International Journal of Grid and Distributed Computing
Vol. 10, No. 6 (2017), pp.29-40
http://dx.doi.org/10.14257/ijgdc.2017.10.6.03

MapReduce System Productivity Modeling and Measurement

Dongyu Feng[1,2], Ligu Zhu[1,2] and Lei Zhang[1,2]

[1] *School of Computer, Communication University of China, Beijing, China*
[2] *Beijng Key Laboratory of Big Data in Security & Protection Industry,*
Beijing, China
{fengdy, zhuligu, leizhang}@cuc.edu.cn

Abstract

How to effectively utilize the computing resource remains a longstanding challenge in MapReduce application, and MapReduce system productivity has become a major issue in research field. In the paper, we explored the productivity mathematical models for MapReduce system, defining the productivity as the ratio of the workload and energy consumption per unit time, and proposed the measurement approach for MapReduce system productivity. Based on the measurement model and approach, this paper selected CPU intensive and I/O intensive computing in MapReduce to measure and evaluate MapReduce system productivity. Finally, the measurement and evaluation result showed that productivity model and approach in the paper is reasonable and valid for architecture design of MapReduce system in production environment.

Keywords: *MapReduce system; productivity; measurement model; CPU intensive; I/O intensive*

1. Introduction

With the development of big data technology, MapReduce system is widely applied in big data application. According to a forecast by Forrester Research, the market value of MapReduce application will be 10 billion USD by 2020. However, behind huge economic benefit is the urgent energy consumption problem. The issue of MapReduce system productivity causes more and more concern in industry. Therefore, in current and future, how to effectively utilize the limited computing resource for maximum productivity, to reduce the operation cost is one of the most important issues in MapReduce system research[1].

However, in current the design of MapReduce system is lack of consideration for system productivity and the association between productivity and workload in the system. Thus, this paper explored the mathematical model of MapReduce system productivity, proposing the measurement model and measure approach for MapReduce system productivity. This paper selected CPU intensive and I/O intensive computing to measure and evaluate productivity of MapReduce systems with different hardware configuration, which proved the productivity measurement model and approach is reasonable and valid for architecture design of MapReduce system in production environment.

2. Background

Traditionally, performance was regarded as an important index in measurement for computer system. However, as the improvement of computer performance some new issues have arisen, performance has no longer been the only factor considered, energy consumption has also been noticed by researchers. Existing research held two different views on this point. Xu *et al.*, [3] considered that performance and energy consumption were two different optimization goals, there must be a compromise relationship between

ISSN: 2005-4262 IJGDC

them. Improving performance inevitably leads to high energy consumption, and reducing energy consumption will result in performance loss. Tsirogiannis *et al.*, [4] proposed that the optimization of energy consumption and performance were consistent. The most energy-efficient system is usually the best performance system. In terms of conception, the former accords with hardware design philosophy, considering that the more quickly operation need the more energy consumption. The latter accords with software design philosophy, considering that the more quickly operation need the less time consumption, resulting in the less energy consumption. These two opposite views are due to different evaluation methods. The former haven't considered of idle energy consumption and energy proportional computing[5]. The latter haven't considered the dynamic of computer power. No matter which view, productivity can well measure the relationship between performance and energy consumption.

In current high performance computing has developed into high productivity computing, and productivity has displaced performance as the composite index reflecting the quality of computer system[2].

According to the definition of productivity by DAPRA, productivity is the ratio of utility and cost[2], expressed as

$$P = U / C = U / (C_S + C_O + C_M) \tag{1}$$

In formula 1, P represents productivity, U represents utility and C represents cost. In details, utility is an index user defined, which generally includes performance, programmability and portability. Cost includes software cost (C_S), operation cost (C_O) and machine cost (C_M). Among these software and machine cost depend on application environment without universal, besides energy consumption is an important component in operation cost. Thus this paper only considers energy consumption as the operation cost of MapReduce system.

Productivity is valid workload completed with unit energy, so productivity includes two parts: valid workload and energy consumption. For energy consumption per unit time, the majority of researchers measured energy consumption with Joule, hence this paper still adopts Joule as energy measurement unit. Kumar et al. [3] proposed a series of energy measurement models based on simple power model[4-7] or directly measure energy consumption via execution time, without a universal definition of productivity and calculating and measure approach in details. This paper divides a MapReduce system into several components, and proposed energy consumption measurement characteristic and measure approach of each component.

The other part of productivity measurement is valid workload per unit time. How to measure valid workload is a difficult issue. Abdelsalam *et al.*, [8] simply defined a task, and took execution time of the task as workload. While this method was not suitable for MapReduce system processing complex task. If there is an atomic task unit, then it can measure workload by counting. Actually task unit can be designed based on algorithm, such as sort algorithm, it can define sorting per 10000 items as 1 unit task or a database transaction as 1 task unit. However, the measurement approach above is related to the complexity of algorithm without universal. Therefore, there is need to research an approach to measure workload by system bottom index.

It usually adopts floating point calculating or single-byte instruction as minimum task unit. Song Jie [9] presented the deficiency of this workload measurement method, proposing a new unit of workload, which defined the calculation volume per second of $1GHz$ CPU as 1 workload unit, recording as $1U$. Based on the approach above it is very convenient to measure workload, calculating the product of the frequency and usage of CPU, that is

valid workload = CPU frequency × CPU usage.

However, this measure approach only considered CPU calculation volume as valid workload, and ignored the calculation volume of disk and network I/O, leading to the productivity of CPU intensive task is higher than I/O intensive task, which is the disadvantage of this approach. Thus, valid workload can't be measured by single factor. It should adopt a variety of fine granularity performance index related to hardware to measure the productivity of MapReduce system rather than measure productivity of hardware. In aspect of software, though there are several kinds of task, there are much consistency in hardware measurements - CPU frequency, CPU usage, disk I/O and network I/O. However, how to collect these measurements is a difficult issue. First, workload should be measured once rather than repetition measurement. If workload includes reception and transmission of network, then there is no need to measure data processed by router since they both represent network I/O. Secondly, some hardware devices are used to optimize performance rather than for task defined by software. Therefore, it cannot simply add the calculation volume of all energy-consuming components to measure valid workload. Besides, it is difficult to unify measurements with different dimension, such as CPU frequency is GHz, disk I/O is kb/s and network I/O is bit/s. The measurement model and measure approach for MapReduce system productivity is further described and illustrated in the following section.

On the basis of modeling for MapReduce system productivity, this paper measures the productivity of MapReduce system and evaluates the productivity of MapReduce system with different hardware. We selects CPU intensive and I/O intensive computing in MapReduce as test cases, which are the most common computing types, can lead to the mixed workload of CPU, disk and network.

3. Modeling for MapReduce System Productivity

According to the universal definition of productivity, the productivity of computer system is workload per unit cost. In the paper we only consider energy consumption as cost. Therefore, productivity can be defined as the ratio of workload and energy consumption of the MapReduce system, expressed as:

$$\eta = \frac{L}{E} \tag{2}$$

In the following, energy consumption and workload of MapReduce system are modeled respectively.

3.1. Energy Consumption Model of MapReduce System

MapReduce system includes two parts: computer nodes and network devices. Thus, the energy consumption of MapReduce system can be defined as:

$$E(T) = E_c(T) + E_n(T) \tag{3}$$

In formula 3, $E_c(T)$ representes energy consumption of computers, and $E_n(T)$ representes energy consumption of network devices. This paper takes Joule as the unit of energy consumption.

3.2. Workload Model of MapReduce System

To reflect the characteristic of software's system clearly, the measurement of workload in the paper is based on application rather than hardware, and makes workload as a whole, which is beneficial to optimize productivity in software level. Selecting the performance parameters of hardware is aim to make the model more universal and independent of algorithm, ensuring measurement result can reflect the characteristic of algorithm.

Essentially any calculation includes two parts[10]:1.computing task; 2.I/O task. As for the former, computing task is completed by CPU, regardless of GPU in processing media data in the paper. Therefore, computing task could be measured by CPU index. For the latter, I/O task can be divided into local disk I/O and network I/O.

As the view of algorithm, reading and writing memory is aim to improve performance, and the number of disk I/O and network I/O operations will be affected by the number of memory I/O operations[11,12]. On the one hand, memory is a media device. In operating system, memory is the media between CPU and external storage such as disk and network to conciliate the speed contradiction between them[13]. All external storage data is read to memory, as well as cache at all levels, these read/write operations on memory can’t be counted in valid workload. On the other hand, operating system has made numerous performance optimization based on memory[14-16], while these read/write operations on memory aren’t the demand from software system or algorithm. In Linux system, memory are divided into actual memory, shared memory, cache and buffer[17]. Buffer is memory space for data to be written into disk, while cache is for data to be read from disk. In actually, as long as there is memory space available, it will be adequately used by buffer and cache. As consequence, the read/write on memory are not only operations for actual memory and shared memory even differ greatly, leading to the immeasurability of memory. Whereas, disk and network I/O can truly reflect the volume of input data, result data and intermediate data in computing task. In summary, in the paper we don’t adopt the read/write on memory into valid workload.

In theory, all input data is from disk and all output data are stored in disk, so disk I/O can be counted in valid workload. Even though the decreasing of network I/O will not affect local disk I/O, network I/O is the characteristic of distributed computing, determining the performance of computing to a great extent, and the quantity of network I/O is associated with algorithm, so network I/O is also counted in valid workload. For some data transmission device such bus and network device whose I/O operation don’t affect other I/O, so they are not counted in valid workload. Therefore, in the paper we select the working state of disk and network to measure I/O task. It defines $L(T)$ as workload during T time period:

$$L(T) = L_c(T) \oplus_1 [L_d(T) \oplus_2 L_n(T)] \tag{4}$$

In formula 4, $L_c(T)$ represents the calculation volume of CPU, $L_d(T)$ represents the I/O volume of disk and $L_n(T)$ represents the I/O volume of network, besides $\oplus_1$ and $\oplus_2$ are both aggregate function. In the following, we research the method to unify unit in the measurement of $L(T)$. $L_c(T)$ can be measured by the volume of floating point calculating. For any t time, then

$$L_c(T) = C_f(t) \times C_u(t) \times C_c \times C_{fn} \tag{5}$$

In formula 5, $C_f(t)$ represents the frequency of CPU at t time, $C_u(t)$ represents the usage of CPU at t time, C_c is the number of CPU cores, and $C_{fn}(t)$ is the times of floating point calculating. The objects of floating point calculating are two values with two machine word-length. So, formula 5 can be transformed to

$$L_c(T) = C_f(t) \times C_u(t) \times C_c \times C_{fn} \times 2C_{mw} \tag{6}$$

In formula 6, C_{mw} is machine word-length. Therefore, the unit of $L_c(T)$ is

$$1024M \times cycles / \sec \times 1 / cycles \times bit = 1024MB / 8 = 2^7 MB.$$

As above, we unify the unit of CPU calculation volume to MB, just as the unit of network and disk I/O.

For MapReduce system, considering the transmission of data in network, the data volume of sending and receiving should be consistent, which has already reflected in actual measurement.[18,19] Therefore, we only consider data sent or received in network in calculating workload.

3.3. Productivity Model of MapReduce System

As the definition of Productivity: during T time period, the productivity of MapReduce system $\eta(T)$ is the ratio of workload $L(T)$ and energy consumption $E(T)$, then

$$\eta(T) = \frac{L(T)}{E(T)} = \frac{L_c(T) \oplus_1 [L_d(T) \oplus_2 L_n(T)]}{E_c(T) + E_n(T)}; E(T) \neq 0 \tag{7}$$

4. Measuring Approach for MapReduce System Productivity

For a MapReduce system, supposing there are N computing nodes denoted by $c_i(1 \leq i \leq N)$ and M network devices denoted by $n_j(1 \leq j \leq M)$.

4.1. Measurement for MapReduce System Energy Consumption

As discussed above, the energy consumption of MapReduce system comes from two parts, computing nodes and network devices.

1) energy consumption of computing nodes

In MapReduce system, at t time the power of computing node c_i is $p_{c_i}(t)$, then

$$E_c(T) = \sum_{i=1}^{N} \int_T p_{c_i}(t)dt \tag{8}$$

There are several approaches to measure $E_c(T)$. All the computers have a power rating, while the actual power is dynamic change, thus it is difficult to find a mathematical expression for $p_{c_i}(t)$ and calculate $E_c(T)$ based on formula 8. In the paper it samples the active power of computing nodes every Δt period, and there are M times sampling from $0 \sim T$ time. As the definition of definite integral,

$$E_c(T) = \sum_{i=1}^{N} \int_0^T p_{c_i}(t)\mathrm{d}t \approx \sum_{i=1}^{N} \sum_{j=1}^{M} p_{c_i}(\Delta t \square j)\Delta t \tag{9}$$

When Δt is enough small, formula 9 can represent the value of $E_c(T)$. In order to simplify the description, supposing that at t time a measurement is $x(T)$, then

$$x(T) = \int_0^T x(t)\mathrm{d}t \approx \sum_{i=1}^{M} x(\Delta t \square j)\Delta t .$$

Thus, formula 9 can be simplified as

$$E_c(T) = \sum_{i=1}^{N} p_{c_i}(T) \tag{10}$$

2) energy consumption of network devices

The energy consumption of network devices include two parts, the energy consumption of switch or router $E_{rou}(T)$ and the energy consumption of cables $E_{cab}(T)$. Since the power of network devices has a small fluctuation with load changes, it can be considered static. Besides $E_{rou}(T)$ and $E_{cab}(T)$ are not significant related with whether there is data transmitted in network, therefore it is reasonable to adopt the product of power and time as energy consumption.

For the measurement of $E_{rou}(T)$, it selects active power rather than apparent power or rated power. In AC system, apparent power includes electrical current generated by inductor, capacitor and other circuit elements. The circulating power flow return to power at the end of each cycle, leading to net zero energy transfer. So this type of power that make no contribution to the useful work can't be included in $E_{rou}(T)$. Thus, the measurement of active power is the non-net zero power contributing to actual work. For a MapReduce system with M network devices, at t time the power of network device n_j is $p_{n_j}(t)$, then

$$E_n(T) = \sum_{j=1}^{M} \int_T p_{n_j}(t)dt \tag{11}$$

The electrical current in cables is high-frequency alternating current, so $E_{cab}(T)$ can be calculated as the power of alternating current. Take example for unshielded twisted pair, there are three AC-voltage U in twisted pair, -2.5V, 0 and 2.5V. The highest of transmission frequency f is $100\,MHz$, then the impedance $100\,MHz$ z is about $22\ \Omega$. The electrical current in twisted pair $I = U / z \approx 0.11\ \mathrm{A}$, so the power factor of circuit is $\cos\Phi = R / z \approx 0.43(l = 100\ m\ , R = 9.38\ \Omega)$. As the calculation formula of AC power, the average power of circuit $P = UI\cos\Phi = 0.11 \times 2.5 \times \cos\Phi \approx 0.12\ \mathrm{W}$. $E_{cab}(T)$ is so small that it can be ignored in the energy consumption of MapReduce system, therefore we can conclude that $E_n(T) \approx E_{rou}(T)$.

4.2. Measurement for MapReduce System Workload

In the following, we research the magnitude unification and measurement of $L_c(T)$, $L_d(T)$, and $L_n(T)$. Since the magnitude of $L_d(T)$ is consistent with $L_n(T)$, it can define $A \oplus_2 B = A + B$. However, the unit of CPU frequency is transformed from GHz to MB based on formula 6, leading to $L_c(T)$ is much larger than $L_d(T) + L_n(T)$. If $\oplus_1$ represents simple sum, which lead to $L_c(T)$ plays a decisive role in the result of $L(T)$, decreasing the impact of $L_d(T)$ and $L_n(T)$ in total workload. Thus, it is inappropriate to select simple sum as the aggregation function $\oplus_1$. To keep A and B in the same magnitude, it defines $A \oplus_1 B = \Phi A + B$, in which Φ is adjustment coefficient. So formula 4 can be turn to

$$L(T) = \Phi L_c(T) + [L_d(T) + L_n(T)] \tag{12}$$

As discuss in chapter 2, computing and I/O are equally important to algorithm. Thus, we define $\Phi = T_{IO} / T_{CPU}$. T_{IO} represents the maximum throughput of I/O including disk and network two parts. T_{CPU} represents the maximum throughput of CPU. Due to the difference of hardware between different computer systems, the values of Φ are different, to ensure the accordance between the impact of $L_c(T)$ and $L_d(T)+L_n(T)$ in $L(T)$.

Table 1 shows characteristic attributes of CPU, disk and network in the i computer node $(1 \le i \le N)$ to measure the workload of MapReduce system. Every Δt period sample characteristic attributes from each node.

Table 1. Characteristic Attributes for Workload Calculation

characteristic attribute	unit	description
$C_u^i(t)$	%	the usage of CPU
$C_f^i(t)$	*GHz*	the frequency of CPU
$D_r^i(t)$	*MB*	the output amount of disk
$D_w^i(t)$	*MB*	the input amount of disk
$N_r^i(t)$	*MB*	the output amount of network
$N_w^i(t)$	*MB*	the input amount of network

According to the characteristic attributes above, we define calculation formulas of $L_c(T)$, $L_d(T)$, and $L_n(T)$ as follows,

$$L_c(T)=\sum_{i=1}^{N} 2C_c C_{fn} C_{mw} \int_T C_f^i(t) C_u^i(t)dt \quad (13),$$

$$L_d(T)=\sum_{i=1}^{N} \int_T [D_r^i(t)+D_w^i(t)]dt \quad (14),$$

$$L_n(T)=\frac{1}{2}\sum_{i=1}^{N}[N_r^i(T)+N_w^i(T)]=\sum_{i=1}^{N} N_r^i(T)=\sum_{i=1}^{N}\int_T N_r^i(t)dt \quad (15).$$

4.3. Measurement Formula for MapReduce System Productivity

According to formulas 5-15, we define the productivity measurement formula for MapReduce system with N computer nodes and M network devices as:

$$\eta(T)=\frac{L(T)}{E(T)}=\frac{\Phi L_c(T)+[L_d(T)+L_n(T)]}{E_c(T)+E_n(T)}=\frac{\sum_{i=1}^{N} 2\Phi_i C_c C_{fn} C_{mw}\int_T C_f^i(t) C_u^i(t)dt\} + \sum_{i=1}^{N}\int_T [D_r^i(t)+D_w^i(t)]dt + \sum_{i=1}^{N}\int_T N_r^i(t)dt}{\sum_{i=1}^{N}\int_T p_{c_i}(t)dt + \sum_{j=1}^{M}\int_T p_{n_j}(t)dt]} \quad (16)$$

5. Measurement and Analysis for MapReduce System Productivity

To validate the measurement model and summarize the law of MapReduce system productivity, in the paper we measure the productivity of two MapReduce systems consisted of servers and PCs respectively.

5.1. Measurement Environment

We take HP DL380 as server equipped with Intel Xeon E5645 CPU, and Lenovo R4900d as PC equipped with Intel I3 3240 CPU. The scale of two MapReduce systems are both 8 nodes, deployed with Ganglia monitoring software. All nodes in each MapReduce system are connected by H3C switch, computer nodes and network devices are connected to the same power supply, and we take electric power meter to measure the total energy consumption. The topology of measurement environment is shown in Figure 1.

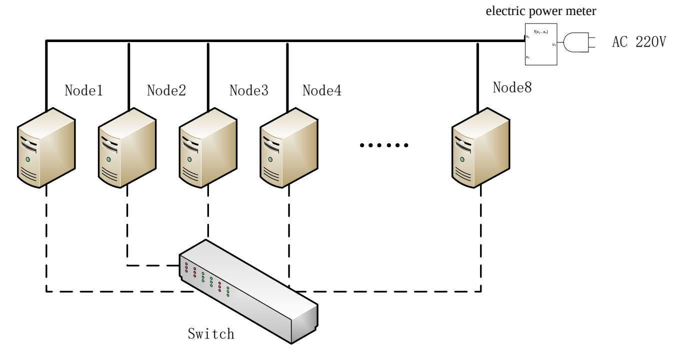


Figure 1. Topology Diagram of Measurement Environment

5.2. Index Measurement and Calculation for MapReduce System Productivity

This paper selects CPU intensive and I/O intensive task in MapReduce as test cases. CPU intensive task is consisted of massive computing jobs, and I/O intensive task is consisted of massive I/O jobs, which are the most common task types in MapReduce. We respectively select Wordcount and Sort as CPU intensive and I/O intensive task, and record the performance index by Ganglia monitoring software and energy consumption by electric power meter to calculate productivity according to the measurement model. All test cases are from the Benchmark for MapReduce, and the load of each case is $100\ MB$, $1\ GB$, $10\ GB$, $100\ GB$, $1\ TB$.

To measure the productivity of MapReduce system, it first need to calculate the coefficient Φ of each MapReduce system. Since computer nodes in MapReduce system are isomorphic, the Φ and characteristic attributes of each node are the same. The network bandwidth in measurement is $1000\ Mb/s$, whose the maximum theoretical transfer rate is $1000\ Mb/s$. The disk interface is SerialATA 3.0, whose the maximum theoretical transfer rate can reach $125\ MB/s$. Thus $T_{IO} = 725\ MB/s$. According to formula 5,

$$T_{Server-CPU} = 2.4 \times 100\% \times 6 \times 2 \times 64 \times 2^7\ MB/s = 235929.6MB/s$$
$$(C_f = 2.40GHz, C_u = 100\%, C_c = 6, C_{fn} = 2, C_{mw} = 64bit) \quad ,$$

$$T_{PC-CPU} = 1.8 \times 100\% \times 2 \times 1 \times 2 \times 64 \times 2^7\ MB/s = 58982.4MB/s$$
$$(C_f = 1.80GHz, C_u = 100\%, C_c = 2, C_{fn} = 1, C_{mw} = 64bit) \quad .$$

Thus, $\Phi_{Server} = 1.55 \times 10^{-3}$, $\Phi_{PC} = 1.22 \times 10^{-2}$.

According to the productivity measurement model, the productivity measurement result of two MapReduce system consisted of servers and PCs in computing intensive and I/O intensive cases are shown in Table 2.

Table 2. MapReduce System Productivity of Servers and PCs

		Servers	PCs
100MB	Wordcount	0.283	0.976
	Sort	0.351	1.223
1GB	Wordcount	0.305	1.081
	Sort	0.352	1.251
10GB	Wordcount	0.326	1.065
	Sort	0.368	1.286
100GB	Wordcount	0.331	0.906
	Sort	0.392	1.200
1TB	Wordcount	0.427	0.888
	Sort	0.45	1.138

5.3. Analysis of Measurement for MapReduce System Productivity

Wordcount is CPU intensive task, and Sort is I/O intensive task, which is not because one has more CPU calculation volume while the other has more I/O calculation volume, but because Wordcount has more I/O calculation than CPU calculation, and Sort has more I/O calculation than CPU calculation.

Based on further algorithm analysis: 1) under the same workload, the CPU calculation of Sort is larger than Wordcount's. Because the CPU calculation of Sort are mainly distributed among Hash calculation in map stage and sort calculation in reduce stage, while the CPU calculation of Wordcount are mainly distributed among Hash calculation in map stage and sum calculation in reduce stage, and sort is more complex than sum. 2) under the same workload, the I/O calculation of Sort is also larger than Wordcount's. Because there are lots of read operations to sort data in Sort's reduce stage and output data volume is equivalent with input, while in Wordcount's reduce stage there are just sum operations and the output data volume is much less than Sort's. Table 2 shows that under the same workload the productivity of Sort are always higher than Wordcount's, thus we can draw the following conclusions:

1) Sort is I/O intensive task, and the productivity measurement model proposed in this paper synthetically considers CPU and I/O calculation. We can conclude that there are a great number of CPU and I/O operations in Sort, thus the productivity is higher than Wordcount, reflecting that productivity measurement model accords with actual task characteristic.

2) In the case of small scale workload, both CPU and I/O calculation volume is small with a low resource utilization rate, leading to low productivity of both Wordcount and Sort. As the increasing of workload, the productivity of them improve to some extent, reflecting the reasonableness of productivity measurement model.

Based on the productivity measurement for two type MapReduce system, we compared and analyze the productivity of two MapReduce systems in the same task load. Considering execution time and energy consumption are the most concerned index for users, we respectively present the productivity, execution time and energy consumption of two MapReduce system in the same coordinate.

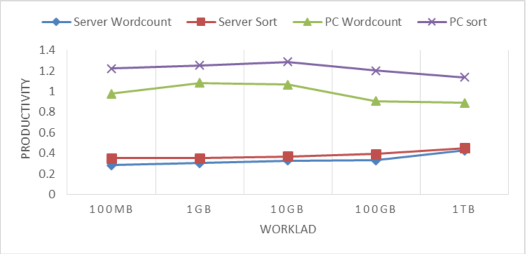


Figure 2. Productivity of MapReduce System

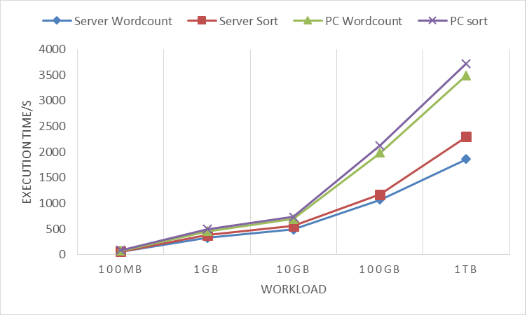


Figure 3. Execution Time of MapReduce System

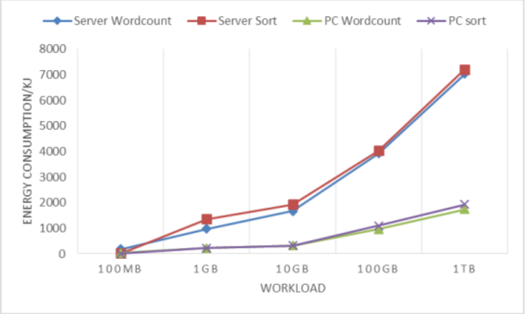


Figure 4. Energy Consumption of MapReduce System

As shown in Figure 2, the productivity of MapReduce system with servers is relatively stable within $1\,TB$ load, and reflects a rising tendency over $1\,TB$ load. While the productivity of MapReduce system with PCs reaches the maximum at $10\,GB$ load, and reflects a decline tendency with the increasing of load. As shown in Figure 3 and Figure 4, there is a positive correlation between the energy consumption and execution time of two MapReduce systems. The execution time of MapReduce system with PCs is always longer than servers'. When the load is less than $10\,GB$, the gap between the execution time of two MapReduce system is about 200s, no more than 4 minutes. Once the load is more than $10\,GB$ the gap between the execution time of two MapReduce system represents a rising tendency, can reach above 1000s, more than 15 minutes. Obviously under more than $10\,GB$ load, the execution time of MapReduce system with PCs is beyond user's tolerance compared to the MapReduce system with servers.

In summary, we can conclude that the MapReduce system composed of servers is suitable to process TB level large scale data, while the MapReduce system composed of PCs is suitable to process less than $10\,GB$ small scale data in batch processing application, which is not critical for response time, reflecting the design idea that MapReduce cluster is usually composed of cheap computers. All above reflect that the measurement result conform to actual requirement, proving that the measurement model and approach is reasonable and valid.

6. Conclusion

How to utilize the computing resources effectively to improve the productivity of MapReduce system has become an urgent issue in MapReduce system research [20]. In this paper we research the productivity of MapReduce system, proposing the measurement model and measure approach for MapReduce system productivity. The productivity measurement model can exactly define productivity. Productivity model admit the dynamic of computer power, and it defines productivity as the ratio of workload and energy consumption, which accord well with productivity optimization in software layer. This model can not only evaluate productivity via CPU, disk and network, but also can analyze energy consumption of computer and network device. All in all, we proposes a more comprehensive productivity model existing models, and analyze the productivity characteristic of different MapReduce task. Finally, we measure and evaluate the productivity of MapReduce system with different hardware configuration, validating the model for MapReduce system productivity is reasonable and instructive for architecture design of MapReduce system in production environment.

Acknowledgments

We would like to thank National Natural Science Foundation of China Project ‘Key techniques research of continuous data storage system energy-saving’(6137006).

References

[1] Cheng XQ, Jin XL, Wang YZ, GUO JF, Zhang TY, Li GJ. Survey on big data system and analytic technology. Ruan Jian Xue Bao/Journal of Software, 2014,25(9):1240 1252 (in Chinese). http://www.jos.org.cn/1000-9825/4674.htm

[2] H. W. Meuer. The Mannheim Supercomputer Statisties 1986 一 1992, TOP500 Report 1993, University of Mannheim, 1994, 1-15.

[3] Kumar K, Luyh. Cloud computing for mobile users: can offloading computation save energy?. IEEE Computer, 2010, 43(4):51-56.

[4] Orgerie A C, Lefevre L, Gelas J P. Save Watts in Your Grid: Green Strategies for Energy-Aware Framework in Large Scale Distributed Systems[J]. 2008:171-178.

[5] Younge A J, Von Laszewski G, Wang L, et al. Efficient resource management for Cloud computing environments[C]// International Conference on Green Computing. IEEE Computer Society, 2010:357-364.

[6] Gaggero M, Caviglione L. Predictive Control for Energy-Aware Consolidation in Cloud Datacenters[J]. IEEE Transactions on Control Systems Technology, 2015:1-1.

[7] Srikantaiah S, Kansal A, Zhao F. Energy aware consolidation for cloud computing[C]// Conference on Power Aware Computing and Systems. USENIX Association, 2008:10-10.

[8] Abdelsalam H S, Maly K,Mukkamala R, et al.Analysis of energy efficien in clouds. Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns Computation World. Norfolk: IEEE,2009:416-421.

[9] SONG Jie,LI Tian-tian,YAN Zhen-xing,et al.Energy-efficiency model and measuring approach for cloud computing[J].Journal of Software,2012,23(2):200-214.

[10] Hyndman D E. Computer System Architecture[M]. Tsinghua University Press, 1998.

[11] Olivier P, Boukhobza J, Senn E. Flashmon V2: Monitoring Raw NAND Flash Memory I/O Requests on Embedded Linux[J]. Acm Sigbed Review, 2013, 11(1):38-43.

[12] Qin X, Jiang H, Zhu Y, et al. Dynamic Load Balancing for I/O- and Memory-Intensive Workload in Clusters Using a Feedback Control Mechanism[J]. Lecture Notes in Computer Science, 2003, 2790:224-229.

[13] Ong L. Predictive memory caching for media-on-demand systems: US, US5815662[P]. 1998.

[14] Panda P R, Catthoor F, Dutt N D, et al. Data and memory optimization techniques for embedded systems[J]. Acm Transactions on Design Automation of Electronic Systems, 2001, 6(2):149-206.

[15] Neri F. Disturbed Exploitation compact Differential Evolution for limited memory optimization problems[J]. Information Sciences, 2011, 181(12):2469–2487.

[16] Qiu M, Ming Z, Li J, et al. Phase-Change Memory Optimization for Green Cloud with Genetic Algorithm[J]. IEEE Transactions on Computers, 2015, 64(12):1-1.

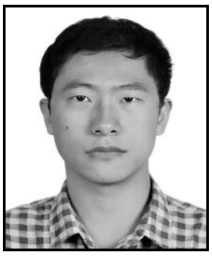
[17] Hill M D. Aspects of cache memory and instruction buffer performance[M]. University of California, Berkeley, 1987.

[18] Lin C. Heuristic Contention-Free Broadcast in Heterogeneous Networks of Workstations with Multiple Send and Receive Speeds[J]. Journal of Supercomputing, 2004, 30(30):37-64.

[19] Marsden P V. Network Data and Measurement[J]. Annual Review of Sociology, 2003, 16(4):435-463.

[20] Dai C, Ye Y, Liu T J, et al. Design of High Performance Cloud Storage Platform Based on Cheap PC Clusters Using MongoDB and Hadoop[J]. Applied Mechanics & Materials, 2013, 380-384:2050-2053.

Authors



Dongyu Feng was born in 1989, and received master degree in Communication University of China. He is a Ph.D candidate in School of Computer at Communication University of China. His research interests focus on distributed system and storage system.

Ligu Zhu was born in 1965, and received Ph.D degree in China University of Mining and Technology. He is a professor in School of Computer at Communication University of China. His research interests focus on computer system architecture and storage system.

Lei Zhang was born in 1982, and recieived Ph.D degree in Communication University of China. He is a lecturer in School of Computer at Communication University of China. His research interests focus on distributed system and storage system.