ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 10, April 2013

Oracle10g Workload Performance Measurement and Tuning Toolkit

Pinki singh, Rakesh Kumar singh, Jayant shekhar

Abstract— Database tuning practice is mainly conducted as a consequence of user's complaints on the per-formance There is a need for a reactive monitoring and tuning tool enabling a real-time overview of the main resource consumers in order to detect and solve performance bottlenecks. With an assessment of the Oracle's high-availability database, in terms of the main architectural components and their impact on performance, we have developed a Java tool for the efficient and re-source-effective tuning of Oracle databases. Our tool, called Automatic Database Diagnostic Monitor (ADDM and Tuning (WPMT), In order to tune the DBMS within their complex environments, highly skilled database administrators (DBAs) are required. Unfortunately, they are becoming rarer and more and more expensive illustrating the efficiency of memory initialization parameters versus component's workload performance.

Keywords: **Database Management Systems Automatic Tuning, Workload Performance.**

I. INTRODUCTION

Tuning and administration of database systems within their heterogeneous runtime environments is a complex and non-trivial task. Achieved DBMS performance highly depends on individual skills of expensive DBAs who need to have deep knowledge about the DBMS itself. Regardless of IT environment-specific requirements for running mission-critical database-driven applications, the two basic requirements are always requested: firstly, high availability, in terms of database resource availability; and secondly, a high performance, in terms of access and code efficiency (Silberschatz, Korth & Sudarshan, 2002). Autonomic Computing (AC), a well-known initiative of IBM® started in 2001, aims to reduce Total Cost of Ownership of IT systems by making software systems self-regulating. Routine administrative and tuning tasks are shifted towards the IT systems themselves, thus enabling the administrators to concentrate on long-term strategic decision making. These database management systems, i.e. the most demanding systems with 24x7 availability (24 hours a day, 7 days a week), maintain the optimal database workload performance. Although Oracle, IBM, and Microsoft provide a solution for a high availability (or multi -node) computing environment, availability, scalability, and manage-ability among them differ significantly (Buch & Cheevers, 2002; Chandrasekaran, & Kehoe, 2003; MSDN, 2004). Ultimately responsible for database tuning, the most inconvenient situation for a database administrator (DBA) is when users submit complaints about the data-base performance. Forced to act immediately (reactive tuning), for the DBA it is a critical task due to the fact that it often happens during the peak database activity. A common solution would be to optimize the SQL (Structured Query Language) statement's Execution Plan that improves the execution performance. However, performing the object optimization during the peak database activity shall decrease the overall performance, since the optimization process consumes resources (CPU and RAM), and may even bring some objects temporary unavailable. Identifying the cause of performance overhead is not an easy task. For example, knowing that excessive disk I/O operations may bring down the data-base performance up to 2500 times (Burleson, 2003), a DBA must have a real -time overview of major I/O users, as well as the most expensive user's SQL statements causing the slowdown. The root cause can be inaccurate SQL's Execution Plan, or unbalanced memory allocation due to in-appropriate initialization of database, or because the database server has no more resources to handle the workload and requires the upgrade. In any case, the DBA must be able to assess the current database resource consumption and to proceed with investigation to discover the root of the problem and to solve the problem. Based on conducted on-line survey and an interview with Oracle Support Manager (Zildžić, 2005), we may conclude that database memory and index tuning practice needs an improvement. Since it is mainly conducted as a consequence of database performance overhead, i.e. when the user submits a complaint, a proactive monitoring of database performance should be implemented to anticipate performance leakage and ultimately avoid severe performance degradation. The above considerations led to a custom developed monitoring and tuning tool which delivers a unique solution for the efficient and resource-effective Oracle database workload performance measurement and tuning. This tool, called Workload Performance Monitoring and Tuning (WPMT), is developed in Java. It can proactively and reactively be used during Oracle database tuning in order to measure the current workload performance of the main database components, anticipate resource shortage, and optimize database objects for the achievement of the optimal database performance (Zildžić, 2005).

II. ORACLE'S MONITORING AND TUNING TOOLS

Effective data collection and analysis is essential for identifying and correcting system perform-ance problems. Oracle provides tools, such as Statspack and Oracle Enterprise Manager (OEM) packs allowing a DBA to gather information on database performance (Metalink A87503-02, n.d.). These tuning tools, recommended by Oracle

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 10, April 2013

Corporation and currently the most often used, use data dictionary views to capture database activity metrics. In this section, a part from Oracle Company designed tools, we shall assess the third-party tuning tool, i.e. the MGA's EagleEye, developed by a recognized Oracle database expert Mark Gurry.

III. STATS PACK UTILITY

Stats pack utility consists of a set of Programming Language/Structured Query Language (PL/SQL) scripts, executed against the database, in order to gather, store data and metrics, and generates reports of the database activity. These metrics (Table 1), collected from dynamic per -formance tables, are automatically recorded in utility -created tables within a database with the DBA's choice of collection and threshold level to be used (Metalink 228913.1, n.d.). Bundled with the Oracle Server, it is also known as Utlbstat/Utlestat and as of database release version 8.1.6, it is renamed as Stats pack.

Table 1: Stats Pack Utility Collection Levels

Collection level*	Description
0 to 4	General performance metrics on all memory areas, latches, pools, and events.
5 to 9	Same metrics from the lower levels, plus the most resource-intensive SQL statements.
10 +	Same metrics from the lower levels, plus parent/child latch data.

*By default and recommended value of the collection level is 5, but it depends on the scope of the tuning.

Stats pack tool has been designed by the Oracle Server architects, and it generates a set of reports based on user -defined collection level. With a higher collection level, metrics capturing consist-ing of almost all database performance activity related segments. Executing this utility has high costs in terms of space usage and query performance. For example, just for STATS$ SYSSTAT table, Oracle will generate 255 rows for each snapshot, and there are 35 tables more. Whether required or not, the thousands of rows will be generated anyway with this utility. Managing historical metrics (dozen of snapshots) requires significant space allocation. The reports are generated in text format only, and it is difficult to establish "chain" conclusion-analysis with dependent parameters. Report analysis is a time consuming task due to non self-intuitive and complex report presentation. Stats pack is not supported with database releases earlier than 8.1.6, and storing data from multiple databases in one PERFSTAT user account is currently not supported.

IV. ENTERPRISE MANAGER (EM) TOOLS

Enterprise Manager (EM) is Oracle's single, integrated solution for managing, administering and monitoring global enterprises.EM management framework consisting of three tiers: The Console (client tier) and its integrated tools provide a graphical interface for administrators to manage the complete Oracle environment. Management Servers and a database repository provide a scalable middle tier for processing system management tasks. Intelligent Agents are installed on each network node (database) to monitor the services that node provides, and to exe-cute tasks from the Management Server (Metal ink A96674-01, n.d.). Complex events are used to trigger tuning plans. A tuning plan is an encapsulated tuning workflow, consisting of a number of tuning actions and control structures, enriched with definitions of input and output parameters, as well as further metadata. Oracle offers several tools to administer the complete Oracle environment, including: databases, Internet Application Servers (iAS), applications, and services. For this research we have assessed two types of tools that help the DBA to tune and monitor Oracle database: Performance Tuning pack (Metalink A86647-01. n.d.) with Oracle Expert tool for automated database tuning, and Di-agnostics pack (Metalink A88748-02, n.d.) with a tool called Performance Manager that provides a set of graphical charts to monitor the performance of the database. Oracle Expert provides a methodology that is used to collect, evaluate, verify, and implement da-tabase tuning changes (Metalink 93490.1, n.d.). Tuning areas cover instance parameter tuning, database structure and placement, index tuning, and SQL statement reuse evaluation. During the tuning , Oracle Expert collects metrics from dynamic performance views, processes this input through the various rules and creates recommendations, tuning scripts, and reports. Before creating a tuning session, Oracle establishes a current workload baseline (application SQL data and statistics or SQL History) of the database environment. Tuning session consists of five steps: defining the scope(SGA tuning, PGA tuning,SQL reuse, storage management), metrics collection, metrics review, recommendations based on built-in rules, and generating the SQL scripts based on the recommendations.

V. OBSERVATION

Oracle Expert tool, with 900 tuning rules built-in, sub-applications SQL Analyze and Index Wizard, represents Oracle's most enhanced graphical tool, especially with the release 9i. To host and manage collected metrics, this tool creates near 700 objects: 249 tables, 270 indexes, and 63 views. **User interaction** steps allow to involve the DBA in the tuning tasks that cannot be performed automatically (e.g. because of semantic-related changes). These steps allow the tuning plan de-finer.

- to send notifications containing some requests or suggestions, such as a request to restart the DB2 instance in order to let some offline configurable parameters take effect;
- to ask the DBA for a permission to proceed with some semantic-changing or high-impact configuration changes;
- to provide the DBA with some information collected

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 10, April 2013

during the tuning plan execution and describing the problem situation in case the problem cannot be resolved automatically.

Algorithms for processing recommendations based on the collected metrics are high memory and CPU consumers. Using this tool with a repository installed on a production (main) database would not be an advantage due to its negative impact on the overall database performance. This tool provides a complete overview of Oracle database state, in the form of multiple- choice charts in HTML format. Also it can monitor Microsoft SQL server database. It is available for LINUX, and Windows platforms only. For the database release prior to 9i,10G tool is available only with OMS repository connection. With 10g release, without OMS only the current activity can be monitored. Collection and review of historical data is available with OMS repository connection exclusively. With a frequent refresh rate (snapshot time interval), the database has drawbacks in the overall performance due to increased activities (CPU and RAM consumption) necessary to calculate all performance metrics.

VI. MGA's EagleEye

MGA EagleEye is a third- party software tool developed by Mark Gurry & Associates (MGA) that monitors the performance of Oracle databases and provides real-time diagnostic information to allow for immediate rectification of performance problems (http://maxdbt.com/eagleeye.htm) (Gurry, 2001; Gurry & Corrigan, 1996). This tuning tool is reviewed by 'SELECT' magazine, Australian Oracle Users Group, and recommended for users experiencing extensive problems with performance and robustness on their mission critical database.

Observation

The tool runs a set of SQL statements and records tuning information when ex-pected response times are exceeded, including details of logged on users, the applications and specific actions they are running, and where the contention is occurring. It also produces graphi-cal information on:

- Shared Pool Problems
- Analysis (and not rebuilt) of tables and indexes
- Waits in tables and indexes
- Sort efficiency
- Disk I/O analysis

Eagle Eye is a product that has been developed to help DBAs to quickly identify where the per-formance problems are occurring and the cause of the problems. When used with the Mark Gurry's books Oracle Performance Tuning and Oracle SQL Tuning, the tool enables the DBA to get performance problems under control. This tool does not provide a solution for recording his-torical metrics. It is designed to give an instant overview of database performance, and therefore is not suitable for long-term proactive database maintenance. Also, it does not enable object optimization nor cache hit-ratio analysis.

VII. WORKLOAD PERFORMANCE MEASUREMENT AND TUNING

Workload Performance Measurement and Tuning (WPMT) is a custom developed monitoring and tuning tool enabling a DBA to perform reactive and proactive Oracle database tuning, with an efficient and resource-effective solution for managing the memory related historical resource and performance metrics. This tuning plan () determines the current size of the problematic buffer pool and increases it by 3000 pages With an assessment of the high-availability Oracle database architectural components and their impact on workload performance, we have designed a tool with following features:

- Database connections assessment. A feature required for the successful implementation of developed tuning strategy for administrating efficiently mission-critical Oracle data-base-driven applications.
- Database workload assessment. Capturing and displaying in real-time the database users with the most demanding resource allocations, in terms of top memory and CPU consumers, as well as the most expensive SQL statements in terms of I/O executions.
- Instance memory allocation assessment. In order to determine the efficiency of allocated memory to manage the workload, the current memory allocation breakdown, calculated cache hit ratios using a ratio-methodology, memory sort ratio, and a redo log space re-quests, are calculated and displayed in real-time as well.
- Bottleneck or Wait Event assessment. With wait-methodology, we have determined total instance session's wait time and corresponding event description, required to identify the root causes of session's hang-up.
- Storage or space allocation assessment. In order to anticipate space shortage and eliminate unnecessary database performance overhead or downtime, we have developed a feature to provide an overview of the currently free/used space. Furthermore, with calculated fragmentation index, the efficiency of the overall data file free space has been determined and displayed.
- Disk or I/O activities assessment. A real-time assessment feature detecting inappropriate or unbalanced throughput of physical storage allocation has been developed in terms of highest physical reads/writes per data file.
- RAC inter-node performance assessment. An overview of top wait events related to in-stance cache has been determined using RAC's dictionary views. The efficiency of inter-instance

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 10, April 2013

intercommunication component has been assessed using average current block time metric.

- Memory metrics management. Instead and as a consequence additionally increase the database workload; we have developed a solution that uses a flat file for recording and reviewing current/historical metrics. Recorded memory related performance metrics keep track of database performance achieved with historical initialization parameters, and enable a DBA to continuously measure the efficiency of allocated memory compared to the component's workload performance (delta metrics).
- Automated object optimization feature updates the database data dictionary with the cur-rent table and index statistics required by the Oracle's Cost Based Optimizer to calculate the best SQL Execution Plan. With accurate table and index statistics, a database work-load performance will be improved.
- Tuning a database system heavily depends on the current workload type. A DBA has different – sometimes disjoint – tuning goals in mind when dealing with several workload types. For example, a database system configuration suitable for one workload type might be totally insufficient for another workload type. On the other hand, trying to use one database system configuration for all workload types usually results in suboptimal per-formance due to insufficient resource utilization.
- These features successfully integrated in a user-friendly Java application (WPMT tool), enable a DBA to continuously measure, resource-effectively and efficiently, the workload performance excellence of high-availability Oracle databases.

VIII. WPMT TOOL'S TESTING

These features successfully integrated in a user-friendly Java application (WPMT tool), enable a DBA to continuously measure, resource-effectively and efficiently, the workload performance excellence of high-availability Oracle databases. Our tool has been tested with three workload setting criteria: Low, Medium, and High, as de-scribed in Table 2. Transactions have been executed by each user independently and simultaneously against a single CPU P-IV 2.8GHz/1GB of RAM, on Personal edition Oracle 10g data-base. These transactions, containing a mixture of SQL commands such as INSERT, UPDATE, SELECT, and DELETE, are executed via sql scripts from Oracle SQL Plus editor.

Table 2: Workload criteria

	Low	Medium	High
Users	2	5	13
Transactions	12000	19500	43000
Exec. Time	1h00	1h15	2h00

A Low workload criteria is defined as two (2) users executing 12 000 transactions in one hour. A Medium workload criteria is defined as five (5) users, executing 19 500 transactions within an hour and fifteen minutes. A High workload criteria is defined as thirteen (13) users, executing 43 000 transactions within two (2) hours. Hit ratios at the beginning (start) and at the end (end) of the simulation, as well as adjusted ini-tialization parameters SHARED_POOL_AREA (SPA) and DB_CACHE_ SIZE (CACHE), are illustrated in Tables 3, 4, 5 for Low, Medium, and High workload settings respectively.

Table 3: Low Workload*

	start	end
Dic. Hit	83,7	95,8
Lib. Hit	99,3	100,0
Buff. Hit	76,9	99,5
SPA	8MB	12MB
CACHE	4MB	4MB

*2 users/12 000 trans./1h

Table 4: Medium Workload**

	start	end
Dic. Hit	83,6	95,8
Lib. Hit	99,3	99,9
Buff. Hit	78,1	99,4
SPA	8MB	16MB
CACHE	4MB	8MB

**5 users/19 500 trans./1h10m

Table 5: High Workload***

	start	end
Dic. Hit	83,9	95,7
Lib. Hit	99,3	99,6
Buff. Hit	76,7	99
SPA	8MB	20MB
CACHE	4MB	8MB

***13 users/43 000 trans./2h

Table 5: High Workload***

	start	end
Dic. Hit	83,9	95,7
Lib. Hit	99,3	99,6
Buff. Hit	76,7	99
SPA	8MB	20MB
CACHE	4MB	8MB

***13 users/43 000 trans./2h

Low Workload results, as illustrated in Table 3, show that for the achievement of an optimal da- tabase performance in case of dictionary hit (95.8%) and a peak performance in case of library hit (100%), a size of shared pool area initialization parameter (SPA) had to be increased from 8 to 12 MB.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 10, April 2013

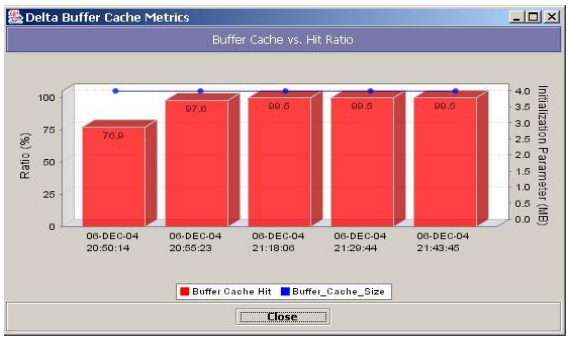

Fig 1: Buffer Hit Ratio – Low workload

Medium Workload results, as illustrated in Table 4, show that bigger values for both initializa-tion parameters: shared pool area size (SPA) and buffer cache (CACHE), were required to achieve optimal/peak database workload performance. In case of buffer cache ratio a CACHE size was increased from 4 to 8 MB achieving 99.4 %; dictionary hit ratio achieving 95.8% with 16 MB, and a peak performance for library hit ratio (99.9%) with 16 MB. The results of a 100 minute tuning run are pro-vided in Figure 3. During the first 50 minutes OLAP workload has been generated on the system. It was followed by a 40 minute OLTP and an ap-proximate 10 minute OLAP period. After the first 10 minutes of the test run, ATE tuning was started, as indicated by a solid line.

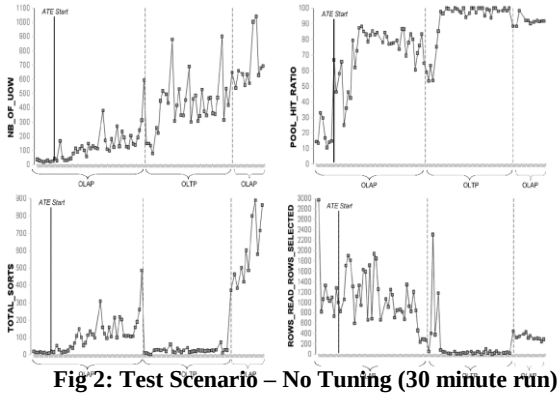

Fig 2: Test Scenario – No Tuning (30 minute run)

After only a few executions of the tuning plan increasing the buffer pool size, the BPHR reaches 80%. During further OLAP execution it slightly goes down. However, once it falls below predefined threshold of 70%, the corresponding tuning plan gets executed again. After the workload shift to OLTP, the threshold gets adapted, which results again in buffer pool size changes further increasing the BPHR.describe requirements, in terms of parameters size, for an achievement of optimal database workload performance. A 20 MB of memory allocated to shared pool area (SPA) were needed to achieve 95.7% and 99.6% hit ratios for dictionary and library respectively. A buffer cache size (CACHE) was increased from 4 to 8 MB in order to achieve 99% buffer hit ratio. As illustrated in Tables 3,4, and 5, and as demonstrated in Figure 1 and 2, we may conclude that our WPMT tool, can be used efficiently and resource-effectively to tune database initialization parameters in order to achieve optimal/peak workload performance regardless the current data-base workload.

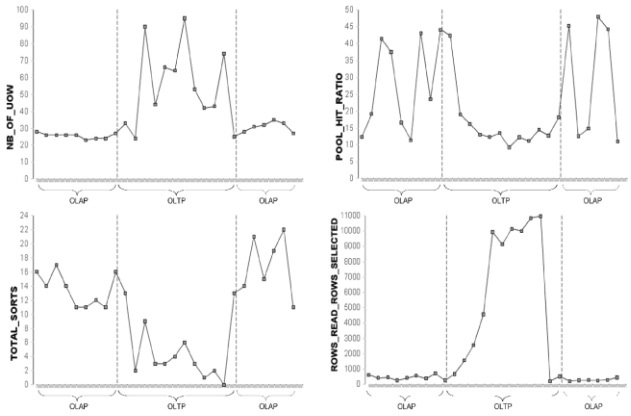

Fig 3: Test Scenario - ATE Tuning (100 minute run)

IX. WPMT TOOL'S EVALUATION

Comparison between Oracle tuning tools and custom developed WPMT tool has been described in Table 6.

(2.1)Metrics are hidden and used by internal algorithms to provide recommendations in the form of tuning scripts

(4.1)Enhanced with the Explain plan feature (trace the execution of the SQL in order to discover all objects involved in the execution)

(4.2)Limited to Shared Pool Area, and no cache-hit ratio analysis

(5.1)Cache hit-ratio metrics and top initialization parameters

Oracle Statspack utility is an effective tool measuring the performance of each database com-ponent. To have an overview of metrics generated by this tool, the database must pay high costs in terms of resources, since to host the collected metrics, the utility will create additional schema in the production database. This utility maintains only the cumulative metrics, since all collected and recorded metrics are automatically deleted when the instance is restarted. The main disadvantage of this Oracle's recommended tool for proactive maintenance, in our opinion, is an additional performance overhead created with an each utility's snapshot that collects metrics from almost all database performance views (default and recommended is level 5). For example, if we want to apply Oracle's recommended tuning methodology, i.e. one-by-one parameter, and to evaluate the effect of change on one component, Stats pack's snapshot will capture metrics for other components anyway, thus increasing unnecessary database workload. Further more, the generated text file (report) is very complex to analyze and understand. For example, to evaluate the efficiency of hit ratios compared to dependent initialization parameters, the DBA must browse txt report from the beginning until the end since the information is provided on separate locations within a same report file.

Oracle Enterprise Manager (OEM) tools, i.e. tuning and diagnostic packs are effective tools but should be used jointly with a repository on a remote database (separately from production database) due to high cost in resources. Using these tools independently would not be an advantage since

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 10, April 2013

tuning pack provides only recommendations and not an overview of performance metrics. Algorithms (rules) are hidden, and the DBA has no insight of how and why did Oracle server draw the conclusions and which parameter should be changed. A tuning pack in combination with diagnostic pack provides a complete monitoring and tuning mechanisms to measure overall and independently database component's workload. For example, when Oracle Expert (tuning pack) recommends altering SHARED_POOL_SIZE parameters with optimal value, it will generate the script, and the DBA, after executing the script, may monitor the library and dictionary hit ratio through performance manager (diagnostic pack). Maintaining the OEM repository and tools high financial costs (new database on new server) are the main disadvantages in our opinion.

MGA's EagleEye is an efficient tool to optimize SQL statements for the best Execution Plan. The Author's book (Gurry, 2001) and EagleEye tool focus on database component that mainly measures SQL execution efficiency, i.e. Shared Pool Area. Although it is enhanced with storage and I/O metrics, the tool is quite limited with features for measuring overall database workload performance. User-friendly GUI, low cost in resources, and a quick overview of the performance is an advantage of this tool. In our opinion, it is suitable only for proactive database maintenance. With **WPMT tool**, we have combined the best practices to monitor and measure the overall data-base workload performance.

X. SUMMARY AND CONCLUSION

A custom developed Java application (WPMT v1.0) combines the best practices of the most common Oracle tuning tools. WPMT's user-friendly GUI enables a DBA to efficiently perform proactive and reactive database tuning regardless of the current database workload. Our tool combines two approaches for gauging database workload performance hit-ratio and wait (or bottleneck) analysis. Measuring a workload performance with wait analysis enables the DBA to have an insight where a database is spending time, and with hit-ratio, the efficiency of instance memory allocation is instantly detected. Overcoming the limitations of measuring only the current (cumulative) metrics is achieved with a cost-effective and efficient solution, i.e. the flat-file metrics management. This architecture enables administrators to specify their tuning procedures (tuning plans) that can be executed automatically when the corresponding problem has been recognized. For definition of tuning plans, a formal model is needed that pro-vides a mapping between high-level user-defined tuning plans and underlying atomic tuning actions in oracle 10g(ADDM).With this solution the burden of performance overhead created with a historical metrics is completely removed from a production database. Measurement with delta metrics and monitoring of memory allocation for each SGA component enables the DBA to efficiently readjust the memory for optimal database workload performance. Effective combination of the memory allocation with initialization parameter and overall efficiency of the related component responsible to manage the current work-load is an advantage compared to all Oracle tools. In case of reported performance overhead (for example, low hit-ratios), based on available SGA memory, a DBA can reallocate additional memory to area(s) with low performance (SPA or Buffer Cache).

REFERENCES

[1] W.J. Chen, U. Baumbach, M. Miskimen, et al. DB2 Performance Expert for Multiplatforms V2.2, March 2006.

[2] S. Elnaffar. Towards Workload-Aware DBMSs: Identifying Workload Type and Predicting its Change. PhD Thesis. Queen's University, Canada, 2004.

[3] S. Elnaffar, W. Powley, D. Benoit, et al. To-day's DBMSs: How Autonomic Are They? First International Workshop on Autonomic Computing Systems, Prague, 2003.

[4] Enterprise Management Associates: Practical Autonomic Computing - Roadmap to Self Man-aging Technology. White Paper. 2006.

[5] P. Horn. Autonomic Computing: IBM's Perspective on the State of Information Technology. White Paper. IBM Research. 2001.

[6] IBM Corp. Autonomic Computing Toolkit: Developer's Guide. Technical Report SC30-4083-02, IBM, August 2004.

[7] IBM Corp. DB2 UDB ESE V8 non-DPF Performance Guide for High Performance OLTP and BI. IBM Redbooks. 2004.

[8] IBM Corp. An Architectural Blueprint for Autonomic Computing. White Paper. IBM Research Lab, 2004.

[9] IBM Corp. DB2 Universal Database - Administration Guide: Performance (Version 8). 2002.

[10] IBM Corp. DB2 Universal Database - System Monitor Guide and Reference (Version 8). 2004.

[11] G. Lanfranchi, P. Peruta, A. Perrone, et al. To-ward a New Landscape of Systems Manage-ment in an Autonomic Computing Environment. IBM Systems Journal 42, No. 1 (2003).

[12] M. Peleg, A. Boxwala, O. Ogunyemi, et al. GLIF3: The Evolution of a Guideline Representation Format. In Proceedings of the AMIA Fall Symposium, 2000.

[13] M. Peleg, S. Tu, J. Bury, et al. Comparing Computer-interpretable Guideline Models: a Case Study Approach. Journal of the American Medical Informatics Association 10(1) (2003) 52–68

[14] W. Powley, P. Martin. A Reflective Database-Oriented Framework for Autonomic Managers. In Proceedings of International Conference on Autonomic Systems, San Jose, CA, USA, 2006.

[15] G. Rabinovitch, D. Wiese. Non-linear Optimization of Performance Functions for Autonomic Database Performance Tuning. In Proceedings of The Third International Conference on Autonomic and Autonomous Systems, Athens, 2007.

[16] M. Sedlmayr, T. Rose, R. Röhring, et al. A workflow approach towards GLIF execution. In Proceedings on Workshop AI Techniques in Healthcare: Evidence-based Guidelines and.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 10, April 2013

[17] Metalink 228913.1. (n.d.). System wide tuning using Stats Pack reports. Oracle Support Services. On-line library:

[18] Metalink A96674-01. (n.d.). Oracle enterprise manager concepts guide release 9.2.0. On-line: http://download-west.oracle.com/docs/html/A96674_01/toc.htm

[19] Metalink A86647-01. (n.d.). Oracle enterprise manager database tuning with the Oracle tuning pack re-lease 9.0.1. On-line library: http://download-west.oracle.com/docs/cd/A91202_01/901_doc/em.901/a86647/tun_ovw.htm#1001025

[20] Metalink A88748-02. (n.d.). Oracle enterprise manager getting started with the Oracle diagnostics pack release 9.0.1. On-line: http://download-west.oracle.com/docs/cd/A91202_01/901_doc/em.901/a88748/toc.htm

[21] Metalink 93490.1. (n.d.). How to use Oracle Expert. Oracle Support Services. On-line library: www.metalink.oracle.com

[22] Metalink 169551.1. (n.d.). FAQ Oracle diagnostics pack 9.0.1. Oracle Support Services. On-line library: www.metalink.oracle.com

[23] MSDN. (2004). Federated SQL Server 2000 Servers, MSDN. Microsoft library. On-line: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/architec/8_ar_cs_4fw3.asp

[24] Silberschatz, A., Korth, H. F. & Sudarshan S. (2002). Database system concepts (4th ed.). McGraw – Hill.

[25] Zildžić, Dž. (2005). Workload performance measurement and tuning of high-availability Oracle databases. Masters Dissertation. University Malaya, Malaysia.

AUTHOR BIOGRAPHY

Pinki Singh is a M.Tech (CSE) student at the Database and Information Systems Group at the University of Swami Vivekanand shubharti university with interests on meta data management for autonomic database tuning.

Rakesh Kumar singh is a M.Tech (CSE) Asst. prof. at the Data-base and Information Systems Group at the IIMT college, greater noida. His current research focuses on management, evolution and application of formalized tuning knowledge for autonomic database tuning.

Prof. Jayant Shekhar is a Principal in Swami Vivekanand Shubharti University. His current research focuses on management, evolution and application of formalized tuning knowledge for autonomic database tuning.