DISTRIBUTION OF TASKS IN THE GRID, TOOL TO OPTIMIZE LOAD

LUKASIK, P. & SYSEL, M.

Abstract: *The article focuses on the problem using a grid computing as a tool for optimizing the load master server that is deployed in planning and production control. The motivation for the use of the grid is a uneven load on the master server and on the end-user workstations.*

Problem is solved so that some algorithms of the ERP system are executed on the workstations in the grid. The improvement was mainly achieved by batch jobs that system was replaced by remote objects. These objects carried out some of the standard tasks of the planning process on the user's workstation. The result is optimal load distribution of the server in a time. The article discusses the possibility of using JSDL (Job Submission Description Language) as a tool to facilitate the distribution of tasks in Grid environment.

Key words: Intranet Grid, Planning, JSDL, Grid Scheduling

Authors´ data: Doc. Dipl.-Ing. **Sysel**, M[artin]; Ph.D, Dipl.-Ing. **Lukasik**, P[etr], Department of Computer and Communication Systems, Faculty of Applied Informatics, Tomas Bata University in Zlin, nam. T. G. Masaryka 5555, 760 01, Zlin Czech republic, plukasik@tajmac-zps.cz, sysel@fai.utb.cz

1. Introduction

Some of the basic features that are part of the ERP systems are implemented as the batch jobs. These are jobs that have high demands on computing system load and during the workday is difficult to find a suitable plan for their activities.

The aim is to implement a solution that converts these tasks from batch to on-line processing mode. This design concept watches especially better distribution of computational load and improves the planning process. Any imbalances in the system are solved immediately. Therefore, we chose the tasks that have a direct impact on the balance sheet resources and requirements in the process of production.

Tasks for the solution:

- Low-level code calculation. The low-level code defines the lowest level of usage of a material in all tree product structures. Low-level code determines direction of planning.
- Production planning – MRP (Material Requirement Planning) algorithm, is a computer-based inventory and production management system designed for scheduling of the production.
- Product cost calculation – Calculation of the production costs, based on knowledge of the product structure (BOM) and costs of the components entering into the product.

2. Requirements on the project conception

The basic requirement is to optimize the power load on the master server and move some of the tasks on the client workstations. Other requirements are formulated in the following paragraphs:

- Independent on the hardware platform and operating system.
- Independent on the database platform.
- Easy definition for another type of the job.
- Possibility of solving some different types of tasks at once.
- Installation on the IBM iSeries model 520.

3. The main objective for grid use - distribution of computational load

Measuring the load of the individual components of the network infrastructure, we found differences between the load of servers and end-user workstations. (Fig. 1, Fig. 2).

Therefore, there is a possibility to move part of computing power on the client workstation using a grid infrastructure. Better load of the distribution on the server, will increase the throughput of the entire system.
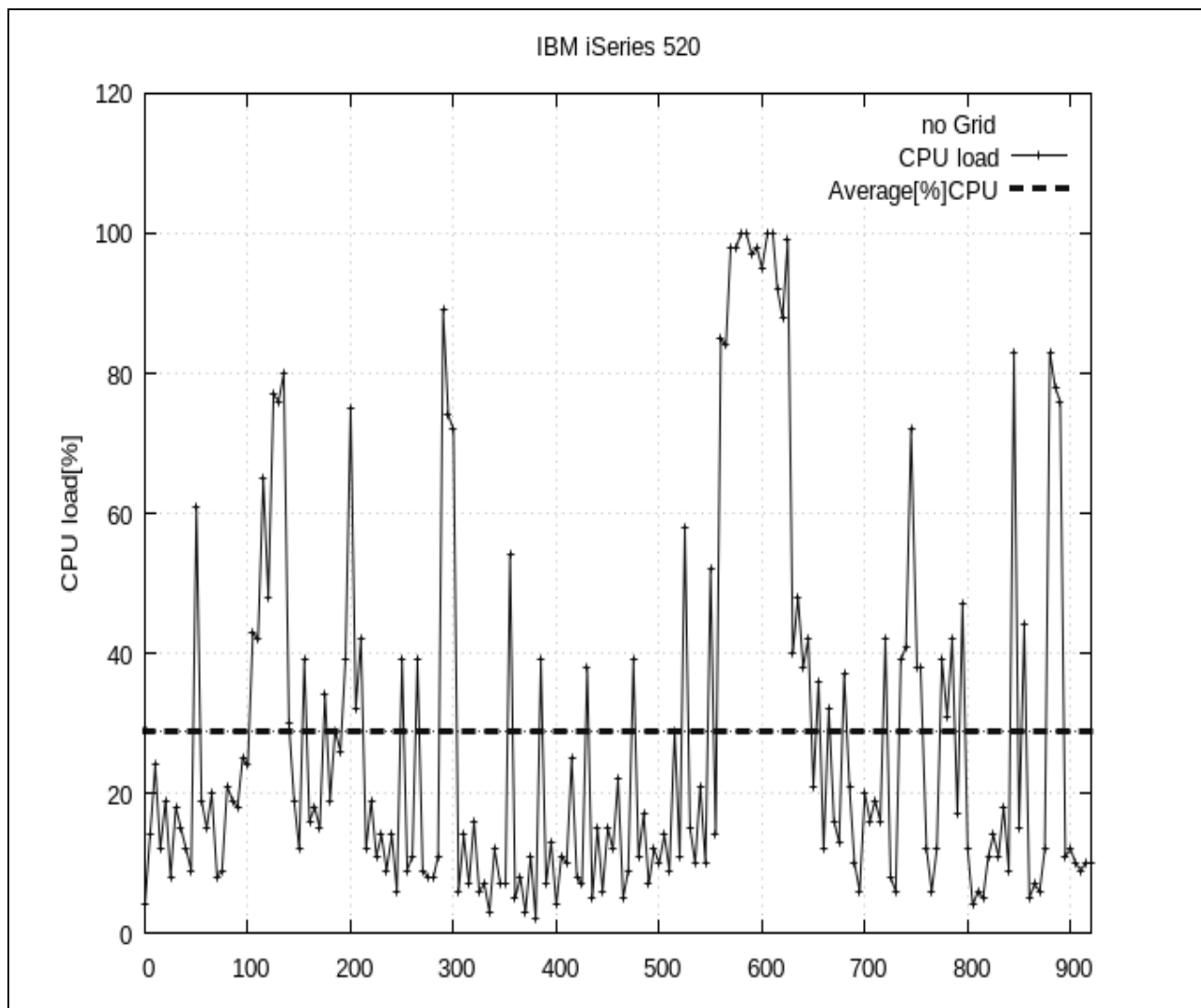
Fig. 1. CPU load on the IBM eServer without Grid

Measurement is carried out by using the features offered by each operating system. We observed the CPU load. For the IBM eServer we use component of **System i Navigator**. For the end-user workstation and Linux we use **Sar** (System Activity Report). On the Windows OS, we use **Typeperf** utility.

4. Principle of the grid

The task distribution is managed by system messages. These messages are generated by SQL triggers that intercept changes in the data model of the ERP system.

It was planned to deploy it only within the corporate Intranet. Therefore there were no high demands on the security policy operation. Java RMI technology was used for the distribution of program objects to the clients. It was also considered for the CORBA standard, which is not limited to Java objects. This technology was rejected for higher complexity concept.

Grid Server – maintains a current list of clients connected to the network and ensures distribution of tasks using the Task Scheduler. Apache Commons DBCP

project is used to the database connection. This project optimizes and increases permeability of database operations. Grid server application is designed to consume the least processing load and system resources of the main server.

Job Scheduler – is designed as a FIFO queue to determine the tasks for clients. After emptying the queues other tasks are loaded. Task Scheduler tries to optimize system resource at the lowest level, while trying to maintain the optimal computing load across the grid system.
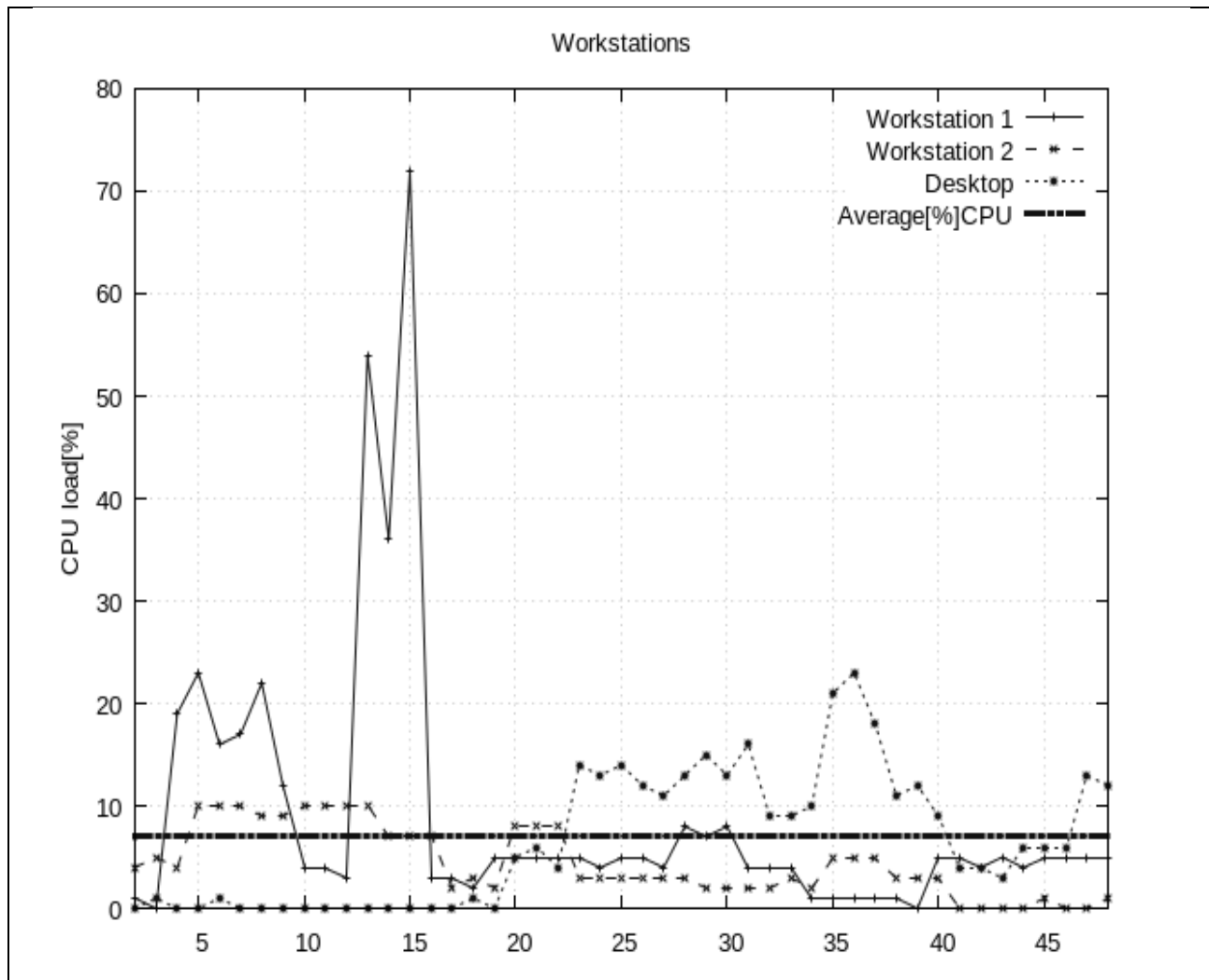

Fig. 2. CPU load on the workstations

Scheduler strategy:

- Optimum performance is controlled by the speed loop. Speed loop has a feedback control depending on CPU load. If a hardware load is growing, the loop scheduler slows down.
- A strategy for assigning tasks is managed, based on their priorities. Priority is dependent on the type of the job. For example, LLC task must be solved sooner than the MRP. Therefore, calculations of low-level codes have priority over the other tasks.

- Each client has the same priority. The concept of client grid is designed so that you can process multiple jobs at the same time. Number of parallel tasks depends on the performance of the client hardware.
- Resolves collision conditions during the fall task on the client.

Grid Client – shown in Fig. 3 is ready for the multi-threaded support. The motivation for this approach is the fact that in the case of a small number of clients currently connected to the grid should be able to enter multiple tasks to one client at a time. Allocation strategy of threads is depending on the performance of the client side hardware. If a client has a low power, scheduler provides a maximum two tasks at once to him. The maximum number of threads for the grid client on the workstation is limited to four.
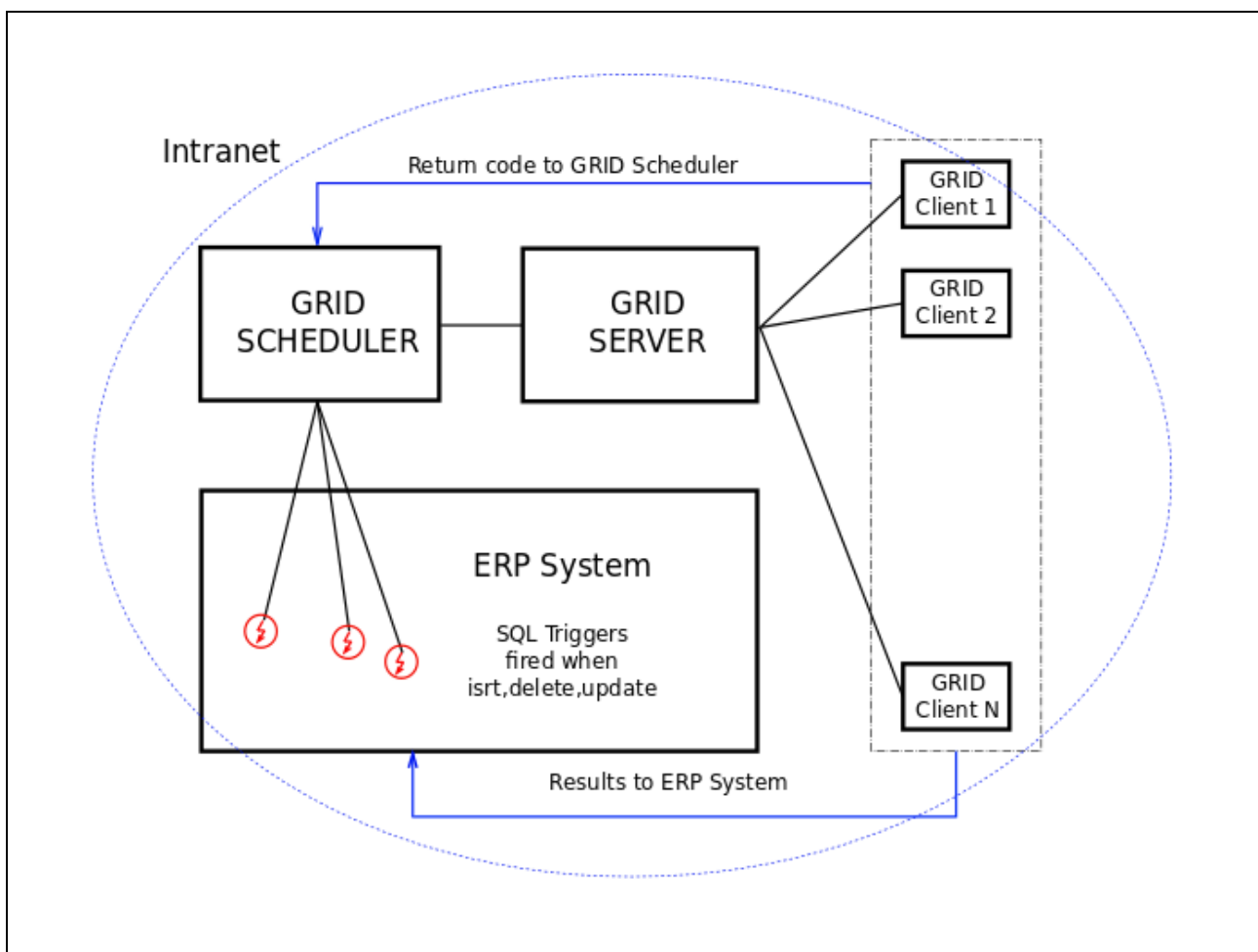

Fig. 3. Grid scheme

Above the data system planning and management are defined SQL triggers, which capture the events with an impact to change the data consistency. For example, such a change is a process of opening the production charge. This will disturb the balance in the planning database. Batch job returns balance between customer demands, product inventory, and production levels. This cannot be done after every change in the database. Planning tasks can be run in a selective or regenerative mode,

when the entire database is rescheduled. However, both modes are challenging for a time computing.

Intranet grid, (as a substitute for batch jobs), reschedules the specific item immediately at the time when the change occurred. This means that the database is permanently in a consistent state. The event is captured by a trigger and recorded in the grid scheduler. Scheduler sends the task to the first available client. The client calculates the task and returns the result to the planning database.

Disadvantage of this solution is a higher load of the network traffic. The positive effect is a better distribution of power to the main server.

5. Result

The distribution of tasks in a grid improves load of distribution server of very little value, max. 2-3%. Most of the computing power consumes database service.

The method described above allows a better distribution of the load over time. This means that it subsequently avoids overloading of the system. This result meant significantly better availability of the ERP system. The result is shown in Fig. 4.
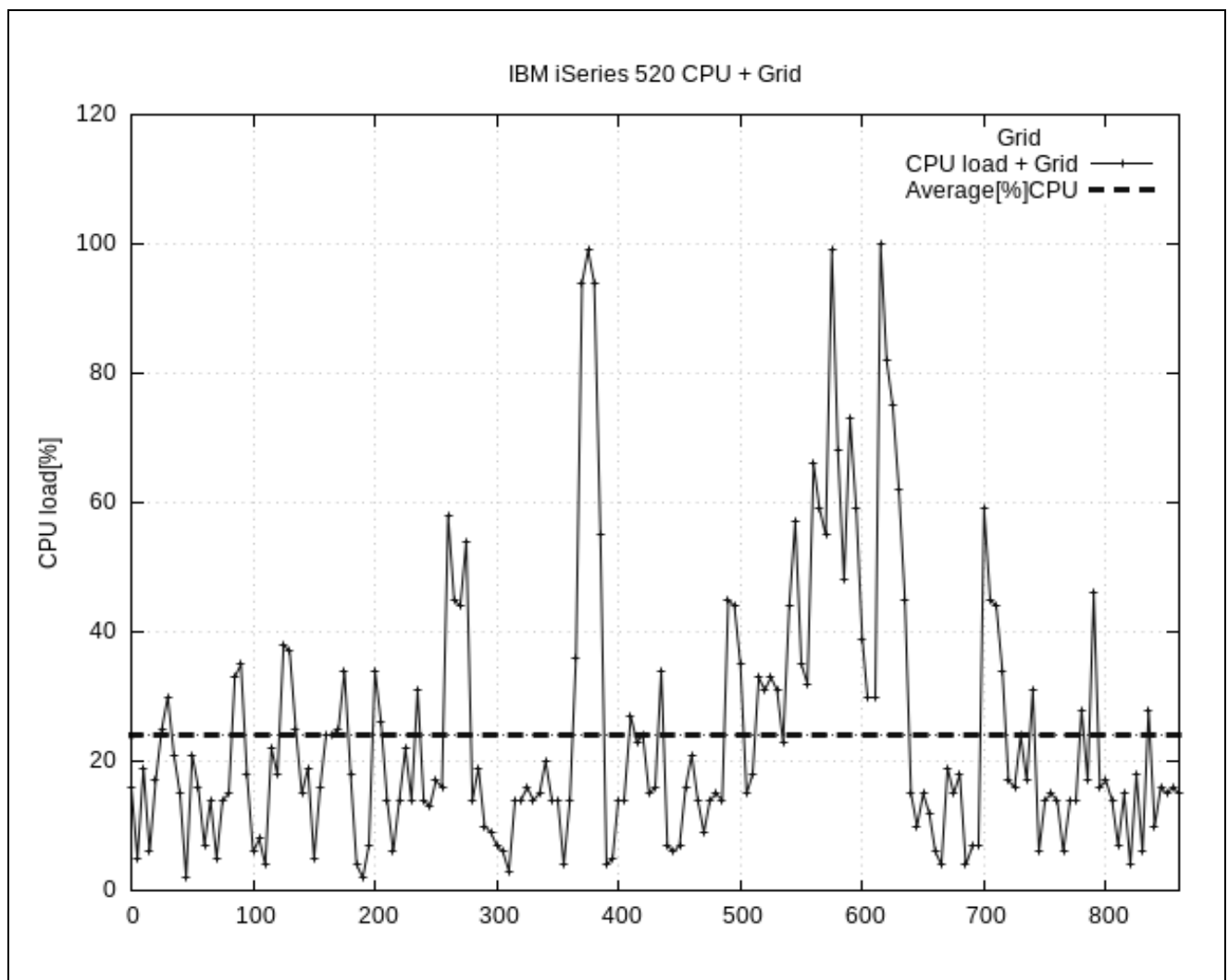

Fig. 4. CPU load on the IBM eServer with Grid

6. Discussion

Distribution of database to a cluster environment can improve the above results. The project BigSQL uses the Hadoop cluster together with SQL database PostgreSQL.

This idea was not carried out. It would mean a complete redesign of the existing ERP system. This system is in the production deployment and it is not possible to increase the risk of incorrect functionality.

It was considered to increase the reliability of Grid services. It offers a solution that does not use native workstation users. They represent an expanding risk of data loss such as shutdown or system crash on the client side of the grid.

This risk can be minimized by building a grid infrastructure clients based on cheap Mini PC systems. They have very good hardware performance. The advantage is that this can be installed in a uniform system to all clients. Although the disadvantage is that this approach does not use much power reserve workstation users. This is offset by much higher reliability, since it will reduce the amount of conflict situations.

To increase reliability, is planned to build a client farms based on cheap mini pc systems. This solution is also planned for use in large-scale calculations in the design of machine tools using Hardware in The Loop. The aim is to replace the very expensive computational server, which is currently used.

For this implementation will be necessary to change the concept of the Grid scheduler. It is planned to deploy JSDL specification for defining the jobs. JSDL (Job Submission Definition Language) is a general specification for the management and distribution of batch jobs (shell scripts) on the Grid. The current version 1.0 also defines a support to the POSIX. This solution is a perspective in local Intranet Grid environment. Here it is possible to define a standard interface and install specific programs needed to solve problems.

The aim is to achieve an environment that provides standard software with a relatively small investment. The environment will provide a powerful hardware platform.

7. Conclusion

The aim is to achieve better solutions time distribution of computational operations master server. Part of tasks that relate specifically to the needs of the immediate increase in power (especially batch Jobs) were referred to the clients of the grid.

Measuring the load of individual components of the network infrastructure has been found out considerable disparity between the load terminal servers and client stations.

Reducing and better load distribution server time increased throughput of the entire system. The advantage of this concept is also the online data consistency in the ERP system. The disadvantage is of course larger load of network communication.

8. References

Pavlech, Michal. "Framework for Development of Distributed Evolutionary Algorithms Based on MapReduce." Annals of DAAAM & Proceedings. DAAAM International Vienna. 2011

Niansheng Cheng; Yisheng Zhang, "Intranet Based Security Infrastructure for Manufacturing Grid," Genetic and Evolutionary Computing, 2008. WGEC '08. Second International Conference on , vol., no., pp.247,250, 25-26 Sept. 2008 doi: 10.1109/WGEC.2008.91

Lenk, A., Klems, M., Nimis, J., Tai, S., & Sandholm, T. (2009). What's inside the Cloud? An architectural map of the Cloud landscape. In Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing (pp. 23-31). IEEE Computer Society

Hernando Bedoya, Fredy Cruz, Daniel Lema, Satid Singkorapoon: Stored Procedures, Triggers, and User-Defined Functions on DB2 Universal Database for iSeries, IBM RedBook, (October 2006)

Bart Jacob, Michael Brown, Kentaro Fukui, Nihar Trivedi: Introduction to Grid Computing, IBM RedBook (December 2005)

Fangpeng Dong, Selim G. Akl: Scheduling Algorithms for Grid Computing: State of the Art and Open Problems, School of Computing, Queens University Kingston, Ontario (January 2006)

Information technology - Object Management Group Common Object Request Broker Architecture (CORBA), Interfaces, ISO/IEC 19500-1:2012(E) Date: April 2012

Ali Anjomshoaa, EPCC, Fred Brisard, CA, Michel Drescher # , Fujitsu, Donal Fellows , UoM An Ly, CA Stephen McGough # , LeSC Darren Pulsipher, Ovoca LLC Andreas Savva #, Fujitsu GFD-R.136 Job Submission Description Language (JSDL) Specification http://forge.gridforum.org/projects/jsdl-wg 28 July, 2008 Copyright © Open Grid Forum (2003-2005, 2007-2008). All Rights Reserved