Energy efficient resource allocation strategy for cloud data centres

Dang Minh Quan[1], Robert Basmadjian[2], Hermann De Meer[2], Ricardo Lent[3], Toktam Mahmoodi[3], Domenico Sannelli[4], Federico Mezza[4], Corenten Dupont[1]

[1] CreateNet, Italy, {quan.dang,cdupont}@create-net.org

[2] University of Passau, Germany, {Robert.Basmadjian,hermann.demeer}@uni-passau.de

[3] Imperial College, UK, {r.lent,t.mahmoodi}@imperial.ac.uk

[4] ENI, Italy, {domenico.sannelli,Federico.Mezza}@eni.com

Abstract: Cloud computing data centres are emerging as new candidates for replacing traditional data centres. Cloud data centres are growing rapidly in both number and capacity to meet the increasing demands for highly-responsive computing and massive storage. Making the data centre more energy efficient is a necessary task. In this paper, we focus on the organisation's internal Infrastructure as a Service (IaaS) data centre type. An internal IaaS cloud data centre has many distinguished features with heterogeneous hardware, single application, stable load distribution, lived load migration and highly automated administration. This paper will propose a way of saving energy for IaaS cloud data centre considering all stated constraints. The basic idea is rearranging the allocation in a way that saving energy. The simulation results show the efficiency of the method.

Keywords: energy efficient, resource allocation algorithm, cloud data centre

1. Introduction

An IaaS cloud data centre is used to host computer systems and associated components. Cloud computing data centres are emerging as new candidates for replacing traditional data centres that are growing rapidly in both number and capacity to meet the increasing demands for computing resources and storages. Clearly this expansion results in a significant increase in the energy consumption of those data centres. According to [1], data centres in USA consumed about 61 BkWh and accounted for 1.5 % of total US electricity consumption in 2006.

The reported work in this paper is shaped based on the research collaboration within the Fit4Green project on energy efficiency in data centres, where the ENI's IaaS data centre is used for the investigation purposes. An internal IaaS data centre like the ENI's data centre has the following characteristics: Mixed hardware environment with different technologies, Single application with stable load, Live load migration capability, Highly automated administration.

To this end, we propose an energy efficient resource allocation approach that utilises the capability of VMs live migration to reallocate resources dynamically. The basic idea is to use a heuristic that is consolidating and rearranging the allocation in an energy efficient manner.

This paper is organised as follows. Section 2 describes the related works. After analysing the energy consumption model of different entities in the data centre in Section 3, the energy efficient resource allocation and performance investigation are discussed in Section 4. Finally, Section 5 concludes the paper with a short summary.

2. Related work

Currently, resource allocation mechanisms which are widely used in data centres include load balancing, round robin and greedy algorithms. The load balancing module tries to distribute the workload evenly over the computing nodes of the system. In the round robin algorithm, the servers are in a circular list. There is a pointer pointing to the last server that received the previous request. The system sends a new resource allocation to the next node with enough physical resources to handle the request. Having the same set up as the round robin algorithm, the greedy algorithm continues to send new resource allocation request to the same node until it is no longer has enough physical resources, then the system goes to the next one.

The work in [2] studies general energy saving approaches. Saving energy in ICT divides into two main fields, saving energy for network [3,4] and saving energy for computing nodes [5,6]. Related to saving energy in cloud data centre, the work in [7] adjusts the working mode of the server according to load to save energy. In [8], the authors focus on saving energy for PaaS (Platform as a Service) cloud data centre. Our work is different from previous in two main ways. We focus on IaaS scenario and use moving workload as the main method. [9] is the most closely work with our work. However, in [9] the authors use the predefined MIPS to present the capacity and the load which is quite difficult to the user. We use the measured load value instead.

3. Energy consumption model

Server power consumption

Processor

Based on our observations of a custom benchmark that puts on each core variations of load ranging from 10-100%, we noticed that the power consumption of individual core of a processor increases linearly with its utilisation, and is given by:

$$P_{Core} = P_{\max} * \frac{L_{Core}}{100}, \quad (1)$$

where L_{Core} is the utilization, the workload, of the corresponding core and $P_{\max}$ denotes the maximum power of the processor. Then, the power consumption of multi-core processors is:

$$P_{CPU} = P_{idle} + \sum_{i=1}^{n} P_{Core_i}, \quad (2)$$

where P_{idle} is constant and denotes the power of the processor in the idle state.

Memory

Given an unbuffered SDRAM of type DDR3, then its power consumption is:

$$P_{RAM} = P_{RAM\_idle} + \delta * \beta, \quad (3)$$

where $\delta = 1.3$, $\beta = 7.347$, and P_{RAM_idle} is the idle power consumption given by:

$$P_{RAM\_idle} = \sum_{i=1}^{n} s_i * p, \quad (3)$$

such that *i* denotes the number of installed memory modules of size *s*, whereas

$$p = \frac{f}{1000} + \alpha \sqrt{f}\,(f_c - f),$$

where f_c = 1600, f denotes the input frequency, and α =0.000026.

Hard Disk

We noticed that the startup and accessing mode power consumptions are in average respectively 3.7 and 1.4 times more than that of the idle (P_{idle}):

$$P_{HDD} = a * 1.4 * P_{idle} + b * P_{idle} + c * 3.7 * P_{idle} , \quad (4)$$

where $a, b, c \in [0,1]$ denote respectively the probability of accessing, idle and startup.

Mainboard

The power consumption of the mainboard is given by the following equation:

$$P_{Mainboard} = \sum_{i=1}^{l} P_{CPU} + P_{RAM} + \sum_{j=1}^{m} P_{NIC} + \sum_{k=1}^{n} P_{HDD} + t, \quad (5)$$

where t is a constant having a value of 55.

Server Power

Given a server composed of several components, then its power consumption is:

$$P_{Server} = \sum_{i=1}^{l} P_{Mainboard} + \sum_{j=1}^{m} P_{Fan} + \sum_{k=1}^{n} P_{PSU} ,$$

such that j and k denote respectively the number of fans and power supply units whose power consumption models can be found in [10].

Network power consumption

Given that a data centre consists of N stations (servers) that connected through S network switches, which all have the same number of ports n. Switches are interconnected forming a mesh topology, and each computing device i can generate and consume a total traffic of λ_1 packets per second. Let us define T as the total network traffic (in packets) generated or consumed by all stations. The power consumption of all network switches, P_S can be estimated from the expression,

$$P_S = \sum_{j=0}^{S-1} \Phi_j \left(\alpha T + (1 - \alpha) \sum_{i=jn}^{jn+n+1} \lambda_i + \Gamma_j \right), \quad (10)$$

where α is the fraction of inter-switch traffic. Γj represents external traffic demand entering and leaving the switch. A linear approximation of Φ can be utilised basing on the maximum (nominal) power consumption, ρ, and the switch bandwidth, Λ that can be obtained from the specification sheets of the device.

$$\Phi^*(\lambda) = \rho \left(\alpha + \frac{\beta - \alpha}{\Lambda} \lambda \right) \quad (11)$$

Data centre power consumption

Let E(i) be the function calculating energy consumption for node i, and w_i denotes the number of edges at node i. Then the overall energy consumption of a data centre is given by the following equation:

$$E_{tot} = \sum_{i=1}^{n} \left(\sum_{j=0}^{wi} E(c_{ij}) \right) + E(i), \quad (12)$$

4. Optimisation algorithms

Problem statement

Assume that we have a set of servers S. Each server $s_i \in S$ is characterised with number of cores, amount of memory and amount of storage ($s_i.nr\_Core$, $s_i.nr\_RAM$, $s_i.nr\_Stor$).

Each server s_i has a set of running virtual machine V_i including k_i virtual machines. Each virtual machine $v_j \in V_i$ is characterised with required number of virtual CPU, amount of memory, amount of storage ($v_j.r\_vCPU$, $v_j.r\_RAM$, $v_j.r\_Stor$) and the average CPU usage rate computed in %, amount of memory, amount of storage ($v_j.a\_Urate$, $v_j.\ a\_RAM$, $v_j.\ a\_Stor$). Because the load in each VM is quite stable, we assume that those values do not change through time.

When there is a new resource allocation request, the new VM must be allocated to a server in a way that total usage of VMs does not exceed the capacity of the server and the added energy usage is minimum.

For the global optimisation request, the VMs must be arranged in a way that total usage of VMs does not exceed the capacity of the server and the energy usage is minimum.

F4G-CS (Traditional single algorithm)

When a new VM comes, the F4G-CS algorithm will check all computing nodes to find the suitable node using the least amount of energy.

Step 0: Determine all servers meet the constraint and store in array A

Step 1: If the array A is empty, stop the algorithm and inform no solution

Step 2: Set index i at the beginning of the array A

Step 3: Calculate energy consumption E_i of the data centre if the VM is deployed on the server at array index i

Step 4: Store (i, E_i) in a list L

Step 5: Increase i to the next index of A

Step 6: Repeat from Step 3 to Step 5 until i goes out of the scope of A

Step 7: Determine min E_i in the list L

Step 8: Assign the VM to the server at index i_{min}

Step 9: If server at index i_{min} is OFF, put action turn ON to action list

It is noted that E_i is calculated for servers having workload. The server without workload will be shutdown.

F4G-CG (Cloud global optimisation) algorithm

The F4G-CG algorithm has two main phases. In the first phase, we will move the VMs from low load servers to higher load servers if possible in order to free the low load server. The free low load server can be turned off. As the increasing energy by increasing the workload is much smaller than the energy consumed by the low load servers, we can save energy. The algorithm for phase 1 is as following:

Step0: Forming the list LS of running servers

Step1: Find the server having the smallest load rate.The load rate of a server is defined as the maximum (CPU load rate, Memory load rate, Storage load rate)

Step2: Sort the VMs in the low load server according to the load level in descending order list LW

Step3: Remove the low load server out of running server list

Step4: Use F4G-CS algorithm to find the suitable server in LS for the first VM in the list LW

Step5: If F4G-CS finds out the suitable server, update the load of that server, remove the VM out of LW

Step6: Repeat from Step4 to Step5 until LW is empty of F4G-CS cannot find out the suitable server

Step7: If F4G-CS cannot find out the suitable server, reset the state of found servers and mark Stop=true

Step8: Repeat from Step 1 to Step7 until Stop=true

In the second phase, we will move the VMs from the old servers to the modern servers. The free old servers can be turned off. As one modern server can handle the workload of many old servers, the energy consumed by the modern server is smaller than the total energy consumed by those many old servers. The algorithm is:

Step0: Sort free servers into a descending order list LFS according to resource level.The resource level of a server is defined as the minimum(s_i*.nr_Core/max_Core*, s_i*.nr_RAM/max_RAM*, s_i*.nr_Stor/max_Stor*) with *max_Core*, *max_RAM*, *max_Stor* are the maximum number of Cores, memories, storages of the server in the pool.

Step1: Sort running servers into an ascending order list LRW according to load rate.The load rate of a server is defined as the maximum(CPU load rate, Memory load rate, Storage load rate)

Step2: Set m_count=0

Step3: Remove the first server *s* out of LFS

Step4: Remove the first server *w* out of LRW

Step5: If we can move workload from *w* to *s* m_count+=1

Step6: Repeat from Step4 to Step5 until we cannot move workload from *w* to *s*

Step7: If m_count <= 1, reset the state of moved *w,s* and mark Stop=true

Step8: Repeat from Step2 to Step7 until Stop=true

Simulation result

The simulation is done to study the saving rate of the resource allocation mechanism in different resource configuration scenarios. To do the simulation, we use 4 server classes correlated to single core, duel cores, quad cores and six cores. We generated 3 resource scenarios: modern data centre, normal data centre and old data centre. In the modern data centre, the percentage of many cores server is dominated. In the old data centre, the percentage of small number of cores server is dominated. With each resource scenario, we generated a raw set of jobs. Those jobs randomly come to the system within the period of 1000 time slots. We select the runtime for each job from 1 to 100 time slots. To perform simulation for global optimisation request, with each resource and workload scenario, we still use linear resource allocation algorithms to allocate the new coming job to the resource. However, every 5 time slots, we execute the F4G-CG to perform global optimisation. The simulation result is in Table 1.

Scenario	**1**	**2**	**3**
Round robin(MWt)	61.23	48.45	41.89
Round robin + F4G-CG (MWt)	47.64	40.83	37.21
Greedy (MWt)	50.74	40.35	35.78
Greedy + F4G-CG (MWt)	45.67	37.78	33.47
Load balance (MWt)	60.46	47.87	41.15
Load balance + F4G-CG (MWt)	46.56	39.67	36.85
F4G-CS (MWt)	46.58	36.13	32.18

F4G-CS + F4G-CG (MWt)	44.76	34.47	30.7

Table 1: F4G-CS/F4G-CG simulation result

The simulation result shows the efficiency of the energy aware global optimisation algorithm. It can significantly reduce the energy consumption of a data centre.

6. Conclusion

This paper has presented a method that potentially reduces the energy consumption of the internal IaaS data centre. To save energy, we rearrange the resources allocation by the workload consolidation and frequency adjustment. In the reallocation algorithm, we take advantage of the fact that new generation computer components have higher performance and consume less energy than the old generation. Thus, we use the heuristic that move the heavy load applications to the new servers with larger number of cores while moving light load applications to the old servers with smaller number of cores, and thus switch off the old servers as many number as possible. The investigations show that our presented algorithm can enhance the performance further when data centre consists of larger number of old server, and also in the case when many old servers are working with heavy load rate and many modern servers are working with light load rate.

References

1. U.S. Environmental Protection Agency. Report to congress on server and data center energy efficiency. Technical report, ENERGY STAR Program, Aug. 2007.
2. A. Berl, E. Gelenbe, M. di Girolamo, G. Giuliani, H. de Meer, M.-Q. Dang, and K. Pentikousis. Energy-Efficient Cloud Computing. The Computer Journal, 53(7), September 2010, doi:10.1093/comjnl/bxp080.
3. E. Gelenbe and C. Morfopoulou, A Framework for Energy Aware Routing in Packet Networks. accepted for publication in The Computer Journal.
4. E. Gelenbe and T. Mahmoodi, Energy-Aware Routing in the Cognitive Packet Network. In International Conference on Smart Grids, Proceeding of Energy 2011 conference, 2011.
5. T. Heath, B. Diniz, E. V. Carrera, W. M. Jr., and R. Bianchini. Energy conservation in heterogeneous server clusters. Proc. of the tenth ACM SIGPLAN symposium on Principles and practice of parallel programming - PPoPP'05, 2005, pp. pages 186-195.
6. C. Xian , Y. H. Lu, Dynamic voltage scaling for multitasking real-time systems with uncertain execution time, Proc. of the 16th ACM Great Lakes symposium on VLSI, 2006
7. T. V. Do ,Comparison of Allocation Schemes for Virtual Machines in Energy-Aware Server Farms, The Computer Journal, 2011.
8. J. Leverich and C. Kozyrakis, On the Energy (In)efciency of Hadoop Clusters, SIGOPS, 2010
9. A.Beloglazov and R. Buyya, Energy Efficient Resource Management in Virtualized Cloud Data Centres, Proceeding of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, 2010.
10. R. Basmadjian, N. Ali, F. Niedermeier, H. d. Meer and G. Giuliani, A Methodology to Predict the Power Consumption for Data Centres, Proceedings of e-Energy 2011, 2011.