Advanced Science and Technology Letters
Vol.43 (HCI 2013), pp.1-5
http://dx.doi.org/10.14257/astl.2013.43.01

Abstract Q~ GDWDFHQWHUV~YLUWXDOVHUYHUVDUHFUHDWHGDQGLVRODWHGDWHU DEVWUDFWLQJWKHSK\VLFDOVHUYHUVWRDOORZGHSOR\PHQWRIYLUWXDOL]DWLRQVHUYLFHV~ +RZHYHU~WKHUHDUHPDQ\SUREOHPVRIODFNRIUHVRXUFHVDQGKRZWRUHGXFH GDWHQF\DQGLPSURYHQHWZRUNSHUIRUPDQFHLQWKHFORXG9LUWXDOVHUYHUVUHTXLUH XVHUJURXSVWREHFUHDWHGLQDSSURSULDWHGDWDFHQWHUV~7KHUHIRUH~WKHVHOHFWLRQRI WKHGDWDFHQWHUVWKDWLVFRQYHQLHQWIRUFUHDWLQJDOOWKHQHFHVVDU\YLUWXDOVHUYHUV FRQRWKHUFDVHV~WKHDUUDQJHPHQWRIYLUWXDOVHUYHUVWKDWLWFUHDWLQJGDWDFHQWHUV KDVEHHQDQLPSRUWDQWFKDOOHQJHLQFORXGFRPSXWLQJ~QWKLVSDSHU~ZHZLOOSURYLGHD SROLF\WKDWLVHIIHFWLYHEDVHGRQDQLPSURYHGYHUVLRQRIUHGXFHGDOJRULWKP~WR VFKHGXOHWKHSROLFLHVRIUHVRXUFHVXSSO\IRUYLUWXDOVHUYHUV~+HXVHWKHRSHQ VRXUFHVLPXODWLRQ+DLJDWRVHWXSDQGHYDOXDWHWKHUHVXOWV~RFXVLQJFDLQRQ WKHFUHDWLRQRIUHVRXUFHOHDVHFRQWUDFWDQGSUHVHWFDSDELOLWLHVRI&3Vs~

Keywords: Virtual Server, Distributed, Resource, Virtual Machine, Lease.

1 Introduction

QUHFHQWBHDUV~WKHGHPDQGIRUFRPSXWLQJUHVRXUFHVKDVEHFRPHDNH\ UHTXLUHPHQWLQPDQ\ILHOGVRIVFLHQFHDQGLQGXVWU\~QPDQ\FDVHV~WKHQHHGIRUWKHVH UHVRXUFHVFRQFHQWUDWHVLQDYHU\VKRUWSHULRGRIWLPH~IRUH[DPSOH~DVFLHQWLVWPD\QHHG DFOXVWHUQXPEHURIFRPSXWHUVWRUXQDVLPXODWLRQLQDIHZKRXUVDWDSDUWLFXODUWLPHD WHDFKHUPD\ZDQWWRFUHDWHDFOXVWHURIFRPSXWHUVZLWKYHU\VSHFLILFVRIWZDUH FRQILJXUDWLRQIRUKHUVWXGHQWVWRXVHGXULQJFODVVSUDFWLFHVHVVLRQVWKDWKDSSHQVDW FHUWDLQWLPHVRIWKHZHHNRUDZHEFRPSDQ\QHHGWRXVHDVHUYHUWRKRVWDZHEVLWHEXWFDQQRW HVWLPDWHWKHQHHGHGUHVRXUFHVIRUWKHVHUYHU~DVWKLVZLOOGHSHQGRQWKH DFWXDOWUDIILFWRWKHZHEVLWH~

$WSUHVHQW~UHVHDUFKLQJFRQFXUUHQWVROXWLRQVIRUUHVRXUFHGLVWULEXWLRQ~ZHIRXQGWKDWWKHUH HZDVQRVROXWLRQWKDWFDQGLVWULEXWHUHVRXUFHVWRPHHWPXOWLSOHXVHFDVHVRIGLIIHUHQW XVHUV~$QGWKHSUREOHPRIUHVRXUFHRSWLPL]DWLRQKDVQRWEHHQWKRURXJKO\DGGUHVVHGHLWKHU~ &XUUHQWO\WKHUHLVQRFRPSUHKHQVLYHUHVRXUFHVXSSO\V\VWHPWKDWPHHWVDOOWKH XVHFDVHVRIGLIIHUHQWFRPSXWLQJUHVRXUFHQHHGV~QDGGLWLRQ~HDFKXVHFDVHRIWHQKDV~ LWVRZQPHDVXUHPHQWRIHIILFLHQF\OHYHORIUHVRXUFHDOORFDWLRQ~

ISSN: 2287-1233 ASTL

7KLV SDSHU LV RUJDQL]HG DV IROORZV. 6HFWLRQ GHVFULEHV WKH XQGHUOLQJ FRPSXWDWLRQDO PRGHO DQG SUREOHP GHILQLWLRQ. 6HFWLRQ SUHVHQWV WKH SURSRVHG DOJRULWKP DQG LWV FRUUHFWQHVV SURRI. 6HFWLRQ HQYLURQPHQW DQG QHFHVVDU\ VRIWZDUH GRHV WKH SHUIRUPDQFH DQDO\VLV. 6HFWLRQ WKH UHVXOWV IURP RXU DVVHVVPHQW. 6HFWLRQ FRQFOXGHV.

2 System model and problem definition

,Q WKLV VHFWLRQ ZH ZLOO VXPPDUL]H WKH FRQFHSWV RI GLVWULEXWHG UHVRXUFHV VXSSO\ IRU YLUWXDO VHUYHU V\VWHP >@

2.1 Resource model

7KLV ZRUN IRFXVHV RQ OHDVLQJ FRPSXWDWLRQDO UHVRXUFHV IURP D ZLWK D VLQJOH DGPLQLVWUDWLYH GRPDLQ RU VLWH. 7KH VHW RI OHDVDEOH FRPSXWHUV RU QRGHV LQ D VLWH LV GHQRWHG 3. (DFK QRGH [3 KDV OHDVDEOH UHVRXUFHV DQG DWWULEXWHV. /HDVDEOH UHVRXUFHV ZLWKLQ D QRGH FDQ EH DOORFDWHG WR RQH RU PRUH OHDVHV, XS WR D PD[LPXP FDSDFLW\, DQG PD\ LQFOXGH SURFHVVRUV, PHPRU\, GLVN VSDFH, QHWZRUN EDQGZLGWK HWF.

7KH VHW RI WKH W\SHV RI OHDVDEOH UHVRXUFHV LQ D VLWH LV GHQRWHG G_R H.J. G_R ^ PHPRU\, GLVN «`. 7KH VHW RI W\SHV RI DWWULEXWHV LQ D VLWH LV GHQRWHG G_A

H.J.[P] G_A ^DUFK, YPP «`.

6RPH GLVFXVVLRQV LQ WKH IROORZLQJ VHFWLRQ ZLOO DVVXPH D VLWH ZLWK KRPRJHQHRXV UHVRXUFHV, VXFK WKDW HYHU\ QRGH KDV S SURFHVVRUV, P 0% RI PHPRU\, G RI GLVN VSDFH, EL 0%/V RI LQFRPLQJ EDQGZLGWK, DQG ER 0%/V RI RXWJRLQJ EDQGZLGWK. 171717111117KHMVWHBASHILQHGNIROORZV G_R NSURFEPHPNGLVN114HWLQ QHWRXWE

2.2 Lease

$ OHDVH LV D QHJRWLDWHG DQG UHQHJRWLDEOH DJUHHPHQW EHWZHHQ D UHVRXUFH SURYLGHU DQG D UHVRXUFH FRQVXPHU, ZKHUH WKH IRUPHU DJUHHV WR PDNH D VHW RI UHVRXUFHV DYDLODEOH WR WKH ODWWHU, EDVHG RQ D VHW RI OHDVH WHUPV SUHVHQWHG E\ WKH UHVRXUFH FRQVXPHU.

7KH OHDVH WHUPV HQFRPSDVV WKH KDUGZDUH UHVRXUFHV UHTXLUHG E\ WKH UHVRXUFH FRQVXPHU, VXFK DV &38V, PHPRU\, DQG QHWZRUN EDQGZLGWK, D VRIWZDUH HQYLURQPHQW UHTXLUHG RQ WKH OHDVHG UHVRXUFHV, DQG DQ DYDLODELOLW\ SHULRG GXULQJ ZKLFK WKH UHTXHVW KDUGZDUH DQG VRIWZDUH UHVRXUFHV PXVW EH DYDLODEOH.

3 The proposed algorithm for distributing virtual servers

99 GLVWULEXWLRQ RQ SK\VLFDO QRGHV DW D VSHFLILF WLPH. 7R GHWHUPLQH WKH GLVWULEXWLRQ FDSDELOLWLHV RI DOO 99 V RI D OHDVH RQ SK\VLFDO QRGHV DW UHTXLUHG WLPHV, VWDUWLQJ DW WLPH W DQG ODVWLQJ G VHFRQGV, WKH ILUVW DOJRULWKP ZLOO DUUDQJH WKH SK\VLFDO QRGHV DFFRUGLQJ WR WKH IROORZLQJ FULWHULD.

6WHS)LUVW, SULRULWL]H WKH SK\VLFDO QRGHV WKDW KDV WKH OHDVW OHDVH UXQQLQJ DW WLPH W.

Step 2: If the criteria of (1) are equal for some nodes, compare the number of free resources at time t of these nodes.

Step 3: Finally, prioritize the nodes with most stable free resources.

```
Algorithm 1 Greedy mapping without preemption
Input: A lease l, time t, a duration d
Output: A mapping, if one can be found, nodes[l] → P
starting at time t+d.
map ← empty dictionary
P' ←sort(P) {Sorted based on aforementioned
method} cur_node ← First Node in nodes[l]

For ∀ ∈P' do
p_done ← false
While not p_done do
    if cur_node can distribute to p from t to t+d then
         mapcur_node ← p
        cur_node ← next Node in nodes[l]
        else
                p_done ← false
         end if
end while.
end for



if ∀   ∈ nodes[l] is distributed then
    return map
else



    return ∅
end if
    End.
```

When the nodes has been sorted, the model uses a greedy algorithm to distribute all VM's.

Aforementioned algorithm has the ability to distribute multiple VM's on the same node. With this research aiming to provide efficient distribution of resources, we propose next technique in distributed environments. In this case, the algorithm tries to distribute as many VM's as possible on multiple physical nodes.

Algorithm 2 proposed for distributed environments

When a lease (l) requests resources to create a virtual machine VM (including software, data operating systems, etc.) of any Data Center DC_i.

1. *If DC_i already has VM then l_k already has resources, no deadlock detects and algorithm ends.*
2. *Else, if DC_i does not have VM but l has been issued for transaction l_j then send message l_j block l_k for $DC(l_j)$ and $DC(l_k)$. The message content is (l_j, l_k).*

When any DC_i received a notification message for blocked pair (l_j, l_k) then:

3. *If DC=DC(l_k) then add l_j to set P(DC) if l_j does not belong to T(DC).*
4. *If P(DC) fl T(DC) = {$_j$} then deadlock detection succeeds and algorithm ends..*
5. *Else, send message (l_j, l') for all servers DC(l'), with each l' being a member of set B(S).*
6. *Else if DC≠DC(l_k) then add l_k to T(DC).*

From the above algorithm, it can be concluded that the proposed solution computational complexity is O(N), with N being the total number of sites in the system.

4 The results from our assessment

This section presents the results to the simulation experiments on simulated scheduling software Haizea.

Table 1. showing the most effective time to complete a contract when combined with deadlock detection algorithms

The ability of the CPU p	10%	20%	30%	40%	50%
NOVM with NO suspension/recovery	0,15%	0,37%	10,71%	0,15%	0,37%
NOVM with suspension/recovery	0,46%	1,26%	6,09%	0,46%	11,26%
Can create VM	75,46%	77,40%	85,99%	90%	97,40%

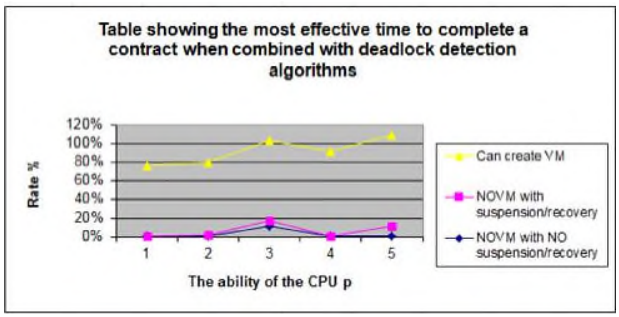

Fig. 1. Graph showing lease contract completion time for each CPU capability in distributed environments, using deadlock detection algorithm.

Through chart 1, by applying scheduling using greedy algorithm combined with deadlock detection algorithm to the requirement to provide resources for 10 lease contracts (given the condition that the ability of CPU is pre-determined), we found that the success rate to create VM is very high with the CPU's ability at 50%. As for

CPU's ability at 10% and 20%, the success creation is also higher than that of failed creation.

5 Conclusion

In the context of this paper, we are interested primarily in the criteria of readiness, because it affects the most to preparing costs. The use of virtualization technology has great potential to meet the requirements of complex computing systems.

Two algorithms proposed in this research on providing efficient resources for virtual workspaces can grow up by utilizing the above advantages. Security problems, isolation, and the ability to adjust resources can impact positively on the standard of environmental quality by ensuring sufficient workspace resources (CPU, RAM, etc.) to support execution. Independence ability also improves the standard of resources openness, expanding pool of physical resources to run certain workspaces.

6 References

1. B.Sotomayor, R.Santiago Montero, I.Martín Llorente, I.Foster. Virtual Infrastructure Management in Private and Hybrid Clouds, IEEE Internet Computing, vol. 13, no. 5, pp. 14-22, Sep./Oct. 2009.
2. B.Sotomayor, R.Santiago Montero, I.Martín Llorente, I.Foster. Resource Leasing and the Art of Suspending Virtual Machines, The 11th IEEE International Conference on High Performance Computing and Communications (HPCC-09), June 25-27, 2009, Seoul, Korea.
3. B.Sotomayor, R.Santiago Montero, I.Martín Llorente, I.Foster. Capacity Leasing in Cloud Systems using the OpenNebula Engine, Workshop on Cloud Computing and its Applications 2008 (CCA08), October 22-23, 2008, Chicago, Illinois, USA.
4. Borja Sotomayor, Kate Keahey, Ian Foster. Combining Batch Execution and Leasing Using Virtual Machines, ACM/IEEE International Symposium on High Performance Distributed Computing 2008 (HPDC 2008), Boston, MA. June 2008.
5. Borja Sotomayor, Kate Keahey, Ian Foster, Tim Freeman. Enabling Cost-Effective Resource Leases with Virtual Machines, Hot Topics session in ACM/IEEE International Symposium on High Performance Distributed Computing 2007 (HPDC 2007), Monterey Bay, CA (USA), June 27-29, 2007.
6. D. P. Mitchell and M. J. Merritt, "A distributed algorithm for deadlock detection and resolution," in Proc.ACM Symposium on Principles of Distributed Computing, pp. 282-284,1984.

