

Database Performance Dashboard

CS5226 Database Tuning Project

Presented By

Gu Yang A0040661R

Lu Fangjian A0040740W

Yang Zhaoyu A0040953J

Zhang Haojun A0032586A

2013/04/09

Project Admin

S/N	Task	Member*
1	3 levels of breakdown for each database parameter being monitored.	Gu Yang Lu Fangjian Yang Zhaoyu Zhang Haojun
2	Provide a configuration page(s) to define the various thresholds used to determine whether a specific database parameter is healthy, not so healthy or need DBA attention.	Gu Yang Lu Fangjian
3	For each of the performance issue identified that is in the “red” area, the application should provide information on the parameter in init.ora that should be modified in order to solve the performance issue.	Gu Yang Zhang Haojun
4	On-demand reports – where the users can specify the date ranges for the database parameters being monitored.	Lu Fangjian
5	A Debug interface – allow the user to issue SQL commands to the database. The results should be displayed neatly on the webpage.	Zhang Haojun
6	README.txt	Gu Yang
7	Database Setup Scripts	Gu Yang
8	Configure the web development framework (Spring 3.2.0). Setup the project on google code.	Lu Fangjian
9	UI design and modification.	Lu Fangjian Yang Zhaoyu
10	Configure connection parameters to dbtune server.	Lu Fangjian Yang Zhaoyu
11	Structure design and implement of the background program to query the database and calculate results.	Lu Fangjian Yang Zhaoyu
12	Configure database for testing.	Yang Zhaoyu
13	Project report writing.	Gu Yang Lu Fangjian Yang Zhaoyu Zhang Haojun

*Members are listed in alphabetic order

Part 1. System Design and Architecture

1.1 The Overall Architecture and Design of the Application

The web application uses the pre-installed Oracle database in dbtune server to perform its main activities, which is to monitor a set of database parameters, and to query database through debug window. The database is also acted as permanent storage for the configuration data.

The development of the web interface follows the popular Model-View-Controller (MVC) approach using Spring Framework 3.2.0. The various application layers are separated distinctly in different classes. The detailed relations are illustrated below.

Class	Folder	Role
*TO.java	domain	Transfer objects are data carriers which are used in Presentation layer (web interface), Service layer and DAO layer.
*.jsp	view	Views are web pages which receive transfer objects from controller and display them to user. They also allow user to send inputs to controller for processing.
*Controller.java	controller	Controllers fetch the transfer objects and send them to the view for display. They also receive user inputs and send them to service layer for analysis.
*Service.java	service	Services are implemented with business logics to perform analysis. They may also fetch transfer objects from DAO layer during the process.
*Dao.java	dao	Data access objects retrieve data from permanent storages such as database, put them into transfer objects and send them to service layer for analysis.
*Util.java	util	Constants and static variables/methods which can be utilized to perform certain functions.

Part 2. Database Design, Statistics Computation and SQL Commands

2.1. Database Support for the Dashboard

The dashboard requires some thresholds parameters to be stored in the database. These thresholds parameters determine the status of database, whether it is **HEALTHY**, **MODERATE** or **ATTENTION**. To store the thresholds parameters, we build a single table and populate it with some default values that can be modified later through our web application. The SQL scripts are setup.sql and data.sql (which are in the subfolder /Scripts):

Filename: **setup.sql**

```
DROP TABLE THRESHOLD_PARAM;

CREATE TABLE THRESHOLD_CONFIG
(
    PARAM_NAME          VARCHAR2(100 BYTE),
    PARAM_WARNING        NUMBER(10, 2),
    PARAM_CRITICAL       NUMBER(10, 2)
);
```

Filename: **data.sql**

```
delete from THRESHOLD_CONFIG;
insert into THRESHOLD_CONFIG(PARAM_NAME, PARAM_WARNING, PARAM_CRITICAL)
values ('Shared Pool',40,60);
insert into THRESHOLD_CONFIG(PARAM_NAME, PARAM_WARNING, PARAM_CRITICAL)
values ('Buffer Cache',80,60);
insert into THRESHOLD_CONFIG(PARAM_NAME, PARAM_WARNING, PARAM_CRITICAL)
values ('Redo Log Buffer',30,40);
insert into THRESHOLD_CONFIG(PARAM_NAME, PARAM_WARNING, PARAM_CRITICAL)
values ('Memory Area Used For Sorting',50,20);
insert into THRESHOLD_CONFIG(PARAM_NAME, PARAM_WARNING, PARAM_CRITICAL)
values ('Redo Log Files',1000,9000);
commit;
```

We will monitor following statistics: “Share Pool Free %” for “Shared Pool”, “Buffer Cache Hit Ratio” for “Buffer Cache”, “Redo Allocation Hit Ratio” for “Redo Log Buffer”, “Memory Sorts Ratio” for “Memory Area Used For Sorting”, and “Redo Log Files Total Waits” for “Redo Log Files”.

Each of them will have two thresholds: PARAM_WARNING and PARAM_CRITICAL. If the value monitored becomes greater than PARAM_CRITICAL, **ATTENTION** is alarmed. If the value monitored is between PARAM_WARNING and PARAM_CRITICAL, **MODERATE** is displayed. Otherwise the database is **HEALTHY**.

However, for “Buffer Cache Hit Ratio” and “Memory Sorts Ratio”, the greater the value monitored, the healthier the database is. Hence for those two, threshold PARAM_WARNING is greater than threshold PARAM_CRITICAL.

2.2. The Statistics Computation

The statistics monitored are “Share Pool Free %” for “Shared Pool”, “Buffer Cache Hit Ratio” for “Buffer Cache”, “Redo Allocation Hit Ratio” for “Redo Log Buffer”, “Memory Sorts Ratio” for “Memory Area Used For Sorting”, and “Redo Log Files Total Waits” for “Redo Log Files”.

Most statistics (except the “Redo Log Files Total Waits”) can be directly retrieved (or calculated) from **DBA_HIST_SYSMETRIC_SUMMARY** table. We only need to specify the corresponding metric_id.

The query will need start and end time, to calculate the average value during that time range. The top-level will calculate the average for last 24 hours. The second and low level will split given time range with given interval, to calculate start and end time for each row to be displayed, and retrieve the value by querying the **DBA_HIST_SYSMETRIC_SUMMARY** table.

```
SELECT metric_id, metric_name, avg(average) value
FROM dba_hist_sysmetric_summary
WHERE
    metric_id = metric_id_passed_in
    AND end_time > start AND end_time < end
GROUP BY metric_id, metric_name
```

Note the metric_id_passed_in, start and end are parameters for the function to create above query. The metric_id_passed_in depends on the statistic that is currently investigated by DBA.

For the “Redo Log Files”, the **DBA_HIST_SYSTEM_EVENT** table records down total_waits for different events. We combine it with the **DBA_HIST_SNAP_SHOT** table to get the average total waits for event_name 'log file parallel write':

```
SELECT avg(a.total_waits)
FROM dba_hist_system_event a, dba_hist_snapshot b
WHERE
    b.snap_id = a.snap_id
    AND a.event_name = 'log file parallel write'
    AND b.end_interval_time > start
    AND b.end_interval_time < end
```

Similarly, start and end are the parameters for the function to create above query.

2.3 The Oracle Advisory

Besides the statistics monitored, we also noticed that Oracle now provide advisory for “Shared Pool” and “Buffer Cache”, which gives estimated performance change if the corresponding parameter in init.ora is adjusted.


We think it would be helpful for DBA to have that information when using our dashboard:

```
SELECT avg(a.total_waits) shared_pool_size_for_estimate,  
shared_pool_size_factor, estd_lc_time_saved  
FROM v$shared_pool_advice
```

```
SELECT size_for_estimate, buffers_for_estimate,  
estd_physical_read_factor, estd_physical_reads  
FROM v$db_cache_advice
```

Part 3. Screenshots

3.1. Login



Please sign in

username:
dba

password:
...

Login in as DBA before starting using the web application.

3.2. Overall Database

Overall Database	Shared Pool	Buffer Cache	Redo Log Buffer	Redo Log Files	Memory Area	Database Report	Threshold Setting	Debug Mode	cs5226
Overall Database									
<i>Overview</i>									
Start Time	08/04/2013 21:19:46								
End Time	09/04/2013 21:19:46								
Shared Pool	MODERATE								
Buffer Cache	HEALTHY								
Redo Log Buffer	ATTENTION								
Redo Log Files	ATTENTION								
Memory Area	HEALTHY								

Home page contains an overall view of five monitored parameters of the database during last 24 hours. The status of ATTENTION is linking to an advisor page, will show in details in 3.3.

3.3. Overview

Overall Database

Shared Pool

Buffer Cache

Redo Log Buffer

Redo Log Files

Memory Area

Database Report

Threshold Setting

Debug Mode

cs5226

Redo Log Buffer

Overview

Start Time

08/04/2013 21:55:33

End Time

09/04/2013 21:55:33

Redo Allocation Hit Ratio

42.21%

Current Status

ATTENTION

Second Level Details

Start Time (DD/MM/YYYY HH24:MI:SS)

08/04/2013 21:55:33

End Time (DD/MM/YYYY HH24:MI:SS)

09/04/2013 21:55:33

Interval X (Min)

60

Submit

Overview of a parameter status in last 24 hours includes **HEALTHY**, **MODERATE** and **ATTENTION**. Provide DBA the option to choose any range and interval for second level details.

Here we also provide a special feature such that provides advisor on ATTENTION status, when click on the ATTENTION link, it will show the advisor for DBA. There are two types of advisor we provide, one is with quantified potential performance gain and the other is text suggestion. In addition, both types of advisor will provide link to official oracle document.

Overall Database

Shared Pool

Buffer Cache

Redo Log Buffer

Redo Log Files

Memory Area

Database Report

Threshold Setting

Debug Mode

cs5226

Shared Pool

The metric monitored for Shared Pool is "Shared Pool Free %".

In init.ora, the parameter **SHARED_POOL_SIZE** needs to be increased, when you see ATTENTION alert.

See official Oracle documentation for [shared pool memory management](#).

The table below is the result from Oracle Shared Pool Advisor:

SHARED_POOL_SIZE_FOR_ESTIMATE	SHARED_POOL_SIZE_FACTOR	ESTD_LC_TIME_SAVED
128	.8889	8247
144	1	8323
160	1.1111	8358
176	1.2222	8389
192	1.3333	8414
208	1.4444	8462
224	1.5556	8496
240	1.6667	8504
256	1.7778	8522
272	1.8889	8528
288	2	8532

Overall Database

Shared Pool

Buffer Cache

Redo Log Buffer

Redo Log Files

Memory Area

Database Report

Threshold Setting

Debug Mode

cs5226

Redo Log Buffer

The metric monitored for Redo Log Buffer is **Redo Allocation Hit Ratio**.

In init.ora, the parameter **LOG_BUFFER** needs to be increased, when you see ATTENTION alert.

See official Oracle documentation for [reasonable estimation of log buffer size](#).

3.4. Second level details

[Overall Database](#) [Shared Pool](#) [Buffer Cache](#) **[Redo Log Buffer](#)** [Redo Log Files](#) [Memory Area](#) [Database Report](#) [Threshold Setting](#) [Debug Mode](#) cs5226

Redo Log Buffer

Second Level Details

Select	S/N	Start Time	End Time	Redo Allocation Hit Ratio	Status
<input type="radio"/>	1	09/04/2013 19:55:00	09/04/2013 20:55:00	44.99%	ATTENTION
<input checked="" type="radio"/>	2	09/04/2013 20:55:00	09/04/2013 21:55:00	33.25%	MODERATE

Low Level Details

(Please select one of the records above)

Interval Y (Min)

The second level details with range two hours and interval 60 minutes. Slightly different from overview page, we provide DBA the option to choose certain date range and any interval. In this page, the status of ATTENTION is also linking to the advisor page.

3.5. Low level details

[Overall Database](#) [Shared Pool](#) [Buffer Cache](#) **[Redo Log Buffer](#)** [Redo Log Files](#) [Memory Area](#) [Database Report](#) [Threshold Setting](#) [Debug Mode](#) cs5226

Redo Log Buffer

Low Level Details

S/N	Start Time	End Time	Redo Allocation Hit Ratio	Status
1	09/04/2013 20:55:00	09/04/2013 21:10:00	27.27%	HEALTHY
2	09/04/2013 21:10:00	09/04/2013 21:25:00	25.00%	HEALTHY
3	09/04/2013 21:25:00	09/04/2013 21:40:00	26.10%	HEALTHY
4	09/04/2013 21:40:00	09/04/2013 21:55:00	70.00%	ATTENTION

The low level details with interval 15 minutes. Similar to second level details, ATTENTION status is linked to an advisor page.

3.6. Database Report

Overall Database

Shared Pool

Buffer Cache

Redo Log Buffer

Redo Log Files

Memory Area

Database Report

Threshold Setting

Debug Mode

cs5226

Database Report

Start Time (DD/MM/YYYY HH24:MI:SS)

08/04/2013 22:05:13

End Time (DD/MM/YYYY HH24:MI:SS)

09/04/2013 22:05:13

Interval (Min)

60

Generate Report

Choose the criteria of database report and then click generate report.

Overall Database

Shared Pool

Buffer Cache

Redo Log Buffer

Redo Log Files

Memory Area

Database Report

Threshold Setting

Debug Mode

cs5226

Database Report

S/N	Start Time	End Time	Shared Pool Free %	Buffer Cache Hit Ratio	Redo Allocation Hit Ratio	Redo Log Files Total Waits	Memory Sorts Ratio
1	08/04/2013 22:06:00	09/04/2013 02:06:00	57.78%	99.98%	56.56%	161,289.583	100.00%
2	09/04/2013 02:06:00	09/04/2013 06:06:00	57.74%	100.00%	37.81%	163,130.583	100.00%
3	09/04/2013 06:06:00	09/04/2013 10:06:00	58.54%	100.00%	38.70%	164,491.458	100.00%
4	09/04/2013 10:06:00	09/04/2013 14:06:00	60.81%	100.00%	40.72%	165,786.525	100.00%
5	09/04/2013 14:06:00	09/04/2013 18:06:00	59.32%	100.00%	39.58%	167,141.375	100.00%
6	09/04/2013 18:06:00	09/04/2013 22:06:00	59.25%	100.00%	40.75%	168,480.083	100.00%

Database Report will be generated with all parameters' status in choosing range and interval.

3.7. Debug Mode

Overall DatabaseShared PoolBuffer CacheRedo Log BufferRedo Log FilesMemory AreaDatabase ReportThreshold SettingDebug Modecs5226

Debug Web Terminal:

select * from threshold

Execute

Error Information:

ORA-00942: table or view does not exist

Results:

Overall DatabaseShared PoolBuffer CacheRedo Log BufferRedo Log FilesMemory AreaDatabase ReportThreshold SettingDebug Modecs5226

Debug Web Terminal:

select * from threshold_config

Execute

Error Information:

Results:

PARAM_NAME	PARAM_WARNING	PARAM_CRITICAL
Shared Pool	40	60
Buffer Cache	80	60
Redo Log Buffer	30	40
Memory Area Used For Sorting	50	20
Redo Log Files	1000	9000

Debug mode executes the input query in database, and subsequently a result table will be populated in *Results* area or an error message will be shown at *Error Information* area.

3.8. Threshold Setting

Overall DatabaseShared PoolBuffer CacheRedo Log BufferRedo Log FilesMemory AreaDatabase ReportThreshold SettingDebug Modecs5226

Threshold Setting

Database Parameter	Metric Name	Warning	Alert
Shared Pool	Shared Pool Free %	<input type="text" value="40.0"/>	<input type="text" value="60.0"/>
Buffer Cache	Buffer Cache Hit Ratio	<input type="text" value="80.0"/>	<input type="text" value="60.0"/>
Redo Log Buffer	Redo Allocation Hit Ratio	<input type="text" value="30.0"/>	<input type="text" value="40.0"/>
Redo Log Files	Redo Log Files Total Waits	<input type="text" value="1000.0"/>	<input type="text" value="9000.0"/>
Memory Area For Sort	Memory Sorts Ratio	<input type="text" value="50.0"/>	<input type="text" value="20.0"/>

Update

DBA is free to set the *Threshold Setting* through this page. After setting, just click *Update* and *Threshold Setting* will be updated.

Part 4. Discussion and Summary

We have developed a web application of database performance dashboard, which is able to monitor the health of the database and to support database performance investigation.

In this project, we get the idea of how to connect to the database as system user and get the table or view from the database (some table cannot be seen by normal user). We also learned how to get the information from the database which consists of snapshots that were done periodically by the Oracle database and use it to monitor the performance of the database. In order to have a real time value, we studied how to modify the period of the snap shot and we set the period of the snap shot to 10 minutes. We also get some idea about how to retrieve information faster from data base, such as disconnect the database only after we finish all the queries rather than start up a standalone connection for every single query. Another thing we do to make the query faster is to use group by and average in the query instead of getting all single records and calculating in the java side.

In our database performance dashboard application, all the information is extracted from three history tables, `dba_hist_sysmetric_summary`, `dba_hist_system_event`, and `dba_hist_snapshot`, which are updated regularly by Oracle. We store and present the average value of last 24 hours data from the current time in the dashboard. We also provide hourly breakdown and third level breakdown in every hour using the interval input by the user. If the user needs data from other intervals, they can choose to use our on demand report. The on demand report allows user to input the starting time, ending time as well as the breakdown interval. In the report, our application uses different colour for different status which can make the user easier to get the idea of the database. If our dashboard application notice the parameter is in attention range, the system will colour the status using red colour and append a hyperlink which carry the solution to this status value. In the solution page, our system suggests what the database administrator can do in order to solve the problem. The range of thresholds can be modified through our system's thresholds' configuration page. DBA can input two values for every field: warning and attention. A debug page is also provided for the user which allows user to test the queries and their performance. The result of the query will be display on the same page.

Our system has some limitations, the overall information page cannot refresh by itself, users need to refresh the page to get the latest information of the database; we only monitor 5 parameters from the database. These can be improved in the future for the dashboard application.