Scale-Out vs. Scale-Up Techniques for Cloud Performance and Productivity

Article · February 2015

DOI: 10.1109/CloudCom.2014.66

CITATIONS
15

READS
519

3 authors, including:

Yue Shi
University of Southern California
4 PUBLICATIONS **127** CITATIONS
SEE PROFILE

Xiaoying Bai
Tsinghua University
89 PUBLICATIONS **1,998** CITATIONS
SEE PROFILE

Scale-Out vs. Scale-Up Techniques for Cloud Performance and Productivity

Kai Hwang and Yue Shi
University of Southern California
Los Angeles, CA. USA
{kaihwang, yueshi}@usc.edu

Xiaoying Bai
Tsinghua University
Beijing, China
baixy@tsinghua.edu.cn

***Abstract* – An elastic cloud provisions machine instances upon user demand. Auto-scaling, scale-out, scale-up, or any mixture techniques are used to reconfigure the user cluster as workload changes. We evaluate three scaling strategies to upgrade the performance, efficiency and productivity of elastic clouds like EC2, Rackspace, etc. We developed new performance models and run the HiBench benchmark to test Hadoop performance on various EC2 configurations.**

The strengths and shortcomings of three scaling strategies are revealed in our HiBench experiments: (1). Scale-out overhead is shown lower than that experienced in scale-up or mixed scaling clouds. Scale-out to a larger cluster of small nodes demonstrated high scalability. (2). Scaling up and mixed scaling have high performance in using smaller clusters with a few powerful machine instances. (3). With a mixed scaling mode, the cloud productivity is shown upgradable with higher flexibility in applications with performance/cost tradeoffs.

Keywords— Cloud computing; elastic resources; cloud productivity; cloud performance modeling

I. INTRODUCTION

To maximize the resource utilization at reduced service cost, elastic and dynamic resource provisioning is the key to assure cloud performance [11]. For data-intensive applications on the cloud, we demand the effective use of scalable resources. *Scaling out* allows one to acquire more processing nodes (machine instances) of the same type. *Scaling up* allows the upgrade of computing nodes from small to large or extra-large nodes. These two strategies could be used separately or mixed to yield high performance.

We study the pros and cons of 3 major elastic scaling techniques that are implemented under program directives or assisted by workload predictions. We will not include the brutal-force *auto-scaling method* due to its mismatch with the workload variations. Our work explores all optimization opportunities brought over by provisioning elastic resources.

This paper models the performance for IaaS, PaaS and SaaS clouds at different abstraction levels. We experiment with the HiBench programs targeted at clouds. Then we analyze some new findings from measured EC2 performance. We assess the state of cloud computing from the perspectives of performance and productivity.

** IEEE CloudCom 2014 Workshop on Emerging Issues in Cloud (EIC), Singapore, Dec.15-18, 2014, Corresponding author: Kai Hwang, kaihwang@usc.edu.*

This work is extended from previous works on cloud performance benchmarking [1, 3, 4, 8, 9, 12, 14, 16, 20].

To analyze the scalability of cloud performance, one needs to address the following 4 fundamental issues [7]. This paper attempts to meet these demands.

1) *Scaling measurement:* Cloud scaling is done with virtualized resources. Hence, the scale of computing power is decided at various abstraction levels of virtual resources.

2) *Workload scenario:* Cloud aims to accommodate workload with large number of small jobs. Scaling strategies must match with such a workload scenario.

3) *Performance attributes:* To benefit a large number of small jobs, performance concerns are the response time and throughput, rather than batch execution time.

4) *Cloud productivity:* Productivity is tied to performance cost ratio. Tradeoffs do exist in high performance versus service costs to massive users.

This paper proposes new cloud performance models using Keviate graphs (radar charts). We identify a comprehensive list of performance metrics for users to select from in the evaluation of cloud performance.

II. WORKLOAD, BENCHMARKS AND METRICS

Due to multi-tenant demands, clouds are facing all sorts of workloads including multi-tasking, batch processing, streaming, data-mining and analytics. The cloud workload must be matched with adequately configured resources to achieve high performance and sustained productivity.

A. Scale-Out, Scale-Up, and Mixed Workloads

Clouds are used primarily for data-intensive and latency-sensitive jobs, search engines, OLTP/business processing, social-media networking, data warehousing and big-data analytics. Cloud wworkloads are characterized by their dataset size, algorithms, memory-access pattern, and service model applied. Our scaling techniques cover three cloud workload types as characterized below:

- *Scale-out technique* allows adding more machine instances or processing nodes of the same type based on the quota agreed in the *service-level agreement* (SLA). Obviously, scaling out appeals more to the use of homogeneous clusters with identical nodes.

978-1-4799-4093-6/14 $31.00 © 2014 IEEE
DOI 10.1109/CloudCom.2014.66

- *Scale-up technique* is implemented with scaling the cloud from using small nodes to more powerful nodes equipped with better processor, memory or storage.
- *Mixed scale-up/scale-out technique* allows one to scale up or scale-down the instance type and adjust the instance quantity by scale-out (increasing) or scale-in (reducing) resources at the same time. Mixed scaling appeals better with using heterogeneous clusters.

CloudHarmony offers an UnixBench [5] workloads to test CPU performance, I/O and storage performance in cloud environment. Elastic cloud relies on virtualization technique to enable dynamic resource provisioning or de-provisioning. The multi-tenant architecture is practiced in most clouds to support big-data processing and composite web services..

B. Existing Cloud Benchmark Suites

Two commercial cloud evaluations were conducted recently. Nine large cloud providers were evaluated by BitCurrent [2] and 144 cloud sites were examined by CloudHarmonics [5]. However, the performance metrics they have applied are far from being adequate to cover the QoS or productivity in clouds. Our performance analysis applies the HiBench by Intel [10].

HiBench is specifically tailored for running Hadoop programs on most clouds. The suite was developed for measuring the speed, throughput, HDFS bandwidth, and resources utilization in a large suite of programs. Other cloud benchmarks include the YCSB (for *Yahoo! Cloud Serving Benchmark*) [6], CloudSuite [8], CloudCmp [15], Phoronix [17], TPC-W [18], CloudStone [19], and C-meter [20]. Interested readers are refer to the assessment by Farber and Kounev [7] for cloud benchmarking trends.

C. Performance Metrics Targetted at Elastic Clouds

We apply an extended concept of "cloud performance" to include three ascending categories: *performance, ccapabilities* and *productivity.* Performance and capabilities are necessary to enhance the productivity of a cloud. Analytical expressions for some new performance metrics like *efficiency, productivity* and *scalability* are given in subsequent sections.

(1) . *Basic Performance Metrics* include traditional measures of speed, speedup, efficiency, etc. [12].

- *Speed* (*S*)*:* Number of *millions of operations per second* (*Mops*). The operation could be integer or floating-point like *MFlops.* The speed is also known as *throughput by some Benchmark* such as *millions of web interactions per second* (*WIPS*), etc.
- *Speedup* (S_u): Speed gain of using multiple nodes
- *Efficiency* (E_f): Percentage of peak performance achieved
- *Utilization* (*U*)*:* Busy resources (CPU, memory, storage).
- *Scalability* (S_c) *:* Scaling ability to upgrade performance.

(2). *Cloud Capabilities* are macroscopic metrics that describe the capabilities of a cloud as listed below.

- *Latency* (*L*): System response time or access latency
- *Bandwidth* (*B*): This is data transfer rate or I/O rate.
- *Elasticity* (E_l): The ability for cloud resources to scale up/down or scale in/out to match with workload variation
- *Software* (S_w) : Software portability, API and SDK tooling
- *Big-data Analytics:*(A_n)*:* The ability to uncover hidden information or predict trends in big data.

(3) . *Productivity Measures*: These refer to the following cost-performance issues.

- *Quality of Service* (*QoS*): Satisfaction on user services
- *System availability* (*A*)*: The s*ystem up time per year.
- *Service costs* (C_o): User renting costs and provider cost.
- *Power Demand* (*W*)*:* Cloud power consumption (MWatt).
- *SLA/Security* (*L*) *:* Compliance of SLA, security, etc.
- *Productivity* (*P*) *:* QoS-satisfied performance per unit cost

III. CLOUD EFFICIENCY AND PRODUCTIVITY

The scalability concept was developed with parallel computing. Elasticity was introduce with the inception of cloud computing. Productivity of clouds is newly introduced to combine technical performance with business gains in cloud systems

A. *Scalability Analysis in Cloud Computing*

Amazon AWS has defined a term *EC2 Compute Unit* (ECU) as an abstract unit to quantify the computing capacity of each instance type. By 2009 standard, the performance of a 1 ECU instance is roughly equivalent to the CPU capacity of a 1.2 GHz 2007 Xeon processor [1]. Each physical CPU (processor) can house a number a number of *virtual CPU* (vCPU). Also the memory and storage may affect the ECU count. Table 1 shows 4 machine instance types that we have used in HiBench scaling experiments.

Table 1: Machine Instance Types used in HiBench Experiments on Amazon EC2

Instance Type	ECU	VCPU (Cores)	Memory (GB)	Storage (GB)
m1.small	1	1	1.7	1 ×160
m1.medium	2	1	3.75	1×410
m1.large	4	2	7.5	2×420
m1.xlarge	8	4	15	4×420

The cloud rents resources by instance types and quantity. Hence, the configuration matrix is simplified to a configuration vector: $\Lambda = [a_{v_i}]_{1\times k}$, where k is the number of instance types and a_{vi} is the number of instance type vi.

B. Historical Development of Scaling Techniques

The performance of CPU-bound jobs is primarily decided by machine instance numbers. Memory-bound problems are limited by the memory (including cache) allocated within the machine instances. The storage-bound problems are limited by the network latency and disk storage and I/O bandwidth encountered.

(a). *Scaling-Out Technique:* This refers to the horizontal scaling of a given machine instance from n1 to n2 in quantity. The process is called *scale-out* if n2 > n1 and *scale-in* if the quantity is reduced. Note that there is no instance type change in this case, just the number of the same instances increasing or decreasing to match better with the workload demand.

(b). *Scaling-Up Technique* : To scale from an instance type v1 to another type v2, we have the *scale-up* process if v2 is more powerful than v1. Otherwise, it is called *scale-down*. For simplicity, we will use the ECU (*Elastic Compute Unit*) defined by AWS cloud for measuring the computing capacity of a given machine instance [11]. Scaling up or down is also known as *vertical scaling* in the traditional sense of parallel computing using multiprocessor or multicomputer systems [12].

(c) *Mixed Scaling-Out and Scaling-Up Technique:* This refers to the use of both scale-out and scale-up techniques at the same time. Obviously, this mixed strategy is the most difficult one to implement. However, it does offer much higher application flexibility to match with the workload demand.

C. Cloud Efficiency of Scaling Strategies

Consider a cluster configuration Λ. Let $T(1)$ be the execution time of an application code on a 1-ECU instance. Let $T(\Lambda)$ be the execution time of the same code on a virtual cluster Λ. The speedup is defined by *Speedup* $(\Lambda) = T(1) / T(\Lambda)$. Assume that the cluster is built with n instance types. The type-i has n_i instances, each with an ECU count c_i. We calculate the total cluster ECU count by:

$$N(\Lambda) = \sum_{i=1}^{i=n} n_i \times c_i \qquad (1)$$

This $N(\Lambda)$ count sets a ceiling of the cluster speedup. Now, we are ready to define the *cloud efficiency* for the cluster Λ in question as follow:

$$Efficiency\,(\Lambda) = Speedup\,(\Lambda) / N(\Lambda)$$

$$= T(1) / \{ T(\Lambda) \times \sum_{i=1}^{i=n} n_i \times c_i \} \qquad (2)$$

D. Cloud Productivity and Scalability

In general, the cloud *productivity* is driven by three technical factors that are related to the scaling factor.

1) System performance such as throughput in terms of transactions per second or response time.
2) System availability as an indicator of QoS measured by percentage of uptime.
3) Cost for rented resources measured by price.

Let Λ be a cloud configuration in use. We define the cloud *productivity* by three factors, all are functions of Λ.

$$P(\Lambda) = \frac{p(\Lambda)\times\omega(\Lambda)}{C(\Lambda)} \qquad (3)$$

where $p(\Lambda)$ is a *performance* metric used, which could the speed or throughput selected from the last section. The $\omega(\Lambda)$ is the *QoS* of the cloud. For simplicity, one can approximate the QoS by the *service availability* measure. According to CloudHarmonics Report on 144 cloud web sites [5], more than half have 99% or higher availability. The $C(\Lambda)$ is the user cost to rent resources to form the virtual cluster Λ.

The *cloud scalability* is driven by the productivity and QoS of a cloud system. This measure is inversely proportional to the service costs As we scale from configuration $\Lambda1$ to another $\Lambda2$. This metric evaluates the economy of scale by a pair of productivity ratio. The higher is the value of a scalability measure, the more opportunity exists to target the desired scaling scheme.

$$S(\Lambda1, \Lambda2) = \frac{P(\Lambda2)}{P(\Lambda1)} = \frac{p(\Lambda2)\times\omega(\Lambda2)\times C(\Lambda1))}{p(\Lambda1)\times\omega(\Lambda1)\times C(\Lambda2)} \qquad (4)$$

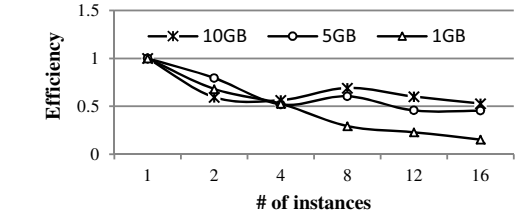
With comparable QoS and cost estimation, the scalability is directly proportional to productivity (Eq.3). Therefore, will demonstrate the measured productivity results and skip the scalability plots in subsequent sections.

IV. BENCHMARK RESULTS AND ANALYSIS

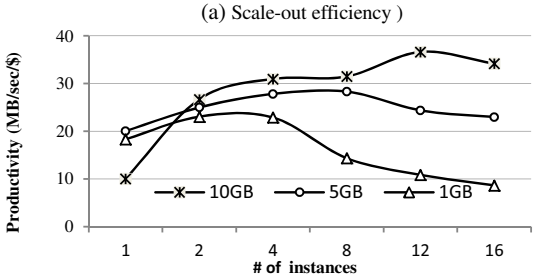
In this section, we report the cloud benchmark results over scale-out, scale-up and mixed scaling workloads.. We have tested two programs: *WordCount* and *Sort* in HiBench suite. Only the raw data on WordCount results are plotted below. Similar plots for Sort results are skipped due to page limit. However, In Fig.4, we compare the results from executing both programs.

A. Scale-Out Performance Results

HiBench is a cloud benchmark for testing Hadoop performance [10]. The experimental settings for scaling out on HiBench simply increase the quantity of machine instances used. We run in Fig.1 the HiBench-WordCount benchmark on *Elastic MapReduce* (EMR) clusters up to16 *m1.small* nodes.



(a) Scale-out efficiency)



(b) Scale-out productivity

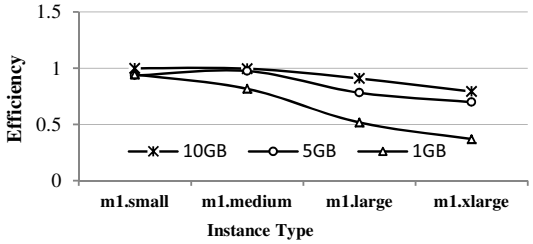
Figure 1. Scale-out performance of HiBench on EC2 built with up to 16 *m1.small* machine instances. Three curves correspond to executing 3 workload sizes in the WordCount benchmark program in HiBench.

We apply three workloads ranging from 1 to 5 and 10 GB of data elements. We report below 2 measured metrics: efficiency and productivity in Fig.1. The efficiency is a monotonic decreasing function of scale-out size (Fig.1a). The productivity in MB/s/$ increases from 1 node to a peak at 12, 8 and 4 nodes and then decreases with increasing cluster size (Fig.1b). The results reveal the fact that larger workload (10 GB) scales out much better than smaller workload (say 1 GB) verified by the high efficiency and productivity achieved.

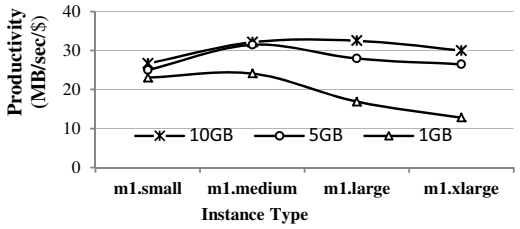
B. Scale-Up Performance Results

In pure scale-up experiments, we upgrade the machine instances from small to medium, large and extra-large types as given in Table 1. The purpose is to increase the computing power (ECU and vCPU), memory and storage capacities. The renting cost increases from small to large nodes as the cluster size stays the same with 2 identical nodes in Fig.2.

For executing the same WordCount program, the cluster built with large or extra nodes performs faster in speed. The speed also increases with workload sharply. Again the large workload (10 GB) has slower drop in efficiency as we scale up in node capacity. The productivity is also directly related to problem size. For all 3 datasets, the peak productivity occurs at using the medium-node cluster.



(a). Scale-up Efficiency



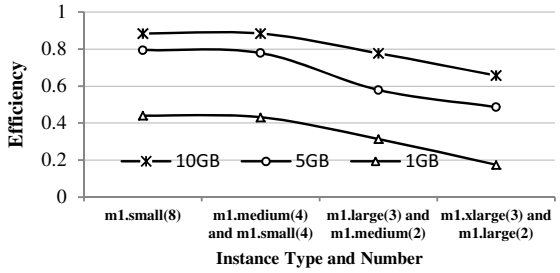
(b) Scale-up productivity

Figure 2. The performance of HiBench WordCount execution on 4 EC2 scale-up clusters with doubling ECU capacity

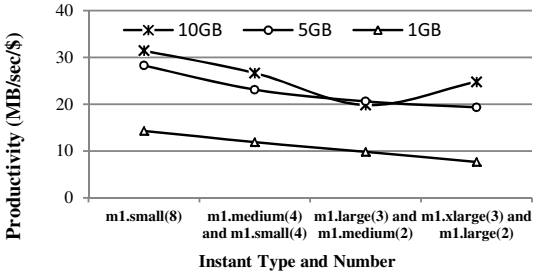
C. Mixed Scale-Up and Scale-Out Results

For mixed scaling, 4 cluster configurations are specified along the x-axis in Fig.3. The leftmost cluster has 8 *small* instances with a total ECU count of 8. The next has 4 *medium* and 4 *small* instances with 12 ECUs. The next one has 3 *large* and 2 medium instances with 16 ECUs. The right cluster has 3 *xlarge* and 2 *large* instances with 32 ECUs.

Obviously, the mixed scaling strategy offers much more flexibility in mapping applications with large workload variation. The speed of mixed scaling is the highest among the 3 methods. The efficiency varies similarly as the scale-up case. However, due to significant increase in computing capacity, the renting cost also increases proportionally. The productivity thus may drop to some extent with sharp increase in costs.



(a) Mixed scaling efficiency



(**b**) Mixed scaling roductivity

Figure 3: HiBench WordCount performance results on 4 EC2 clusters with mixed scale-up and scale-out nodes shown on the x-axis.

V. CLOUD PERFORMANCE MODELS

First, we introduce a generic cloud performance model. Then we will show how to extend or refine the generic framework to model IaaS, PaaS, and SaaS cloud performance. Each cloud performance model is specified by a 4-tuple expression, where *f* is the model function.

$$F(\textit{Cloud model}) = \{ < \textit{Service offerings} >, < \textit{Benchmark or app code under test} >, < \textit{Performance metrics} >, < \textit{Performance map generated} > \} \quad \textbf{(5)}$$

where the *cloud model* could be one or more of the available service modes such as IaaS (*Infrastructure as a Service*), PaaS (*Platform as a Service*) , SaaS (*Software as a Service*), HaaS (*Health-care as a Service*), etc. [11]. Thee *service offerings* vary with the provider such as < EC2, S3, EMR> offered by Amazon AWS cloud. The *benchmark* could be also specific user application code.

A. Modeling of Elastic IaaS Cloud

The IaaS model assumes a flat architecture providing infrastructure to client users. The IaaS providers are concerned about resource utilization and power consumptions. The end users are concerned about response time and prices, etc. The PaaS and SaaS vendors may apply a hierarchical architecture. This graphic model evaluates many other IaaS clouds such as Rackspace, GoGrid, FlexScale [11], and some Eucalyptus and OpenStack clouds [16].

$$F(IaaS) = \{ < EC2, S3, \ldots >, < Benchmarks >, <S_w, E_l, \Phi, B, A, C_o > , < Perf.\ Map > \} \quad (6)$$

We summarize the EC2 performance results in terms of *radar charts* (also known as Kiviat diagram or spider chart) in 6 performance dimensions. These charts are performance maps revealing the strength and weakness of 3 different scaling strategies on the cloud. The performance polygons in Fig.4 are plotted along 6 performance dimensions.

Each spoke of the polygon represents an attribute dimension. The attribute scale is proportional to the directional length along the spoke. The further away from the center, the higher performance is expressed in a scale from 0 to 5. Where value "0" means the least performance and "5" the highest performance value.

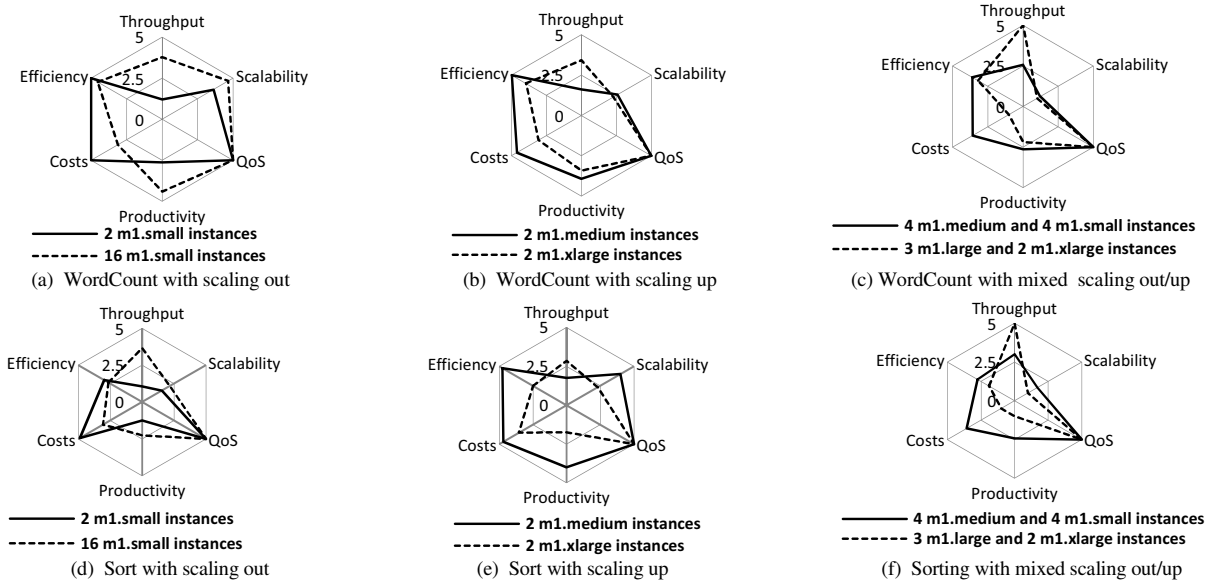
B. Performance of Different Benchmark Programs

The QoS is roughly indicated by system availability which was recorded 99.95% ~ 100% for all cluster configurations. Cost wise for the WordCount, the scale-out small cluster (solid polygons in Fig.4(a, d) has the least service costs. The scale-up clusters in Fig.4(b, e) cost more and the mixed cluster is the most expensive one to implement. Mixed scaling demands lot more considerations on tradeoffs between performance and cost incurred.

Speed wise, all mixed strategy for Sort (Fig.4c and Fig.4e) have the fastest throughput (or speed). The WordCount program shows slow throughput in all cases. The scale-up cluster shows very high efficiency for WordCount. The Sort clusters (dash-line polygons) show poor efficiency and throughput except high throughput for the mixed mode for sorting very large cluster in Fig.4f.

In Fig.4a, we see higher productivity for the large cluster (16 nodes) configuration. The peak values are application-dependent. Different benchmarks may lead to different conclusions. In general, scaling-out should be practiced when the elasticity speed is high.

These performance maps are further compared in Table 2 in terms of their polygon area values. Under each scaling case, we compare two cluster configurations. The polygon area values reported in Fig.4 and Table 2 simply demonstrate a radar-chart method to compare the relative performance of testing various cluster configurations with a common benchmark. .



(a) WordCount with scaling out (b) WordCount with scaling up (c) WordCount with mixed scaling out/up

(d) Sort with scaling out (e) Sort with scaling up (f) Sorting with mixed scaling out/up

Figure 4. The performance maps of two HiBench programs on two EC2 cluster configurations for the scale-out, scale-up, and mixed scale-up/scale-out workloads over 10 GB of data elements.

From Table 2, the 16-small cluster has higher performance in scale-out WordCount execution. In scale-up mode, the 2-medium-node cluster performs better in Sort execution. In mixed scaling mode, we rank the cluster of 4 *medium* and 4 *small* nodes with higher performance. Among the two benchmarks, scale-out performs the highest, scale-up next and the mixed scaling the lowest. In all cases, WordCount outperform Sort for having higher degree of parallelism.

Table 2 Performance Polygon Areas on Radar Charts in Fig.4

Scale-Out Mode (Figs.4a, d)	*Cluster Config.*	*2 small nodes*	*16 small nodes*
	WordCount	34.53	46.85
	Sort	17.02	23.65
Scale-Up Mode (Figs.4b, e)	*Cluster Config.*	*2 medium nodes*	*2 xlarge nodes*
	WordCount	37.25	31.42
	Sort	41.84	21.22
Mixed Scaling Mode (Figs. 4c, f)	*Cluster Config.*	*4 medium and 4 small*	*3 large and 2 xlarge*
	WordCount	23.39	18.28
	Sort	22.81	11.90

VI. CONCLUSIONS AND DISCUSSIONS

In general, higher efficiency promotes the productivity, but the converse may not hold, necessarily. The QoS is often assessed by users objectively. Different users may set their own satisfaction threshold for the QoS they can accept. The efficiency is controlled by the providers considering the interest of all user interests at the same time. We summarize below our major research findings. Then, we suggest a few directions for further R/D effort.

A. Summary of Orginal Findings

Our major contribution lies in the performance metrics and evaluation methodology provided. Cloud performance and efficiency are proven highly relevant to the productivity in evaluating cloud services.

(1). New performance metrics and benchmarking models are proposed and tested with HiBench benchmarks.

(2). With lower reconfiguration overhead, scaling-out should be practiced when the elasticity is high. Scaling-up is in favor of using powerful nodes at a higher cost.

(3). Pure scale-out or pure scaling-up are more cost-effective with higher efficiency and productivity.

(4). The cloud productivity is attributed to system elasticity, efficiency, and QoS (availability) driven by performance. The cloud providers must enforce performance isolation for quota-abiding users at the expense of over-quota users.

(5). Scaling up and mixed scaling have higher performance in using smaller clusters with a few very powerful machine instances.

B. *Impact and Suggestions for Further Work*

Taking a macroscopic view of the reported numerical results, we observe the fact that cloud performance is limited by many service constraints. Over all, we find that scaling-out is the easiest one to implement on homogeneous clusters. The elasticity overhead is much lower in scaled-out cluster configurations.

Scaling up is more complex to implement than scaling out due to the switching of node types. Scaling up will slow down the elasticity speed and prolong the reconfiguration overhead. The mixed scaling is the most difficult one to implement for having much higher reconfiguration overhead. However, mixed scaling offers the best application flexibility to match with workload change.

Two suggestions are made below for further work. The ultimate goal is to generate commonly accepted cloud benchmarks and testing techniques.

(a) Other cloud benchmarks like YCSB, CloudSuite, TPC-W, Phoronix, CloudStone, CloudCmp, and C-meter could be tested with the new performance models.

(b). The cloud community is short of benchmarks to test cloud capability in big-data analytics and machine learning intelligence.

ACKNOLEDGMENT

The research reported here was supported in parts by National Basic Research Program of China under 973 Grant No.2011CB302505, 863 Grant No.2012AA012600, and NSF of China Grant No.61073003. Kai Hwang would like to acknowledge the support of a visiting chair professorship endowed by EMC Cooperation at Tsinghua University.

REFERENCES

[1] Appuswamy, R., et al, "Scale-Up vs Scale-Out for Hadoop: Time to Rethink", *Proc.. of ACM SoCC'13*, Santa Clara, Oct.2013

[2] Bitcurrent, Inc., Cloud Computing Performance Report, http://www.bitcurrent.com, 2010.

[3] Bondi, A. , "Characteristics of Scalability and their Impact on Performance", *Proc. of the 2nd Int'l Workshop on Software and Performance*, 2000, pp. 195-203.

[4] Chen, Y., Ganapathi, A., Griffith, R., & Katz, R. , "The Case for Evaluating MapReduce Performance using Workload suites. *IEEE Int'l Symp. on Modeling, Analysis & Simulation of Computer and Telecommunication Systems* (MASCOTS), 2011.

[5] CloudHarmony, "Benchmark Evaluation of 114 Public Clouds", Website: http://cloudharmony.com/clouds, 2014

[6] Cooper, B., Silberstein, A., Tam, E., Ramakrishnan, R., and Sears, R., "Benchmarking Cloud Serving Systems with YCSB", *Proc. of ACM Symp. on Cloud computing,* 2010.

[7] Farber M. and Kounev, S., "Existing Cloud Benchmark Efforts and Proposed Next Steps", *Slide Presentation,* Karlsruhe Institute for Technology (KIT), Aug.31, 2011.

[8] Ferdman, M., et al, "Clearing The Clouds: A Study of Emerging Scale-Out Workloads on Modern Hardware", *ACM 17th Int'l Conf. on Archi.l Support for Program. Lang. and OS* (ASPLOS), London, March 2012.

[9] Gao, J., Bai, X., and Tsai, W. T., "Cloud-Testing: Issues, Challenges, Needs and Practice", *Int'l Journal Software Engineering:* 2011

[10] Huang, S., et al., "The HiBench Benchmark Characterization of MapReduce- based Data Analysis", *Int'l Conf. on Data Eng. Workshops,* March 2010.

[11] Hwang, K., Fox, G. and Dongarra, J, *Distributed and Cloud Computing : From Parallel Processing to The Internet of Things*, Elsevier Publisher, 2012.

[12] Hwang, K. and Xu, Z. *Scalable Parallel Computing,* McGraw-Hill, San Francisco, 1998..

[13] Iosup, A., Ostermann, S., Yigitbasi, M., Prodan, R., Fahringer, T., and Epema, D. "Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing", *IEEE Trans. on Parallel and Distributed Systems*, 2011.

[14] Krebs, R. Momm, C. and Knounev, S., "Metrics and Techniques for Quantifying Performance Isolation in Cloud Environments, *ACM QoSA'12 Conf.* Bertino, Italy, 2012.

[15] Li, A., Yang, X., Kandula, S., and Zhang, M. , "CloudCmp: Comparing Public Cloud Providers", *Proc. of 10th Annual Conf. Internet Measurement*. Nov. 2010.

[16] Michael, M., et al, "Scale-up x Scale-out : A Case Study using Nutch/Luene", *IEEE Int'l Parallel and Distributed Processing Symp.*(IPDPS), March 26, 2007.

[17] Sharath, S. and Basu, A. "Performance of Eucalyptus and OpenStack Clouds on Future Grid", *Int'l J. of Computer Applications,* Oct. 2013,.

[18] Smith, W., "TCP-W: Benchmarking : An E-commerce Solution", *Intel Internal Report,* 2005.

[19] Sobel, W., Subramanyam, S., et al, "Cloudstone: Multi-platform, Multi-language Benchmark and Measurement Tools for Web 2.0", *Workshop on Cloud Comp.and App.,* Oct. 2008.

[20] Yigitbasi, N., Iosup, A., Epema, D., and Ostermann, S. " C-Meter: A Framework for Performance Analysis of Computing Clouds", *IEEE/ACM Proc. of 9th Int'l Symp. on Cluster Computing and the Grid,* (CCGrid). 2009.