Performance Evaluation of Cloud Service Considering Fault Recovery

Bo Yang[1], Feng Tan[2], Yuan-Shun Dai[3], and Suchang Guo[2]

[1] Collaborative Autonomic Computing Laboratory, University of Electronic Science and Technology of China, Chengdu, China
yangbo@uestc.edu.cn
[2] Department of Industrial Engineering, University of Electronic Science and Technology of China, Chengdu, China
tf.uestc@gmail.com, schguo@uestc.edu.cn
[3] Innovative Computing Laboratory, Department of Electrical Engineering & Computer Science, University of Tennessee, Knoxville, USA
ydai1@eecs.utk.edu

Abstract. In cloud computing, cloud service performance is an important issue. To improve cloud service reliability, fault recovery may be used. However, the use of fault recovery could have impact on the performance of cloud service. In this paper, we conduct a preliminary study on this issue. Cloud service performance is quantified by service response time, whose probability density function as well as the mean is derived.

Keywords: Cloud service, performance evaluation, fault recovery.

1 Introduction

In recent years, new computing paradigms have been proposed and adopted such as cloud computing. Cloud computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services [1].

For cloud service, two issues are of great importance, the reliability and the performance of cloud service. Cloud service reliability is concerned with how probable that the cloud can successfully provide the service requested by users. Cloud service performance, on the other hand, is concerned with how fast the cloud can provide the requested service.

In cloud computing, fault recovery, one of the fault tolerance techniques, may be used to improve cloud service reliability [2]. Nevertheless, the use of fault recovery could have impact on the performance of cloud service. In this paper, we study this issue in detail, and we consider only the situation where the cloud can successfully provide the requested service, because research on cloud service reliability is out of the scope of this paper.

M.G. Jaatun, G. Zhao, and C. Rong (Eds.): CloudCom 2009, LNCS 5931, pp. 571–576, 2009.

2 Service Response Time Modeling

When a user submits a service request to the cloud, the request will first arrive at the *cloud management system* (CMS) which maintains a request queue. If the queue is not full, the request will enter the queue; otherwise it will be dropped and the requested service fails. A request that successfully enters the queue will then reach a *scheduler* which will divide the request into several subtasks and assign the subtasks to different processing nodes (nodes). Generally there are multiple schedulers which are homogeneous with similar structures, schemes and equipments, to serve the requests.

After all subtasks are assigned to corresponding nodes, they will be executed on the nodes. During the execution of subtasks, there may be exchange of data through communication channels. After all subtasks are completed, the results are returned and integrated into a final output, which is sent to the user.

It is thus clear that the *service response time*, T_{SRT}, is given by

$$T_{SRT} = T_{SUB} + T_W + T_S + T_E + T_R, \tag{1}$$

where T_{SUB} is the *submission time* (the time period from the user's initiating a service request until the request arrives at the CMS); T_W is the *waiting time* (the time period from a request's arrival at the CMS until it begins to be served); T_S is the *service time*; T_E is the *total execution time* of the service (which will be explained in detail later); and T_R is the *return time* (the time spent in obtaining results from subtasks, integrating them, and sending the final output to the user).

In this paper, we assume that T_{SUB} and T_R are negligible. Thus (1) can be rewritten as

$$T_{SRT} = T_W + T_S + T_E. \tag{2}$$

We further assume T_W, T_S and T_E are statistically independent (*s*-independent).

2.1 Waiting Time and Service Time

Assume that the service requests arrive at the CMS according a Poisson process with arrival rate λ_a, and the service times of the requests are identical independently distributed (i.i.d.) random variables (r.v.'s) following exponential distribution with parameter μ_r (service rate). Assume that there are S ($S \ge 1$) homogeneous schedulers. Denote by N ($N \ge S$) the system capacity, i.e., the maximum allowed number of requests in the queueing system (including the ones waiting in the queue).

Denote by p_j ($j = 0,1,\cdots,N$) the steady-state probability that there are j requests in the system, then from standard queueing theory we have

$$p_j = \left[\sum_{i=0}^{S-1}\frac{\rho^i}{i!} + \sum_{i=S}^{N}\frac{\rho^i}{S!S^{i-S}}\right]^{-1} (j=0); \frac{\rho^j}{j!}p_0 (1 \le j < S); \frac{\rho^j}{S!S^{j-S}}p_0 (S \le j \le N), \tag{3}$$

where $\rho \equiv \dfrac{\lambda_a}{\mu_r}$.

When a new request arrives, it may or may not enter the queue. For those requests which actually enter the queue, the steady-state probability that a request finds *j* requests (excluding itself) in the system, denoted by q_j, is

$$q_j = \frac{p_j}{1-p_N}, \; j = 0,1,\cdots,N-1. \quad (4)$$

When a request enters the system, if there are *j* ($0 \le j \le S-1$) requests in the system, then it will be served immediately, thus the waiting time $T_W = 0$. If there are *j* ($S \le j \le N-1$) requests in the system, then the waiting time T_W follows gamma distribution of order $j-S+1$ with scale parameter $\mu_r S$. Therefore, the cumulative distribution function (c.d.f.) of T_W is

$$F_{T_W}(t) \equiv \Pr\{T_W \le t\} = 0(t<0); \sum_{j=0}^{S-1} q_j (t=0); \sum_{j=0}^{S-1} q_j + \sum_{j=S}^{N-1} q_j \frac{\Gamma(j-S+1, \mu_r St)}{\Gamma(j-S+1)} (t>0), \quad (5)$$

where $\Gamma(\eta) \equiv (\eta-1)!$ for any positive integer η ; and $\Gamma(\eta,u) \equiv \int_0^u x^{\eta-1} \exp(-x) dx$.

The probability density function (p.d.f.) of T_W is

$$f_{T_W}(t) = \sum_{j=S}^{N-1} q_j \frac{\mu_r S(\mu_r St)^{j-S}}{\Gamma(j-S+1)} \exp(-\mu_r St), t \ge 0. \quad (6)$$

The mean of T_W is

$$E[T_W] = \sum_{j=S}^{N-1} q_j \frac{j-S+1}{\mu_r S}. \quad (7)$$

The service time, T_S, follows exponential distribution with parameter $\mu_r S$.

2.2 Total Execution Time

Suppose a service request is divided into M ($M \ge 1$) subtasks, which are assigned to N ($N \ge 1$) nodes for execution. Suppose the *i*:th subtask is assigned to the *j*:th node. Denote by wp_i the workload of the *i*:th subtask, and denote by ps_j the processing speed of the *j*:th node, then the *required execution time*, denoted by τ_{ij}, is

$$\tau_{ij} = \frac{wp_i}{ps_j}, \; i = 1,2,\cdots,M, \; j = 1,2,\cdots,N. \quad (10)$$

However, when fault recovery is adopted, the *actual execution time*, denoted by T_{ij}, is different from τ_{ij}. When the subtask fails during execution, if the failure is recoverable, then after some time (recovery time), the node will resume the execution of this subtask. In this paper, we only consider the case in which all the failures are recoverable, since if there occurs any unrecoverable failure, the subtask will fail and thus the service fails, which is out of the scope of this paper.

Assume that the *j*:th node has a constant failure rate of λ_j. Denote by $N_j(t)$ the total number of failures that occurs on the *j*:th node during time interval (0, *t*], then

$$\Pr\{N_j(t)=n\}=\frac{(\lambda_j t)^n}{n!}\exp(-\lambda_j t)\,,\ n=0,1,\cdots. \tag{11}$$

Denote by $TR_j^{(k)}$ the *k*:th recovery time on node *j*, and we assume that $TR_j^{(k)}$'s are i.i.d. exponential r.v.'s with parameter μ_j. The *total recovery time*, $TR_j(t)$, is

$$TR_j(t)=\sum_{k=1}^{N_j(t)} TR_j^{(k)}\,. \tag{12}$$

It can be seen that $TR_j(t)$ is a compound Poisson Process, whose mean is:

$$E[TR_j(t)]=\lambda_j t\cdot E[TR_j^{(k)}]=\frac{\lambda_j t}{\mu_j}\,. \tag{13}$$

Since the required execution time is τ_{ij}, we have

$$T_{ij}=\begin{cases}\tau_{ij} & N_j(\tau_{ij})=0\\ \tau_{ij}+TR_j(\tau_{ij}) & N_j(\tau_{ij})=1,2,\cdots\end{cases} \tag{14}$$

To derive the distribution of T_{ij}, we need first obtain the distribution of $TR_j(\tau_{ij})$. If $N_j(\tau_{ij})=n$ ($n=1,2,\cdots$), then from (12) we know that $TR_j(\tau_{ij})$ follows gamma distribution of order n with scale parameter μ_j. Thus the c.d.f. of $TR_j(\tau_{ij})$ is

$$F_{TR_j(\tau_{ij})}(t)\equiv\Pr\{TR_j(\tau_{ij})\le t\}=\sum_{n=1}^{\infty}\Pr\{N_j(\tau_{ij})=n\}\frac{\Gamma(n,\mu_j t)}{\Gamma(n)},t\ge 0\,. \tag{15}$$

From (11), we have

$$F_{TR_j(\tau_{ij})}(t)=\exp(-\lambda_j\tau_{ij})\sum_{n=1}^{\infty}(\lambda_j\tau_{ij})^n\frac{\Gamma(n,\mu_j t)}{\Gamma(n+1)\Gamma(n)},t\ge 0\,. \tag{16}$$

Therefore, from (14), the c.d.f. of T_{ij} is

$$F_{T_{ij}}(t)\equiv\Pr\{T_{ij}\le t\}=\Pr\{T_{ij}-\tau_{ij}\le\kappa_{ij}\}=\exp(-\lambda_j\tau_{ij})\left[1+\sum_{n=1}^{\infty}(\lambda_j\tau_{ij})^n\frac{\Gamma(n,\mu_j\kappa_{ij})}{\Gamma(n+1)\Gamma(n)}\right],t\ge\tau_{ij} \tag{17}$$

where $\kappa_{ij}\equiv t-\tau_{ij}$. The p.d.f. of T_{ij} is

$$f_{T_{ij}}(t)=\frac{\exp(-\lambda_j\tau_{ij}-\mu_j\kappa_{ij})}{\kappa_{ij}}\sum_{n=1}^{\infty}\frac{(\lambda_j\mu_j\tau_{ij}\kappa_{ij})^n}{\Gamma(n+1)\Gamma(n)},t\ge\tau_{ij}\,. \tag{18}$$

From (13) and (14) we can obtain the mean of T_{ij}, which is

$$E[T_{ij}] = \tau_{ij}\left(1+\frac{\lambda_j}{\mu_j}\right) - \tau_{ij}\frac{\lambda_j}{\mu_j}\exp(-\lambda_j \tau_{ij}) . \quad (19)$$

During subtasks' execution, a subtask may need to exchange data with other subtasks being executed on remote nodes, thus communication time is involved. We assume that during the execution of a subtask, if the subtask needs to exchange data with another node for further execution, the subtask has to be paused and wait for the completion of data exchange. During the communication with the remote node, the subtask is idle. After data exchange is completed, the subtask resumes its execution.

Denote by $D(i)$ the set of communication channels that the *i*:th subtask uses to exchange data with remote nodes. Denote by c_{ik} the amount of data that the *i*:th subtask exchanges through the *k*:th communication channel which has a bandwidth of bw_k. Assume that during data exchange, communication channels will not fail, then the communication time of the *i*:th subtask through the *k*:th communication channel is

$$s_{ik} = \frac{c_{ik}}{bw_k}, \; i = 1,2,\cdots,M \;, \; k \in D(i) . \quad (20)$$

The *total communication time* of the *i*:th subtask, denoted by y_i, is

$$y_i = \sum_{k \in D(i)} s_{ik} . \quad (21)$$

The *total execution time* of the *i*:th subtask, denoted by Z_{ij}, is thus given by

$$Z_{ij} = T_{ij} + y_i . \quad (22)$$

From (17), the c.d.f. of Z_{ij}, denoted by $F_{Z_{ij}}(t)$, can be easily derived. Its mean can also be easily derived from (19)

Since all subtasks are executed in parallel, the *total execution time* of the service, T_E, is the maximum of Z_{ij}'s. Assume that Z_{ij}'s are *s*-independent, then T_E has a maximum extreme-value distribution, whose c.d.f. and p.d.f. are respectively

$$F_{T_E}(t) = \prod_{i=1}^{M} F_{Z_{ij}}(t) , \; f_{T_E}(t) = \sum_{k=1}^{M} f_{Z_{kj}}(t) \prod_{1 \le i \le M, i \ne k} F_{Z_{ij}}(t) . \quad (23)$$

The mean of T_E is

$$E[T_E] = \int_0^{\infty}\left[1 - \prod_{i=1}^{M}\int_0^x f_{Z_{ij}}(t)dt\right]dx . \quad (24)$$

2.3 Service Response Time

From (2), the mean of the service response time, T_{SRT}, is

$$E[T_{SRT}] = E[T_W] + \frac{1}{\mu_r S} + E[T_E] , \quad (25)$$

where $E[T_W]$ and $E[T_E]$ are given by (7) and (24), respectively.

By the assumption that T_W , T_S and T_E are *s*-independent, the p.d.f. of T_{SRT} is

$$f_{T_{SRT}}(t) = f_{T_W}(t) \otimes f_{T_S}(t) \otimes f_{T_E}(t) . \tag{26}$$

3 Conclusion

In this paper, we conduct a preliminary study on cloud service performance considering fault recovery. Cloud service performance is quantified by the service response time, whose probability density function as well as the mean is derived.

However, we have made several assumptions which may not be realistic, e.g., the *s*-independence. Moreover, fault recovery may be adopted for communication channels as well. We shall address these issues in our future work.

Acknowledgments. This work is supported by National Natural Science Foundation of China (No. 50805018) and Key Project of Chinese Ministry of Education (No. 109138).

References

1. Armbrust, M., Fox, A., Griffith, R., et al.: Above the clouds: A Berkeley view of cloud computing. Technical Report No. UCB/EECS-2009-28, University of California at Berkeley (2009)
2. Guo, S., Yang, B., Huang, H.Z.: Grid service reliability modeling on fault recovery and optimal task scheduling. In: Proceedings of the 55th Annual Reliability & Maintainability Symposium (RAMS 2009), Fort Worth, Texas, USA, January 2009, pp. 471–476 (2009)