Performance Evaluation of Big Data Complex Event Processing on a Virtual Computing System

Mino Ku[1], Duckwon Chung[2] and Dugki Min[a]*
School of Computer Science and Engineering, Konkuk University, Seoul, Korea
{happykus[1], dwchung[2], dkmin[3]*}@konkuk.ac.kr

Abstract. In this paper, we evaluate a complex event processing (CEP) application based on the Esper, which is a CEP platform, on three types of native computing environment and a virtual computing system. In our experiment result, a VM shows near-native processing performance (96.3%) although the running environment of the VM causes the virtualization overhead. Moreover, the processing performance of the VM is approximately 1.28 times and 4.58 times higher than a native dual-core system and a native single-core system

Keywords: Virtualization, Virtual Computing System, Performance Evaluation, Big Data, Complex Event Processing

1 Introduction

Recently, big data is one of hot issues in research fields and industry fields. In big-data computing [1], dealing the large-scale event streams is an important area. In order to deal the event streams in real time or near-real time, complex event processing (CEP) [2] is an emerging solution. Moreover, advances in H/W and S/W technologies, such as multi-core CPU, high speed network and virtualization technologies, give opportunities to handle the big data with less computing resources than before. However, dealing the big data on virtual computing resources, like virtual machine (VM), is not a simple issue. A virtual computing system is based on at least one or more virtualization software, such as Xen [3], KVM [4], VMWare [5], VirtualBox [6] and so on. Since the virtualization software generates the virtualization overhead caused by its architecture, the unconditional employment of the virtualiza-tion software causes unexpected performance degradations. In this paper, we evaluate the processing performance of a CEP application as a big-data application on several types of physical systems and a virtual computing system for analyzing suitability of running a computation-intensive CEP application. The following sections are organized as follows. Section 2 presents experimental environment of our study. Section 3 presents experimental results. Section 4 summarizes and concludes the paper.

2 Experimental Environment

In this research, we implement a simple CEP application which performs gold cross / dead cross (GC/DC) detection [7].The CEP application inputs 5 kinds of real stock data (Symbol and Close Price) of security companies (Mirae Asset, Daewoo, HMC, KTB, NH) listed on the Korea Exchange (KRX). Each stock data consists of 1000 stock data from Jan, 1, 2008 to Dec, 29, 2011. In order to generate 1,000,000 primitive events, we generate the stock data 200 times iteratively to the CEP engine. In the GC/DC detection, we implement an EPL (Event Processing Language) statement, and each test-bed system runs 60 statements based on a following example. Table 1 shows detail H/W and S/W specification of our experiment environment.

```
select short.symbol, short.price, avg(short.price), avg(long.price) from
StockInfo(symbol=COMPANYNAME).win:length(SHORTTERM_PERIOD) as short,
StockInfo(symbol-COMPANYNAME).win:length(SHORTTERMPERIOD) as long
```

Table 1. H/W and S/W specification of test machines

Test Machine	Specification	Common
Single Core (P)	Pentium 4 2.93 GHz, 2 GB RAM	Linux Kernel 3.3.4 Debian Linux (Wheezy)
Dual Core **(P)**	Intel Core2Duo E6750 2.66 GHz, 8 GB RAM	
Hexa Core **(P)**	AMD Phenom II X6 1055T 2.8 GHz, 16 GB RAM Xen Hypervisor 4.1.2 (For running AMD6-VM)	
AMD6-VM	2 GB virtual memory, 2 virtual CPU, PVM, PVOPS-based VM creation	CEP engine (Esper 4.6.0 [8])

3 Experiment Result

Fig. **1** shows a processing time result of our CEP application on four types of physical systems and a VM with 1M primitive events. Fig. 2 shows a throughput comparison result. A hexa-core system records the shortest processing time while single-core system shows the worst processing time. Especially, a test VM (AMD6-VM) shows near-processing time (96.3% processing performance, 265515 ms) compared with the physical hexa-core system (255705 ms). Also, the processing performance of the VM

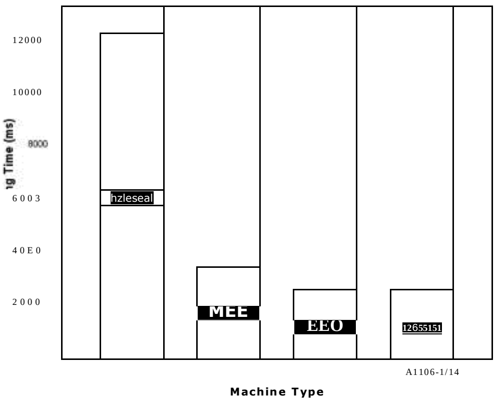

Fig. 1. Comparison of Processing time results on physical systems and a VM

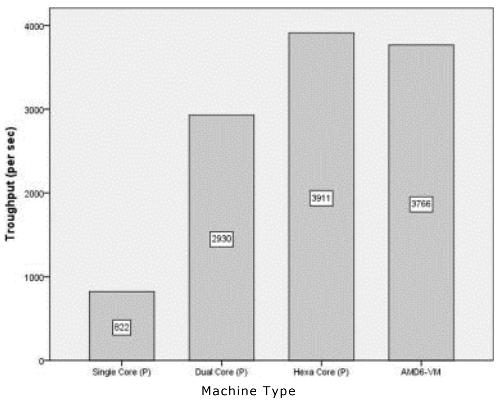

Fig. 2. Comparison of Throughput results on physical systems and a VM

shows approximately 1.28 times higher than the dual-core-based native computing environment. Moreover, the processing performance of the VM shows 4.58 times higher than the single-core-based native computing environment. As we can check in Fig. 1 and Fig. 2, the running a CEP application on a VM shows near-native processing performance compared with its physical system (Hexa Core (P)) although the native-virtualized system allocates a slice of their entire computing resources to the VM. It means that the running of our CEP application and Esper causes an under-utilization situation on a high performance system. Thus, on a high performance, running of multiple VMs for computation-intensive CEP applications can be a considerable method to enhance the utilization of the physical system.

4 Conclusion and Future Work

Now, we are living in the big data world. However, if we don't have techniques and facilities to deal the big data, we lose a change to produce valuable information. In order to handle the big data in an application view point, the CEP is a technology to archive the valuable information production in real-time or near-real time. Furthermore, the advances in H/W and S/W techniques lead to handle the big data with less computing resources than before. In our research, we evaluate the processing performance of a CEP application and a CEP platform on a Xen-based virtual computing system. As a result, the VM shows near-native processing performance (96.3%) although its computing environment involves the virtualization overhead. In future, we will extend our experiments to analyze influences of specifications of VM running environments.

Acknowledgments. This work was supported by the Konkuk University. Dugki Min is a corresponding author to provide; email : dkmin@konkuk.ac.kr

References

1. Randal Bryant, Randy H. Katz and Edward D. Lazowska, Big-Data Computing: Creating Revolutionary Break-throughs in Commerce, Science and Society (2008)
2. Luckham,D, The Power of Events. An Introduction to Complex Event Processing in Distributed Enterprise Sys-tems, Addison-Wesley (2002)
3. Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf eugebauer, Ian Pratt, and An-drew Warfield. Xen and the art of virtualization. In Proceedings of the nineteenth ACM symposium on Operating Systems Principles (SOSP19) (2003)
4. A. Kivity, Y. Kamay, D. Laor, U. Lublin, A. Liguori. kvm: the Linux virtual machine monitor. Proceedings of the 2007 Ottawa Linux Symposium (In OLS 2007) (2007)
5. VMware, http://www.vmware.com/
6. Oracle VirtualBox, https://www.virtualbox.org/
7. T. K. Norio Baba, Toshinori Nomura, Yuuki Sakatani. Utilization of NNs & GAS for Improving the Traditional Technical Analysis in the Financial Market. in SICE Annual Conference Sapporo, (2004) Japan
8. EsperTech Esper, http://esper.codehaus.org/