

# Performance Benchmarking of Geospatial Applications

Vartika Bajpai

**ABSTRACT** - Geospatial applications deal with large volume of datasets. In web based geospatial applications the datasets need to be transferred from server to client tiers as fast as possible. Hence the main concern while developing these applications is Performance. This paper describes a workload model for benchmarking performance of geospatial applications. The proposed model simulates a workload for N number of concurrent users and N is varied each time. The Performance metrics which are evaluated are Response Time, CPU utilization and Wait time, which are then used to identify problems of scalability aspects under heavy load. The case study of Karnataka geo-portal has been considered to analyze the performance of large web-based geospatial applications.

**Keywords** : benchmarking, geospatial, performance, workload.

## 1.0 Introduction:

In computing, Benchmark is the act of running a computer program, or a set of programs, or other operations, in order to assess the relative performance of an object, normally by running a number of standard tests and trials against it. There are 2 main kinds of benchmark: functionality benchmark and performance benchmark [3]. functionality benchmark means to test the purpose and procedures with which the system has been designed. And performance benchmark means testing the speed of the system. Assessing relative performance of applications would be impossible without benchmarks. Benchmarks are designed to mimic a particular type of workload on a system. In case of web based Geospatial applications dealing with vector spatial data sets, benchmark for measuring the performance should satisfy the following requirements [3]:

The database queries which are used for creating a benchmark should not be time consuming.

The benchmark should be able to cover large range of GIS functions as well.

This paper describes a workload model to perform a load (performance) testing on the vector spatial database. The workload parameter used here is the *number of users* of a geo-portal or web based geospatial application. The workload is increased in every iteration to analyze the performance of the database server. The proposed model covers wide range of GIS functions so that the performance measurements can be made accurately. As the paper involves the case study of Karnataka geo-portal therefore the database is populated using the Karnataka geo-portal data on Oracle 11g.

## 2.0 Benchmarking Methodology:

In our approach for developing benchmark, we have used multithreaded model as threads are light weight and are not time consuming and the queries used in the benchmark cover wide range of GIS functionalities and are parameterized. Parameterized queries are used so that every time different data blocks are fetched so the correct load on database can be measured. None of the queries, used in the benchmark, have high elapsed time.

Fig.1 depicts the basic architecture for the workload model. As mentioned above the workload architecture is based on a multithreaded model, the simulator indicated above in the Fig. is a java application which handles N number of concurrent users and maintains a cached set of shared and reusable connection objects. For each user request a thread is created by the simulator as thread is a light weight process. So whenever user request is received a thread is created for that request and a pool is searched for available connection object. If the connection object is free it handles user request otherwise the request is queued up and user has to wait to get free connection. After getting connection user randomly fires all the spatial queries from query set. The response time is recorded for each user and load on the database is also analyzed. When user has finished its request the thread corresponding to that user is killed and connection object is returned to the pool. All the users are concurrently firing queries and per query connection object is created.

### 2.1 Queries Used

As mentioned above that queries which are to be used in benchmark should cover wide range of spatial features and should not be time consuming. Keeping this in mind the queries which have been used in our model cover wide range of spatial functions such as

- Containment operations

*Line inside polygon:* For example National Highways

passing through districts District name is parameterized so that every time different district values are sent.

*Polygon contains Polygon:* For example villages which are inside district.

- Overlap Operations

*Polygon overlap line:* Retrieve all those roads which are partly inside district.

- Intersect Operations
- Buffer Operations

Finding locations of different facilities such as post-office, hospitals etc which are within a specified distance of some Taluks.

- Adjacent Operations

*Polygon Adjacent Polygon:* Locations of all those villages which are bordering some Taluk.

All the queries used are parameterized so that every time different data blocks are fetched and accurate measurement of CPU utilization can be made.

## 2.2 Performance Measurements:

The Performance parameters which are measured are as following:

**2.2.1 CPU Utilization:** The percentage of CPU, the spatial database is consuming is measured using XPERF command. Before running the java application the following command is executed on the command prompt:

*xperf -on DiagEasy.* DiagEasy is the parameter which gives the CPU usage for the particular process running on the system.

*Xperf -d results.etl.* This command generates an event trace file which contains the percentage of the cpu usage of the all the process which were running on the system (Fig.2)

**Response Time and Wait Time:** The wait-time and the response time for each user (each thread) were recorded using Netbeans Thread Profiler. Fig. 3 depicts the scenario when 5 users request for connection and fire different spatial queries. As in the application per user one thread is created therefore threads are shown in the Fig. and their running times and wait times are also shown.

## 3.0 Results and Conclusion:

This paper presented the design of benchmark tool capable of evaluating the performance of web-based geospatial applications against certain workload parameter. This helps in decision making for choosing spatial database while developing geo-portals so that the web based geospatial applications do not suffer in real use. The

approach describes the test scenario, performance metrics evaluation. The experiments have been executed with increasing the number of users on every iteration to measure the response time, wait time and CPU usage by the database. In future this can be extended by including types of query set, simple query set and complex query set. Another constraint which can be added is not allowing the user to fire a single query more than once as in the current approach the user is randomly picking up queries so there is a possibility that sometimes he may pick the same query again so the data blocks would be the same. The table 1 shows the performance metrics measured by increasing the number of users on every iteration.

## Acknowledgement

I would like to thank GISE Advance Research Lab, IIT Bombay for their support and guidance.

## References

1. "The Transaction Processing Council," <http://www.tpc.org>
2. "The Open Source Database Benchmark," <http://osdb.sourceforge.net>.
3. Norman W. Paton, M. Howard Williams, Kosmas Dietrich, Olive Liew, Andrew Dinn, Alan Patrick, "VESPA: A benchmark for vector spatial databases," Lecture Notes in Computer Science, Vol. 1832/2000, pp.81-101, 2001.
4. Suprio Ray, Bogdan Simion, Angela Demke Brown, "Jackpine: A Benchmark to Evaluate Spatial Database Performance".
5. Oracle® Database Performance Tuning Guide 11g Release 2 (11.2)
6. Oracle® Database Concepts 11g Release 1 (11.1)
7. Oracle® Spatial Developer's Guide 11g Release 1 (11.1)
8. "Window Performance Toolkit," [msdn.microsoft.com/en-us/performance/cc825801.aspx](http://msdn.microsoft.com/en-us/performance/cc825801.aspx)
9. J. Myllymaki and J. Kaufman, "Dynamark: A benchmark for dynamic spatial indexing," in Proc. of the 4th Intl Conference on Mobile Data Management (MDM), January 2003, pp. 92-105.
10. A. Crolotte, "Issues in benchmark metric selection," in Performance Evaluation and Benchmarking, ser. Lecture Notes in Computer Science, R. Nambiar and M. Poess, Eds. Springer Berlin / Heidelberg, 2009, vol. 5895, pp. 146-152.

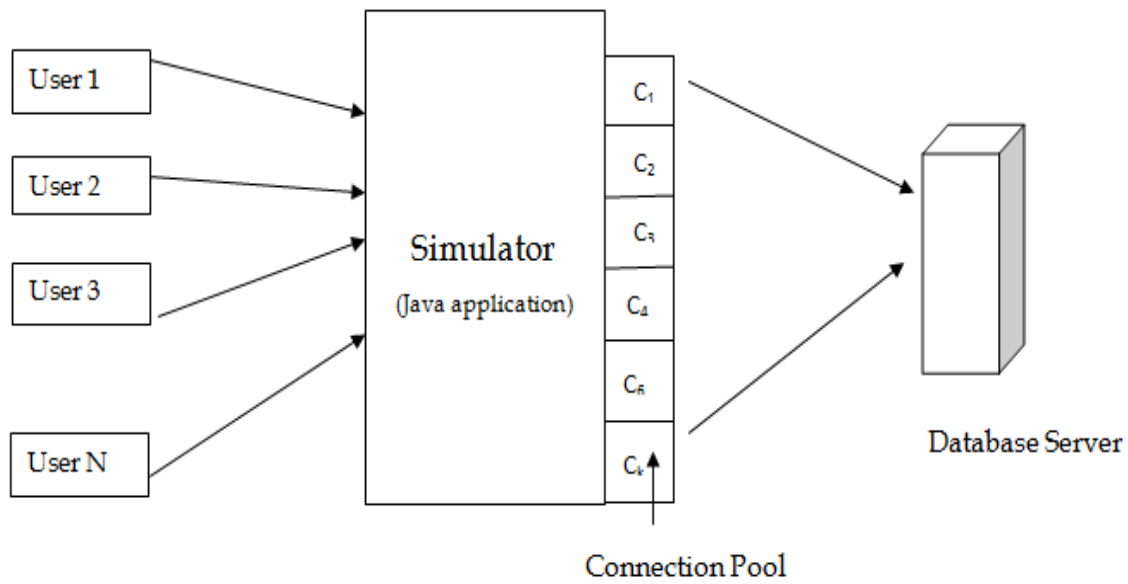


Fig. 1 Architecture for Workload Model

The screenshot shows the 'CPU Scheduling Aggregate Summary Table' in Windows Performance Analyzer. The table lists various processes and their CPU usage. The 'oracle.exe' process is highlighted with a red circle, showing a CPU usage of 64.83%.

Line	Process	Thread ID	Thread Start Module	Thread Start Function	Cpu Usage (ms)	% Cpu Usage	% Relativ
1	oracle.exe (1968)				3,369,133.057 418	64.83	
2	Idle (0)	0	ntkrnlpa.exe	Unknown	1,245,001.230 097	23.96	
3	ekrn.exe (1676)				138,380.115 212	2.66	
4	xperfview.exe (6080)				78,732.814 347	1.51	
5	java.exe (6112)				51,073.265 872	0.98	
6	TNSLNSR.EXE (1928)				35,038.348 509	0.67	
7	dwm.exe (320)				31,065.722 715	0.60	
8	System (4)				26,188.186 167	0.50	
9	chrome.exe (5940)				25,596.078 158	0.49	
10	chrome.exe (4140)				21,635.271 096	0.42	
11	firefox.exe (4268)				17,999.730 669	0.35	
12	explorer.exe (1872)				11,085.119 677	0.21	
13	svchost.exe (860)				10,751.718 225	0.21	
14	LMS.exe (4356)				9,741.476 265	0.19	
15	conhost.exe (4528)				8,341.004 155	0.16	
16	csrss.exe (520)				7,806.224 374	0.15	
17	SynTPEnh.exe (2980)				7,614.056 163	0.15	
18	chrome.exe (3368)				6,913.667 933	0.13	
19	svchost.exe (3088)				6,052.914 866	0.12	
20	chrome.exe (5464)				5,365.361 331	0.10	
21	chrome.exe (448)				4,818.391 865	0.09	
22	lsass.exe (584)				4,195.965 512	0.08	
23	svchost.exe (828)				3,306.684 468	0.06	
24	sidebar.exe (3572)				2,336.669 460	0.04	
25	java.exe (640)				2,002.115 963	0.04	

Fig. 2: CPU utilization consumed by the Database.

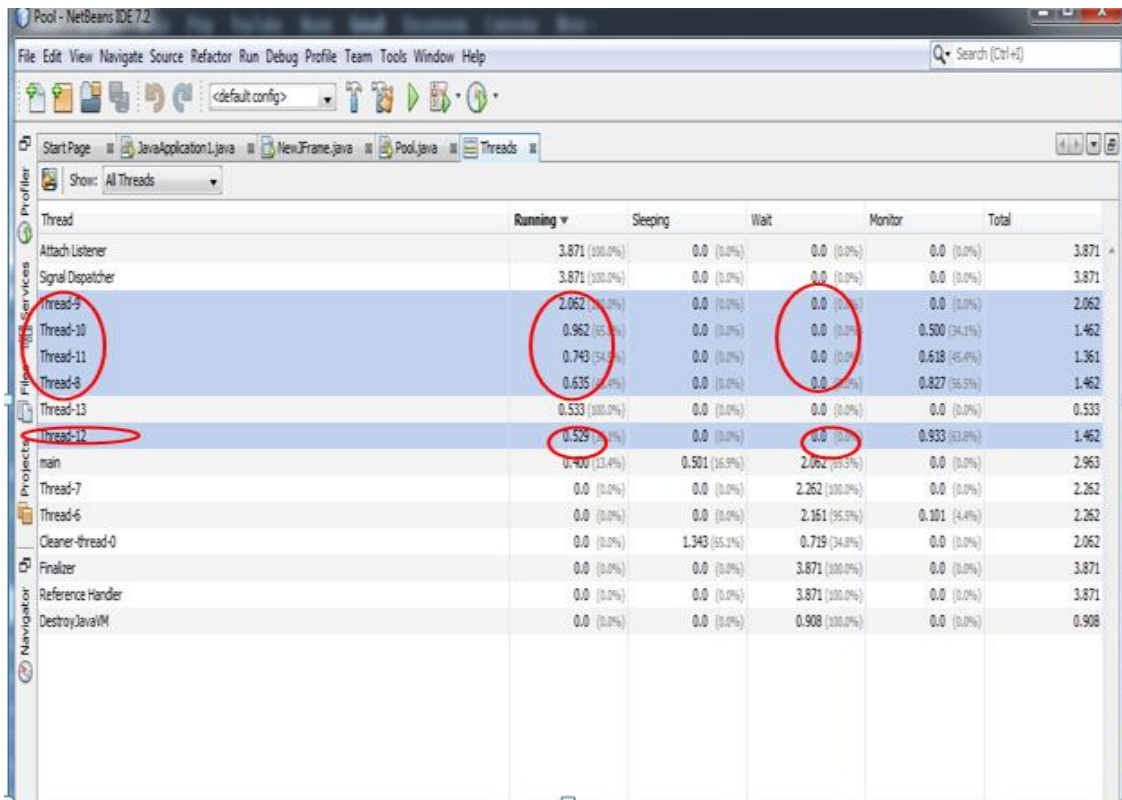


Fig. 3: Response Time and Wait Time using Thread Profiler

No. of Users	CPU usage	Avg. response Time per user	Avg. Wait Time per User
50	61.29	2.51	0
80	70.58	2.66	1.21
110	71.44	2.89	2.3
140	71.56	2.98	3.07
170	69.59	3.08	3.57
200	70.30	3.19	3.63
230	67.56	3.47	4.05
250	69.35	3.73	4.38

Table 1: Benchmark Result



Vartika Bajpai