCESS FULL| CLASS        |  1000 | 64000 |    14|
|   3 |   TABLE ACCESS FULL| REGISTRATION | 80056 |   312K|    72|
-------------------------------------------------------------------
```

Myth 2 – Option 1

- Use MINUS vs. NOT IN
- Which students live in DC area but not in 20002 or 20003 zip
- Cost = 15, LIO = 20

```
select student_id from student
where state in ('VA', 'DC', 'MD')
and zip not in (20002, 20003)
```

```
-------------------------------------------------------------------------
| Id  | Operation                    | Name    | Rows  | Bytes | Cost |
-------------------------------------------------------------------------|
|   0 | SELECT STATEMENT             |         |       |       |    15|
|   1 |  INLIST ITERATOR             |         |       |       |      |
|*  2 |   TABLE ACCESS BY INDEX ROWID| STUDENT |    11 |   110 |    15|
|*  3 |    INDEX RANGE SCAN          | ST_ST   |    11 |       |     3|
-------------------------------------------------------------------------
```

Myth 2 – Option 2

- Try MINUS vs. NOT IN
- Cost = 20, LIO = 23 – Worse Performance

```
select student_id from student
where state in ('VA', 'DC', 'MD')
minus
select student_id from student
where zip in (20002, 20003)
```

```
-------------------------------------------------------------------------------
| Id  | Operation                           | Name    | Rows  | Bytes | Cost |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                    |         |       |       |   20|
|   1 |  MINUS                              |         |       |       |     |
|   2 |   SORT UNIQUE                       |         |    11 |    77 |   16|
|   3 |    INLIST ITERATOR                  |         |       |       |     |
|   4 |     TABLE ACCESS BY INDEX ROWID     | STUDENT |    11 |    77 |   15|
|*  5 |      INDEX RANGE SCAN               | ST_ST   |    11 |       |    3|
|   6 |   SORT UNIQUE                       |         |     2 |    14 |    4|
|   7 |    INLIST ITERATOR                  |         |       |       |     |
|   8 |     TABLE ACCESS BY INDEX ROWID     | STUDENT |     2 |    14 |    3|
|*  9 |      INDEX RANGE SCAN               | ST_ZIP  |     2 |       |    2|
-------------------------------------------------------------------------------
```