ISSN:2321-1156

International Journal of Innovative Research in Technology & Science(IJIRTS)

Mathematical Model – Task Execution and Energy Consumption in Diverse Cloud Computing Environment

R.T.Kaple, Affiliation BNCOE, SGBAU Amravati. Dr.V.R.Chirade, Affiliation GSG Umarkhed, SGBAU Amravati

Abstract

Cloud computing is transforming the entire IT industry, into a high-performance computing, and personal data sharing and management. In cloud computing, computing power is supplied as a utility, similar to electricity or water. As such, service providers can centrally manage, maintain, and upgrade computing resources, discharging the burden from small business owners or those who do not have the expertise or budget to handle the fast-changing computing infrastructure.

Introduction

Using the cloud for High Performance Computing (HPC) can substantially reduce the total cost of ownership by eliminating the need to maintain large-scale parallel machines and their energy-consuming power and cooling systems.[1,2] From a cost efficient perspective, there are adjustments in terms of resource provisioning. An HPC job, which can be perfectly parallelized, takes eight hours to complete using one computing node. If the cloud computing service provider charges for a job on a per-machine per-hour basis (that is, based on the accumulated machine time), instead of running it on one node for four hours, the job can be finished in one hour on four machines with four times speedup with the same utility charge i.e. four machine hours.

One trend that complicates this adjustment is the diversity in a cloud computing environment. Although a cloud service provider can start with nearly alike computing nodes, the facility will likely grow more diverse over time due to upgrades and replacement. Therefore, not only will each computing node's performance and capability continue to deviate, the new computing nodes will also provide better performance for the same amount of power due to technology scaling and architectural innovation. Because of this diversity, response times will vary significantly depending on provisioning policies. To alleviate (minimize) this variation and guarantee quality of service, the cloud provider might want to remove the slowest computing nodes. The question is how slow a physical node can be for a given task to maintain its optimal computing quality in terms of execution time and energy cost.

To tackle this issue, we established a mathematical moel based on statistics for a diverse cloud environment. To understand optimal provisioning in a cloud, we used this model to evaluate the adjustment of a task's execution time and energy consumption.

Cloud Computing Model

(Energy consumption w.r. to task execution time)

For this study, we assume the workload is perfectly parallelizable, which is often the case for throughput oriented computing in HPC and transactional processing applications. For example, the most common cloud computing application is file transferring on the Web. Servers in the cloud; can process all the requests received by a Web service at the same time individually and independently. Therefore, the cloud can achieve **n** times faster when **n** nodes are deployed if and only if the number of concurrent users is always larger than or equal to **n**. Next, we assume that an entire workload can be evenly divided into **m** smaller job units without affecting the workload's scalability. We also assume that **m** is larger than **n**, where **n** represents the maximum number of virtual machines in the cloud. (For simplicity, **m = kn**, where **k** is a positive integer.) In this study, one job unit represents the smallest task running to the end on a single physical node without interruption. However, we do not consider intermittent context switches within one job unit as interruption as long as the task keeps running on the same physical node. In addition, we do not allow a virtual machine to be migrated among physical nodes during a job unit's execution because this migration will not only include the executable image but also all the architectural states, including the memory footprint. Data migration on interconnected cloud computing nodes would likely cause significant performance degradation due to peer-to-peer communication.

Cloud Power and Performance Behaviour

Before going to power and performance in a diverse cloud, we present a scenario from a cloud administrator's perspective.

Typically, cloud service providers begin their cloud computing business with several nearly alike computing nodes. Over time, the cloud provider will replace some of the old computing nodes with newer nodes featuring the latest technologies. Gradually, the capability and performance of all machines in the cloud will become more dissimilar. Although previous studies considered diversity at the microarchitectural [3] and system levels, [4] they all assumed diversity in the same generation of manufacturing technology. We consider computing diversity in a broader sense.

We reviewed the power and performance trends of commercial microprocessors over the past few years and used our observations to justify our model assumption. We first plotted the thermal design power (TDP) numbers and the PassMark performance scores[5] for several processors under 65 W, including Pentium, Core 2, Core i3/i5/i7, and Xeon. This included all commercial desktop and server processors from Intel from January 2006 to February 2011, except Celeron processors and certain processors that did not report TDP or PassMark results. The solid line in Fig. 1. shows their asymptotic power consumption and performance trend between 2006 and 2011. The dashed lines without individual dots show the trends of two other machine groups based on their TDP: 70 W to 120 W and more than 120 W. We applied regression methods to estimate the relationship between power and performance over time. Taking all the samples into account, we plotted our regression models for power and performance (solid lines).

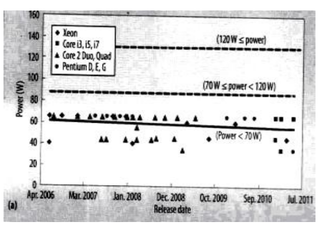

Figure1(a) Power Consumption

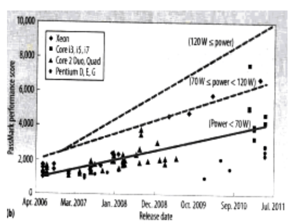

Figure 1(b) Performance

As Figure 1(b) shows, the performance continues to improve for each machine group across different generations. On the other hand, the TDP trend in Figure 1(a) shows negligible growth. More interestingly, the TDP trends for the two lower-power machine classes are decreasing. This decrease is the consequence of a recent awareness of the power wall, which gradually increases the heat dissipation cost. For the same reason, we anticipate that the power grade of future processors will remain below the bar. This also implies that with the same power budget, newer machines can deliver higher performance. In other words, performance per power (a metric derived by dividing the performance score by the power consumption) continues to grow over time. For example, the 95 W Core i7 (Lynnfield), released in September 2009, achieves higher performance than the 95-W Pentium D (Presler), released in January 2006. This difference is largely attributable to technological advances in micro architecture as well as scaled-down feature size and supply voltage. Given these observations, we define our model of power and performance for a future diverse cloud, based on two assumptions.

First, the computing nodes in the cloud we analyze are dissimilar having different micro architectures fabricated using different processes. Thus, the cloud provides varied capability and process technologies.

Second, the performance capabilities of these computing nodes are uniformly distributed (from low to high) but consume exactly the same amount of power. The rationale behind this second assumption is two- fold.

First, for a given power budget, the performance of each machine class continues to improve linearly while their power envelope remains almost unchanged. In other words, the power efficiency measured by performance per power improves over time.

Second, when a data centre phases out some computing nodes due to an upgrade, it can safely deploy new computing nodes only when these upgrades aggregated power consumption does not exceed the original. Otherwise, the data centre must also upgrade its power delivery infrastructure as well as its cooling capacity to accommodate the new servers.

Given this overhead, we expect that the replacement and upgrade will be done without altering the power delivery infrastructure. Therefore, we assume that the newly deployed servers will improve performance linearly across different machine configurations while using the same amount of power.

To express this distribution mathematically, we assume that the response time for executing a job unit in such a cloud is uniformly distributed from **a** seconds (the fastest node) to **b** seconds (the slowest node). Figure 2 shows the probability distribution function (PDF) of the response time for executing a job unit in this cloud.

On the other hand, we assume that the cloud service provider can improve the worst-case response time by terminating physical nodes with the least performance. For example, when a cloud service provider decides to remove one-third of its physical nodes from the slowest batch, we assume that the new response time for executing a job unit of this cloud becomes a uniform distribution from **a** seconds to (a+ 2b)/3 seconds, represented by **U**(a,(a+2b)/3) [17]. As such, we assume that the maximum number of virtual machines that can be allocated on this cloud also reduces in the same ratio.

Figure 3 shows the impact of removing one-third of nodes from the cloud. The variable **p** in this figure represents the maximum number of virtual machines that can be allocated on the cloud, while **n** represents the maximum number of virtual machines for the original cloud as shown in Figure 2. Moreover, the PDF in Figure 3 shows the improved worst-case response time as a result of removing one-third of the physical nodes from the slowest side.

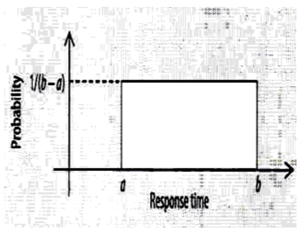

Figure 2

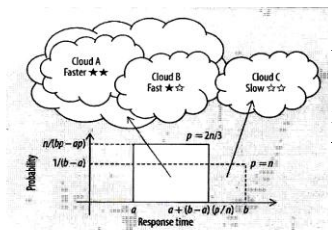

Figure 3

Although dispatching more jobs to newly deployed servers with higher power efficiency increases energy efficiency, this is not the case for a data centre, for two reasons. First, for a data centre, it is important to balance the power draw across the AC phases. [6] The balance will break when jobs are distributed to only certain computing racks. Second, we want to minimize the number of hotspots for a data centre, a common consequence of unbalanced work- loads. Hotspots generally cause higher machine failure rates and require additional attention and effort to remove the heat.

Execution Time and Energy Consumption

We define the execution time of a given workload on a cloud as the time required to finish a workload consisting of **m** job units. When some job units are assigned to more than one virtual machine, the execution time, in our definition, is bounded by the virtual machine that finishes last. For example, when an animator renders a movie comprising **m** independent frames, the movie cannot be released before the last frame finishes rendering. In addition, when comparing the performance of cloud configurations, we use as the baseline the case of executing the same amount of workload on a virtual machine running on the fastest node. When we use more virtual machines to execute the workload in parallel, we use slower nodes to accomplish the task. As a result, the parallelized version could reduce the overall effectiveness of energy consumed in the cloud.

Energy consumption is the total energy needed to complete a given workload. In particular, when some physical nodes finish their assigned job units before the others, we assume that these nodes will not consume energy while waiting for the others to finish. This is because, in a real world scenario, these nodes will either be assigned to other tasks or moved to a near-zero power state to save energy.[7] In addition, given that each computing node consumes the same amount of power, energy consumption as defined will be proportional to the total execution time. Therefore, we calculate a parallelized workload's utility consumption as the summation of each virtual machine's execution time. To quantify the effectiveness of resource provisioning in a cloud, we use the energy-delay product (EDP), [8] which we calculate by multiplying the execution time (seconds) with the energy consumption (joules). We will use this metric in our subsequent evaluation when provisioning resources (that is, the number of virtual machines to allocate to achieve optimal energy efficiency).

Analytical Evaluation

We use analytical models, based on our assumptions, to compare each configurations EDP to the baseline EDP.

A. Assumptions :

The baseline of our study assumes that the entire job is performed on one virtual machine running on the fastest physical node. In this case, the fastest physical node can retire a job unit every **a** seconds. Because there are **m** independent job units in the entire workload, the baseline configuration takes **ma** seconds to finish. This configuration consumes W× ma joules for completing the entire workload, where W represents a physical node's power. Thus, the EDP of this study's baseline is $\mathbf{EDP_{base} = (W \times ma)(ma) = Wm^2a^2}$.

B. Expectation-Based Analysis :

We use an expectation-based analysis to determine a cloud model's execution time and energy consumption. We use a new distribution function to represent the execution time of a virtual machine with more than one job unit. Execution time distribution across virtual machines. The PDF of the response time when using p virtual machines is given by **U**(a,(a+((b – a)p)/n)), as Figure (3) illustrates. However, when a virtual machine is responssible for more than one job unit (that is, **m/p** units), the virtual machine's total execution time cannot be modeled the same way. Rather, we model it as the summation of independently selected **m/p** samples from Figure (3). When we add independent samples from a uniform distribution, the summation's distribution function tends to approach a normal distribution according to the central limit theorem.[9] This theorem proves that when we add more independent samples into the summation, the summation's distribution will become more like a normal distribution. In addition, the summation of 12 samples is known to be good enoughto satisfy the central limit theorem. [9] In this case, we assume that a virtual machine is responsible for more than 12 job units by letting $m \geq 12n$ (that is, m

$\geq 12p$ because $p \leq n$). Now our goal is to obtain the mean and variance of the normal distribution representing the total execution time of a virtual machine responsible for m/p job units. First, we calculate the mean and variance for the original uniform distribution, U(a,(a + ((b – a)p)/n)).

Mean $= \frac{1}{2}\{ a + a + [(b-a)p/n] \} = a + \frac{(b-a)p}{2n}$ and

Variance $= \frac{1}{12}\{a+[(b-a)p/n] - a \}^2 = \frac{1}{12}\{\frac{(b-a)p}{\sqrt{12}} x \frac{1}{n}\}^2$

The central limit theorem shows that the summation of **m/p** independent samples from this distribution will become a normal distribution with the following mean and variance.

$$N\{\frac{m}{p}(a+\frac{(b-a)p}{2n}),(\sqrt{\frac{m}{p}}\frac{(b-a)p}{\sqrt{12}n})^2\} = N(\mu , \sigma^2) \text{ ----------- } (1)$$

For convenience, we use **μ** and $\boldsymbol{\sigma^2}$ to denote the distribution's mean and variance. All in all, when using p virtual machines, each machine's execution time will follow the normal distribution, **N**(μ, σ^2). The ultimate question is, How many seconds of time will it take to finish the entire workload ? To answer this question, first of all we must determine the expectation of the largest sample from **N**(μ, σ^2) when we pick **p** samples. Because the overall execution time depends on the slowest virtual machine that finishes last, the largest of **p** samples will give the total execution time.

C. Largest Sample

Before finding the largest sample's expectation, we discuss the same question for the standard normal distribution, **N**(0,1). Let pdf(x) be the PDF of the standard normal distribution. In this PDF, let **y** be the largest sample among randomly chosen **p** samples. For each case out of **p** cases, the probability of **y** being the largest sample is given as follows.

$$\text{Probability} = pdf(y)\,[\int_{\infty}^{y} pdf(x)dx]^{(p-t)}$$

Following equation gives the expectation of the variable **y**.

$$\int_{-\infty}^{\infty} p\ y\ pdf(y) \text{ x} (\int_{-\infty}^{y} p\ df(x)dx)^{(p-1)} \text{ x } dy = ExBp \text{ --------------- } (2)$$

For convenience, **ExB(p)** denotes the expectation of the largest sample among **p** samples from the standard normal distribution. In addition, by substituting pdf(x) in Eq.(2) with Eq.(3), we can find the numerical values of ExB(p) for various **p**. We show the results in the middle column of Table 1.

$$p\ df(x) = \frac{1}{\sqrt{2\pi}}\exp(-x^2/ 2) \text{ -------------- } (3)$$

Number of samples (p)	**Value using Eq. 2**	**Expected values**
1	0.00000	-0.00001
2	0.56419	0.56419
4	1.02938	1.02938
8	1.42360	1.42360
16	1.76599	1.76599
32	2.06967	2.06967
64	2.34373	2.34373

Table 1 : Expectation of the largest sample (ExB(p)) from N(0, 1).

Because Eq.(2)'s complexity grows exponentially as **p** increases, we cannot find the exact numerical values of ExB(p) for **p>64**. To address this shortcoming, more scalable way of approximating the values in Table 1. In this solution, first implement a random number generator that produces random numbers from the standard normal distribution. Using this random number generator, the solution picks **p** independent random samples and remembers the largest sample among them. This operation continues for a long enough time (for example, to produce the results in Table 1, simulator software may take millions of iterations.) and averages the largest samples.[14] This experimental method generates the exact numerical values of ExB(p), as shown in the third column of Table 1, after averaging more than millions of trials. As a comparison of the second , third and fourth columns in the table shows, the mathematical accuracy is slightly compromised in exchange for scalability. The study of the largest sample in the standard normal distribution gives us an idea about the ExB(p) for other normal distributions.

Let a random variable X follow $N(\mu, \sigma^2)$ with $\mu \neq 0$, $\sigma \neq 1$, $\sigma \neq 0$, and a derived random variable $Y = (X - \mu)/\sigma$. Then, Y follows N(0, 1) by recalling the property that if X follows $N(\mu, \sigma^2)$ and a and b are real numbers, then aX + b follows $N(a\mu + b,(a\sigma)^2)$. From Equ. (2), the expectation of the largest sample for Y is ExB(p) because $Y = (X - \mu)/\sigma$, $X = Y\sigma + \mu$; and the expectation of the largest sample for X is ExB(p) · σ + μ. Now, we can calculate the expectation of the largest sample for any arbitrary normal distribution.

D. Execution Time And Energy Consumption Analysis

In our model, each of the **p** virtual machines is responsible for **m/p** job units, and the response time for each job unit follows U(a,(a+((b– a)p)/n)). We use the following equation to calculate the expectation of the time required on a virtual machine finishing last:

ISSN:2321-1156

International Journal of Innovative Research in Technology & Science(IJIRTS)

Execution time = μ+ ExB(p) x σ

$$= \frac{m}{p}\left(a + \frac{(b-a)r}{2n} + ExB(p) \times \sqrt{m/p} \times \frac{(b-1)r}{\sqrt{12}} \times \frac{1}{n}\right)$$

$$= \frac{ma}{2np}\left(2n + \left(\frac{b}{a} - 1\right)p + ExB(p) + p^{3/2}\sqrt{\frac{1}{3m}} \times \left(\frac{b}{a} - 1\right)\right)$$

$$= \frac{ma}{2np}\left(2n + \left(\frac{b}{a} - 1\right)p + Unbalance\left(\frac{b}{a}, p, m\right)\right)$$

In this equation, we name the second term unbalance, which becomes zero if and only if every virtual machine finishes at the same time.

$$Unbalance\left(\frac{b}{a}, p, m\right) = ExB(p) + p^{3/2}\sqrt{\frac{1}{3m}} \times \left(\frac{b}{a} - 1\right) \quad ---- (4)$$

For example, a higher deviation from the normal distribution indicates that the random samples from this distribution are more spread out, increasing the probability of having more deviated samples. In our case, because we model a virtual machine's finishing time by picking a sample from Equ. (1), more deviated samples indicate that the workload assignment is unbalanced among virtual machines executing this workload. In particular, a larger **b/a** will lead to a larger $\boldsymbol{\sigma^2}$ in Equ.(1) and a larger **Unbalance((b/a), p, m)** in Equ. (4). Hence, we can conclude that a larger **b/a** value cause a more unbalanced workload distribution among virtual machines, degrading the overall performance. Also note that Unbalance ((b/a), p, m) is directly proportional to **1/m** . Because **m** is independent of **p** and **b/a**, changing the value of **m** will not affect other variables in Equ. (4). This implies that a very large m will eventually zero out Eq. (4). Thus, we can express the execution time when m → ∞ as

$$\text{Execution time } (m \to \infty) = \frac{ma}{2np}\left[2n + \left(\frac{b}{a} - 1\right)p\right]$$

We also evaluate the energy consumption probabilistically. Because performance is bounded by the execution time of the virtual machine finishing last, we must calculate the expectation of the largest sample from Equ. (1). In contrast, to evaluate the utility consumption, we must focus on the average execution time of **p** virtual machines. This is because, in a normal distribution, the probability for having **μ + α** samples is exactly the same as having **μ – α** samples. This fact indicates that the difference of having a virtual machine consuming **α** seconds more than the average is the same as having a virtual machine consuming **α** seconds less than the average. Therefore, we conclude that the expectation of the total execution time is given by **μ × p**, the number of virtual machines. Given the power of a physical node in the cloud is W, the total energy consumption will be as follows.

$$\text{Energy consumption} = W \times \frac{m}{p}\left[a + \frac{(b-a)r}{2n}\right]p$$

$$= W \times m \left[a + \frac{(b-a)r}{2n}\right]$$

$$EDP_{exp}(p) = \frac{Wm^2a^2}{4n^2p} \times \left\{2n + \left(\frac{b}{a} - 1\right)p + Unbalance\left(\frac{b}{a}, p, m\right)\right\} \times \left(2n + \left(\frac{b}{a} - 1\right)p\right)$$

$$= \frac{EDP_{base}}{4n^2p} \times \left\{2n + \left(\frac{b}{a} - 1\right)p + Unbalance\left(\frac{b}{a}, p, m\right)\right\}\left(2n + \left(\frac{b}{a} - 1\right)p\right)$$

Similarly, we calculate the EDP for m → ∞ as follows.

$$EDP_{exp,m\to\infty}(p) = \frac{EDP_{base}}{4n^2p} \times \left\{2n + \left(\frac{b}{a} - 1\right)p\right\}^2 \quad --------- (5)$$

To visualize the effect of a large **m** in the EDP_{exp} metric, Fig. (4) shows the EDP analysis for m = 12n, m = 120n, and m → ∞ using the following coefficients: n = 16,384, b/a = 1, 2, 3, 5, and ExB(p) from Table 1. To find the exact value of p that makes the EDP metric a global minimum point, we take the derivative of Equ. (5) with respect to p and set it to zero :

$$\frac{d}{dp}\left\{EDP_{base} \times \frac{\{2n + (\frac{b}{a} - 1)p\}^2}{4n^2p}\right\} = 0 \quad ----------- (6)$$

Here $P = 2n/\left(\frac{b}{a} - 1\right)$ since p > 0.

In the example of m → ∞ in Figure (4), we achieve the minimum EDP when p = 2n/(b/a – 1) = 16384 in Figure (4c) or p = 2n/(b/a – 1) = 8192 in Figure (4d). Again, p = n must be fulfilled while maintaining Equ. (6) to be energy-effective for all n virtual machines in the cloud. By combining two conditions, p = n and Equ. (6), we can calculate the requirement of b/a as n = 2n/(b/a –1); b/a = 3. This equation suggests that in a heterogeneous cloud computing environment with uniformly distributed performance, physical nodes that respond 3 times slower than the fastest node should not be used when attempting to minimize the EDP.

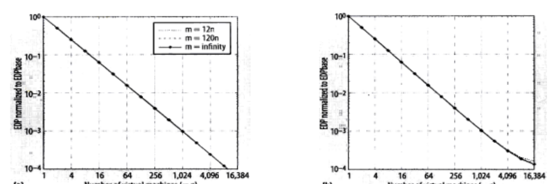

Figure 4(a) when *b*/*a* = 1 **Figure 4(b)** when *b*/*a* = 2

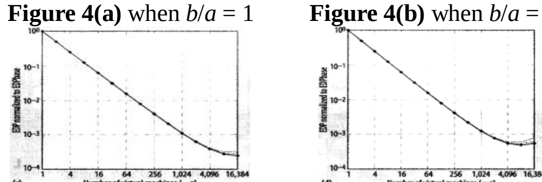

Figure 4(c) when *b*/*a* = 3 **Figure 4(d)** when *b*/*a* = 5

Figure 4. Example of expectation-based analysis where the total number of available virtual machines is 64 and above: When (response time of slowest node b / response time of fastest node a) >3, using all available virtual machines shows deviation in EDP.

Acknowledgments

The authors are thankful to IJIRTS Journal for the support to develop this document.

References

[1] M. Palankar et.al., "Amazon S3 for Science Grids: A Viable Solution?" Proc. 2008 Int'l Workshop Data-Aware Distributed Computing, ACM Press, 2008, pp. 55-64.

[2] L.A. Barroso et.al., "The Price of Performance," ACM Queue, vol. 3, no. 7, 2005, pp. 48-53.

[3] S. Ghiasi, T. Keller, and F. Rawson et.al., "Scheduling for Heterogeneous Processors in Server Systems," Proc. 2nd Conf. Computing Frontiers, ACM Press, 2005, pp. 199-210.

[4] R. Nathuji, C. Isci, and E. Gorbatov et.al., "Exploiting Platform Heterogeneity for Power-Efficient Data centres," Proc. 4th Int'l Conf. Autonomic Computing (ICAC 07), IEEE CS Press, 2007, pp. 5-14.

[5] PassMark Software, "CPU Benchmarks;" www.cpubenchmark.net.

[6] S. Pelley et.al., "Power Routing: Dynamic Power Provisioning in the Datacenter," Proc. 15th Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS 10), ACM Press, 2010, pp. 231-242.

[7] D. Meisner, B.T. Gold, and T.F. Wenisch et.al., "PowerNap: Eliminating Server Idle Power," Proc. 14th Int'l Conf. Architectural Support for Programming Languages and Operating Systems, (ASPLOS 09), ACM Press, 2009, pp. 205-216.

[8] R. Gonzalez and M. Horowitz, "Energy Dissipation in General Purpose Processors," IEEE . J.A. Rice, Mathematical Statistics and Data Analysis, Dux- bury Press, 2007.

[9] Using Mathematical Modeling in Provisioning a Heterogeneous Cloud Computing Environment - Sungkap Yeo and Hsien-Hsin S. Lee et.al., Georgia Institute of Technology – 2011.

[10] Cloud Computing: A Synthesis Model for Resource Service Management, MengkunLi, Ming Chen, JunXie et.al., 201O Second International Conference on Communication Systems, Networks and Applications.

[11] Dynamic Optimization of Multiattribute Resource Allocation in Self-Organizing Clouds, Sheng Di , Member, Cho-Li Wang et.al., Member, IEEE, , VOL. 24, NO. 3, MARCH 2013.

[12] Dynamic Resource Allocation in Cloud Environment Under Time-variant Job Requests, DavideTammaro, Elias A. Doumith, Sawsan Al Zahr, Jean-Paul Smets, and Maurice Gagnair et.al., 2011 Third IEEE International Conference on Coud Computing Technology and Science.

[13] Introduction to Cloud Computing Architecture White Paper 1st Edition, June 2009.

[14] The Economics of Cloud Computing, Diversity Limited, 2011 Non-commercial reuse with attribution permitted.

[15] E-Government Information Systems and Cloud Computing (Readiness and Analysis) RabeaKurdi, A. Taleb-Bendiab, Martin Randles, and Mark Taylor School of Computing and Mathematical Sciences, John Moores University Liverpool, United Kingdom.

[16] http : //www.computer.org / csdl / mags /co / 2011 / 08 / mco2011080055-abs . html

Biographies

FIRST A. R.T.KAPLE received the B.SC. degree in Mathematics from the University of Nagpur, Nagpur, Maharashtra State, in 1985, the M.SC. degree in Mathematics from the University of Amravati, Amravati, Maharashtra State, in 1989, and the Post Graduate Diploma in CS from University of Amravati, Maharashtra in 1988. He (She) is an associate Programmer of Computer Sciecen and Engineering at BNCOE, SGBAU Amravati University of Maharashtra state. His professional and research areas include Application Software, IPv4, IPv6, OS , and Cloud Computing. He may be reached at rtkaple@rediffmail.com .

SECOND B. V.R.CHIRADE received the B.SC. degree in Mathematics from the University of Nagpur, the M.SC. degree in Mathematics from the University of Nagpur, and the Ph.D. from RSTDM University Nagpur, Maharashtra. He is Head and Associate Professor at GSG College Umarkhed, SGBAU Amravati University of Maharashtra state. His teaching and research areas include. He may be reached at vrchirde333@rediffmail.com .