# Virtual machine testing for computing cluster configurations optimization

Ivan Bilokon[1], Sergiy Pogorilyy[1]

[1]Faculty of radiophysics, Taras Shevchenko national University of Kyiv, 64/13 Volodymyrska St., Kyiv, Ukraine

deimos@univ.net.ua, sdp@univ.net.ua

Abstract. This paper contains configurations test results of dynamically reconfigurable cluster computing system (DRCCS) (the Microsoft Windows HPC Server 2008 R2 platform) with virtual machine (VM) nodes (the Microsoft Hyper-V platform) for both static and dynamic random access memory (RAM) settings cases. Cluster nodes central processing unit (CPU) load forecasting method investigation results are shown for different RAM types. Results of the experiments represent virtual memory operation differences for different VM RAM type settings. CPU load behavior predictability deterioration takes place for computing jobs which processes don't use whole allocated RAM.

## Keywords

Computing cluster, virtual machine, Microsoft HPC Server, Microsoft Hyper-V, fitness function, genetic algorithm.

## 1 Introduction

Universal cluster architecture of computing systems has several disadvantages related to calculations performance of various parallel algorithm software implementations. Traditionally, they are avoided using problem-oriented and dynamically reconfigurable computing systems based on programmable logic devices to perform the appropriate tasks, or using job scheduling algorithms based on statistical data about previous calculations process to allocate resources to cluster jobs. In previous papers a new approach was suggested: universal DRCCS with VM nodes utilization [1, 2]. Using VM as DRCCS nodes enables programmatic reconfiguration of the cluster resources, and therefore impacts computational process performance [3]. In order to optimize hardware-based computing resources workload the genetic algorithm (GA) for optimal DRCCS configurations search was proposed [4, 5].

In order to calculate the fitness function (FF) values it has been proposed to use CPU load performance counters of cluster nodes which have been retrieved during several key configurations testing [4, 5]. This paper is dedicated to the DRCCS behavior testing using static and dynamic RAM settings in order to improve FF values calculation methods for optimal configurations search.

## 2 Related Works

In [4] DRCCS configurations testing method via nodes reconfiguration and tested cluster job execution while performing CPU load performance counters collection was proposed. In [6, 7, 8] using the proposed method the research of software implementations of some parallel algorithms was made using dynamic RAM. The results confirmed the importance of further research in this area. In [5] GA was offered to optimize the cluster resource workload with the tasks from the queue. The GA searches for configurations where hardware CPU utilization is maximal while minimizing the competition for CPU resources between the VMs. The FF values calculation method fulfills the prediction of the CPU load for any configuration. The prediction is based on the CPU performance counters average values measured during key configurations testing phase. This work is related to investigation results of the DRCCS with static RAM and its impact on the CPU load predictability.

# 3 DRCCS configurations investigation

Experiments were executed on DRCCS [1] created on the basis of Microsoft IT Academy computer laboratory located in the Information and Computing Center of the National Taras Shevchenko University of Kyiv (8 personal computers: Intel Core 2 Quad CPU Q6600 2.4 GHz, 8 GB RAM, Gigabit Ethernet NIC, Windows Server 2008 R2 Enterprise). DRCCS nodes are VMs on the Microsoft Hyper-V platform [9] with the Microsoft Windows HPC Server 2008 R2 [10] package installed.

The LINPACK [11] benchmark with task size N = 31872, and NB = 192 was used as a test job. CPU load data collection was performed within 10 minutes from the job launch. In order to perform measurements an automatic system was created, which executes computing cluster reconfiguration and experimental data collection.

## 3.1 Description of the experiment

In order to retrieve data for FF construction key configuration research for 8 test job processes was performed. Let's mark these configurations according to the number of job processes on the VM host (VMH) "11111111" (1 process per 1 VMH)), "44" (4 processes per one VMH, 4 per the other), "8" (all the processes per the same VMH). Additionally following configurations were investigated: starting with "11111111" we investigate CPU load performance counters, gradually increasing the number of task processes per some of the VMH (configurations "11111111", "11111120",…,"80000000"). Taking into account the hardware and software configurations of the experimental facility, DRCCS configuration implementation was made as follows. If the number of processes $n$ on 1 VMH is less than 4, they are allocated to the VM of this VMH with a number of virtual CPU equal to the number of processes. With a number of $5-8$, processes are distributed across 2 VMs of corresponding VMH with CPU quantities 4 and $n-4$ accordingly.

During the CPU load prediction the computer system simulation isn't implemented, but instead recursive search among the previously calculated values is fulfilled for those configurations, which can be defined via three parameters: $r$ (total amount of processes), $p$ (maximum amount of processes per VM), $g$ (number of VM per VMH) [4, 5].

## 3.2 Dynamic RAM

Table 1 provides the results of the configurations tests using dynamic RAM [12] (StartupRam=512 MB, MaximumRam=65536 MB, MemoryWeight=5000). According to the table and hardware configurations of the experimental facility, in case of DRCCS configurations with the number of processes per VMH greater than 6, the lack of RAM will take place. This should cause operating system (OS) to use swap file and as a result the CPU load decrease due to waiting for hard drive response. But instead we are facing the opposite phenomenon: there is no considerable waiting. This is evidence of the fact that the processes of the tested computational job actively use only some part of their allocated RAM (irrational usage of RAM) which pages are located directly in the RAM and are not written to the drive.

**Table 1.** The results of key configurations tests using dynamic RAM.

| Configuration | CPU (%) | RAM per process (MB) |
|---|---|---|
| 11111111 | 68 | 1834 |
| 44 | 63 | 1204 |
| 8 | 67 | 2326 |

To study the system behavior and to predict CPU load values for DRCCS configurations, table 2 shows results of additional configurations tests and the predicted values. Analyzing obtained data a conclusion can be made that such behavior of CPU load is unpredictable without information about software implementation details (topology of inter-process communications, RAM resources allocation etc.).

## 3.3 Static RAM

To simplify the tested system instead of dynamic we will use static RAM. This greatly complicates the reconfiguration process due to following factors:

**Table 2.** The comparison of measured and predicted values for the configurations with dynamic RAM.

| Configuration | Measured value (% CPU) | Predicted value (% CPU) |
|---|---|---|
| 11111111 | 68 | 68 |
| 11111120 | 65 | 63 |
| 11111300 | 62 | 64 |
| 11114000 | 59 | 67 |
| 11150000 | 60 | 68 |
| 11600000 | 59 | 68 |
| 17000000 | 59 | 67 |
| 80000000 | 67 | 67 |

- necessity of the explicit required RAM amount settings;
- possible VM launch errors if at the moment of the launch there will be no free RAM amount equal to the VM RAM amount in the VMH.

RAM amount setting was executed according to the following principles:

- RAM amount for VM ($R$) is equal to 90% of free VMH RAM at the moment of the free RAM amount retrieval;
- in case of configuration, which requires $n$ VM on a VMH, each VM receives $R/n$ RAM amount.

The table 3 represents the results of the configurations tests using static RAM. In case of 8 processes per VMH there is obvious CPU load decrease due to swap file utilization. Comparing to the above given results a conclusion can be made that the presence of the dynamic RAM changes the VM OS virtual memory operation behavior. This improves performance of processes, which use RAM irrationally, but this makes additional difficulties in the studying and prediction of the system's behavior.

**Table 3.** The results of key configuration tests using static RAM.

| Configuration | CPU (%) | RAM per process (MB) |
|---|---|---|
| 11111111 | 69 | 1804 |
| 44 | 74 | 1244 |
| 8 | 5 | 1142 |

Table 4 gives the results of additional configurations tests and predicted values. Starting from 6 processes per VMH CPU load decreases, which is a predictable result. Prediction in this case gives better quality results to define best/worst configuration, because predicted values reflect well the behavior of CPU load changing.

**Table 4.** The comparison of measured and predicted values for the configurations with static RAM.

| Configuration | Measured value (% CPU) | Predicted value (% CPU) |
|---|---|---|
| 11111111 | 69 | 69 |
| 11111120 | 69 | 64 |
| 11111300 | 68 | 67 |
| 11114000 | 67 | 70 |
| 11150000 | 62 | 53 |
| 11600000 | 53 | 47 |
| 17000000 | 46 | 37 |
| 80000000 | 5 | 5 |

# 4 Conclusion

The investigations results of the DRCCS configurations with VM nodes on the platform Microsoft Hyper-V for the dynamic and static RAM performed in the paper showed that:

- dynamic RAM influences the OS virtual memory operation behavior in such way that fewer RAM pages are written on the hard disc. This causes performance improvement of the software implementations, which use RAM irrationally;
- performance predictability of the parallel algorithms software implementations for the cases of the irrational RAM usage significantly decreases in case of dynamic RAM utilization because of the virtual memory unpredictable behavior which requires additional investigations.

Investigation results of the DRCCS nodes CPU load prediction method based on set of key configurations tests data showed the prediction possibility of CPU load behavior depending on test job process count per VMH.

# References

[1] I. V. Bilokon, D. B. Gryaznov, S. D. Pogorilyy: Building of dynamically reconfgurable computing architecture using virtualization technologies. *Bulletin of the Taras Shevchenko National University of Kyiv, Series: Radiophysics and Electronics*, 14: 4-7, 2010.

[2] С. Д. Погорілий, І. В. Білоконь: Технологія віртуалізації. Динамічна реконфігурація ресурсів обчислювального кластера. *Математичні машини і системи*, 3: 3-18, 2012.

[3] I.V. Bilokon, D.B. Gryaznov, S.D. Pogorilyy: Testing of computing system configuration impact on parallel algorithm software implementations. *Proceedings of the VII International conference "Electronics and applied physics", October, 19-22*: 181-182, 2011.

[4] S.D. Pogorilyy, I.V. Bilokon: About computational resources utilization problem for computing cluster with virtual machine nodes. *Problems in programming*, 2-3: 93-101, 2012.

[5] I. Bilokon, S. Pogorilyy: Research of Genetic Algorithm for searching optimal configurations of computing cluster with virtual machine nodes. *Proceedings of the 2nd International Scientific Conference of students and Young Scientists "TAAC"*: 13-18, 2012.

[6] С.Д. Погорілий, І.В. Білокінь, О.В. Вареник: Дослідження віртуальних кластерних архітектур на прикладі алгоритму Белмана-Форда. *Наукові праці ДонНТУ. Серія: "Інформатика, кібернетика та обчислювальна техніка"*, 15(203): 40-47, 2012.

[7] S.D. Pogorilyy, I.V. Bilokon, T.Y. Protsiv: Formation and research parallel schemes of Dantzig algorithm. *Bulletin of Taras Shevchenko National University of Kyiv. Series: "Physics & Mathematics"*, 1: 207-216, 2012.

[8] С. Д. Погорілий, І. В. Білоконь, А. В. Шевченко: Створення та дослідження паралельних схем алгоритму Пріма. *Праці дев'ятої міжнародної конференції "Теоретичні та прикладні аспекти побудови програмних систем" TAAPSD'2012 3-7 грудня 2012 р*: 223-235, 2012.

[9] Microsoft Hyper-V. http://technet. microsoft.com/en-us/library/cc753637

[10] Windows HPC Server 2008 R2 Technical Library. http://technet.microsoft.com/en us/library/ee783547

[11] LINPACK. http://www.netlib.org/benchmark/hpl

[12] Hyper-V Dynamic Memory Overview. http://technet.microsoft.com/en-us/library/hh831766.aspx