Distributed applications monitoring at system and network level

Monarc Collaboration[1]

Abstract

Most of the distributed applications are presently based on architectural models that don't involve real-time knowledge of network status and of their network usage. Moreover the new "network aware" architectures are still under development and their design is not yet fully defined.

We considered, as use case, an application using ODBMS (Objectivity/DB) for distributed analysis of experimental data.

The dynamic usage of system and network resources at host and application level have been measured in different client/server configurations, on several LAN and WAN layouts.

The aim was to study the application efficiency and behaviour versus the network characteristics and conditions. The most interesting results of LAN and WAN tests are described.

The monitoring results identified system bottlenecks and limitations and efficient working conditions in the different scenarios have been defined; some critical behaviours observed when moving away from the optimal working conditions are described.

The analysis of the data gathered in the tests have been done off-line. New tools able to visualize on-line the resource usage will give real time information on bottlenecks, which may arise in all the system components, network, host or application itself, and therefore make easier troubleshooting or admission control policy. The current status of tools development is described.

1 Introduction

The HEP communities that need to access and analyse large volumes of data are often large and are almost always geographically distributed as are the computing and storage resources that these communities rely upon to store and analyse their data. This combination of large dataset size, geographic distribution of users and resources and computationally intensive analysis results in complex and stringent performance demands that are not satisfied by any existing data, CPU and network management/monitoring infrastructure. This article describes how the analysis of measurements regarding resource utilization in a distributed environment: CPU usage, network throughput and other parameters as wall clock time of a single job, has identified system and network bottlenecks, software and hardware inefficiencies in different scenarios.

The tests based on with Objectivity 5.1, were part of the activity of the MONARC [1] testbeds working group. The work is still in progress and further developments are foreseen.

Different network scenarios have been set up based on a single federated database, one AMS server and several clients locally or geographically distributed. The client jobs perform sequential read operation from the Data Base. Measures of CPU utilization on Server/Client workstations and network throughput with different number of jobs have been collected and discussed. Future test scenarios have been proposed.

2 Test objectives and description

The distributed analysis of the experimental data can be severely affected by the network for several reasons:

1. overhead due to communication protocols;
2. network throughput can change significantly depending on TCP flow control parameters;
3. application protocols: how client/server exchange data and behaviour in case of network load and congestion;
4. network speed and system capability to use it;
5. end-to-end delay and relationship with link speed and throughput.

The tests described in this article are significant concerning point 4 and 5. In order to investigate the first points it would be necessary to know the details about both Objectivity architecture and application software implementation.

Tests are based on several client/server configurations over different LAN and WAN scenarios with network speed ranging from 2Mbps up to 1000Mbps. Moreover, some tests have been performed in a WAN scenario supporting QoS/Differentiated Services architecture. Test results have been compared and discussed.

The most important specific objectives are:

- check Objectivity AMS behaviour and performance;
- perform stress test by running several analysis jobs accessing the Data Base;
- locate system bottlenecks;
- collect 'response time' measures to give input to computing modeling and simulation;
- understand network traffic characteristics and profiles.

The general test scenario is quite simple regarding to database characteristics and structure. A fast simulation program developed by ATLAS collaboration, Atlfast++ program [2] is used to populate an Objectivity database following the Tag/Event data model proposed by the LHC++ project; one single container per event and no associations in the database.

A single Objectivity federation with one AMS server and many clients, has been populated with 50.000 events, $\sim$40Kbytes each, corresponding to about 2Gbytes of data. Objectivity 5.1 has been used setting the page size at 8192 bytes. In the following two system configurations are examined:

- 1 server / 1 client
- 1 server / many clients.

The network capacity is variable starting from 2Mbit/sec up to 1000 Mbit/sec: LAN tests have been performed at 10Mbps, 100Mbps and 1000Mbps; WAN tests has been done in production environment, at bandwidth from 2Mbps up to 8Mbps and in a QOS/Differentiated services 2Mbps dedicated ATM/PVC.

The client job reads $\sim$3000 events. The LAN FEthernet tests has been done in the INFN-Roma Babar farm and in this case the client job reads $\sim$10.000 events. Stress tests have been performed: the procedure followed consists in submitting an increasing number of concurrent jobs from each client workstation and then monitoring CPU utilization, network throughput and single job execution time (wall clock time). The same kind of tests have been performed on a local federated database (without AMS server) [3]

3 Application monitoring tools

The system parameters that have been selected to be collected and evaluated are:

- Client side: CPU use (by user and system), job wall clock time;
- Server side: CPU use (by user and system), network throughput.

These parameters are significant in distributed application for the following reasons:

- CPU use in client machine is important to evaluate machine load versus number of concurrent jobs with different link speed;
- CPU use on Server is important to evaluate the maximum number of client-jobs that can be served and if this number is related with client characteristics and network link capacity;
- Wall clock time execution is important to evaluate system capacity to deliver workload in connection with the number of jobs and network speed.

The client and server CPU usage is collected issuing periodically 'vmstat' command.

The application program itself records the elapsed time, while the aggregate server throughput is collected tracing the AMS server system calls (read/send and write/receive are the system calls recorded). Every two minutes, with a timestamp, a script write in a log file the number of bytes read from the local disk and sent to the client jobs via network connections. It is possible afterward calculate the effective aggregate throughput from server to the client machines.

A series of scripts have been written in order to collect the parameters from the machines (clients and server) and elaborate the data.

4 Test results

The details of the performed tests have been collected in many working conditions and the most interesting results has been selected and summarized in the following. The table below I shows Max CPU utilization of clients and server, together with the corresponding number of running jobs versus network speed.

Network Speed	**CLIENT**		**SERVER**	
	Max CPU Use	**Number of jobs running**	**Max CPU Use**	**Number of jobs running**
1000M (GE)	100 %	≥ 5	100 %	≥ 50
100M (FE)	60 %, then 20 %	Up to 30, then up to 60	100 %	Up to 60
10M (Eth)	80 %	≥ 20	30 %	≥ 60
2M (PPP ATM WAN)	5 %	Up to 20	10 % (constant)	1-20 (during the all test)

Table I: Test results

The general description of these data values is the following: in a Gigabit Ethernet LAN, the Client CPU (Sun Ultra 5 14 Specint95)is saturated with 5 concurrent analysis jobs. In a Fast Ethernet LAN, where the client machine has higher CPU power (Sun E450, 4 CPUs each with 17 Specint95) the bottleneck is the CPU of the server machine serving 30 concurrent jobs in the client machine. The server machine in this Fast Ethernet test is a SunE450 with 4 CPUs as the client host, but Objectivity 5.1 AMS server is able to use only one CPU. In a Ethernet LAN, the critical resource is the network bandwidth that is completely used. Regarding the network throughput the results are summarized in table II.

In GEthernet LAN the client CPU is 100up to 20 concurrent jobs. Network utilization is optimal for an Ethernet LAN and for 2Mbit/sec ATM PVC with PPP protocol encapsulation, while

Network Speed	Max Throughput	Number of jobs
1000M GEthernet	37Mbps	≥ 20
100M FEthernet	80Mbps	≤ 30
10M Ethernet	9Mbps	≥ 20
2M VC ATM	1.7Mbps	≥ 20

Table II: Network throughput

Gigabit Ethernet network utilization percentage is very low and it must be investigated with future release of Objectivity(5.2) and with more powerful client and server machines.

Regarding the wall clock time elapsed during the execution of a single job, in order to compare the results between the tests, an average wall clock time for one job has been calculated taking into account two conditions: 10 concurrent jobs in the client machine and only one job.

- Gigabit Ethernet LAN: average wall clock time : 360 sec, single job 60 sec.
- Fast Ethernet LAN: average wall clock time: 150 sec, single job 48 sec.
- Ethernet LAN: average wall clock time 1000 sec, single job 200 sec.
- 2Mbit/sec ATM PVC: mean wall clock time 6000 sec, single job 1000 sec.

It could be interesting to enhance that, with the same CPU power conditions, wall clock times, from GEthernet down to 2Mbit/sec, decrease with the same factor as throughput (as it was expected): wall clock time in Ethernet LAN is 2.5 times wall clock time in the GE LAN and the same factor is observed between the two measured network throughputs. The wall clock time in 2Mbit/sec tests is 6 times the wall clock time in Ethernet LAN and similar factor (5.6) is between effective throughputs. Fast Ethernet LAN is an exception since the server and client machines are more powerful, with different architecture.

5 Conclusions

These tests provide a description of Objectivity behaviour on different network layouts, with different link characteristics, in terms of CPU behavior, link throughput and job execution time measures. SUN single and multiprocessor systems have been used.

The inability of Objectivity AMS 5.1 to use multiprocessor systems represents a severe performance limitation in the Fast Ethernet LAN network test. The high CPU usage also on SUN multiprocessor clients running over Fast Ethernet LAN enhances that Objectivity implementation is heavy and it could be improved.

An important parameter of the different configurations is the number of connections on the server and the optimal measured values corresponds to 30 concurrent jobs, that is too small for a distributed analysis of experimental data in a production environment.

Analyzing the results it is possible to identify some boundary conditions for an efficient running of the jobs, with the specific CPU.

Let us suppose that an 'acceptable running condition of the job' is when elapsed wall clock time is less then 10 times the wall clock time of a single job.

On the basis of the measured parameters, a scenario should be based on links with a minimum speed of 8Mbps between client and server. Client machines should run from 6 up to 15 max concurrent jobs and Server should deal with requests of 30 concurrent jobs as a maximum. A general consideration is that global system performance degrades rapidly moving away from optimal condition.

Application monitoring able to real time check the working conditions, network throughput,

CPU usage, job execution time is needed to have the possibility to take the necessary action to maintain the system around this optimal condition.

New tools able to provide real time information about resource usage are under development.

6 Future works

Objectivity 5.2 features will probably override some of the performance limitations and it should be able to use multiprocessor systems in efficient way.

It has been planned to repeat the tests in LAN with 4 SUN machines and the AMS server configured in a SUN E450 multiprocessor system connected both using Fast Ethernet and Gigabit Ethernet links.

Since system behaviour in LAN at 10Mbit/sec has been considered as the lower threshold for acceptable job elapsed time, new tests will be performed over a dedicated WAN at 10Mbps in order to investigate both multi-server configuration and the comparison between LAN and WAN behaviours. The comparison between LAN and WAN at the same speed is very interesting to investigate the influence of WAN latency on the system performance and network protocol tuning.

Since 100Mbps allows good job wall clock time and seems to be a reasonable WAN speed, WAN layout at 100Mbps would be very interesting for testing or prototyping.

References

1 Monarc project, `http://www.cern.ch/MONARC`.

2 `http://atlasinfo.cern.ch/Atlas/GROUPS/PHYSICS/HIGGS/Atlfast.html`.

3 Preliminary Objectivity tests for MONARC project on a local federated database, MONARC internal Note, 25 May 99.