Contribution Resource Fairness with Time Constraints in Cloud Computing

Hongying Luo

College of Mathematics and Information Science, Qujing Normal University,

Qujing Yunnan China 655011

luohongy1982@163.com

Keywords: Contribution resource fairness; Cloud computing; Time constraints

Abstract. Fair resource is a key building block of any shared computing system. Recently fair division theory has emerged as a promising approach for the allocation of multiple computational resources among users. We have introduced Contribution Resource Fairness (CRF), a fair sharing model that generalizes max-min fairness to multiple resource types. CRF has lots of good properties, it satisfies DSI, EF,PE and SP. CRF think about the using mutual resources and allocating resources based on the contribution value to improve the overall effectiveness of the system. If users stay the system more longer, the resources which user contributes will be use more longer by other users and contribution value of her will be more bigger. So she can get more resources when she need by CRF. We construct CRF mechanisms that provably satisfy properties and experimental results can potentially improve the efficiency to the point where the system capacity is almost saturated.

Introduction

Cloud computing[9] has become a hot research applications and is a new computing model. The national institute of standards and technology to define the basic characteristics of cloud computing is on-demand self-service and rapid elasticity[10].To achieve these characteristics, the main use virtualization and its related technologies[4]. At any time, there are tens of thousands of clients concurrent running their high-performance computing applications(e.g., MapReduce[2], MPI, Spark[3], Dryad[5],Mesos[7],Choosy[1],Quincy[8]) on the shared computing system. Resource allocation is a key building block of any shared computer system. One of the most popular allocation policies proposed so far has been max-min fairness[1]. Dominant Resource Fairness(DRF)[6] is a generalization of max-min fairness for multiple resources.

The most-widely used concept of fairness is proportional sharing, which provides allocations to clients in proportion to client-specific weights reflecting their priority or importance. Adaptions of the classic algorithms for network bandwidth multiplexing have been proposed for providing proportional fairness for storage system. Extended proportional share schedulers which provide reservations and limit guarantees in addition to proportional allocation have also been proposed. The question of fair division of multiple resources in computer systems was raised in a fundamental paper by Ghodsi[6], who advocated a model called Dominant Resources Fairness(DRF) to guide the allocation. In this paper we propose a model called Contribution Fairness Allocation(CRF) based on Contribution. We present a new allocation policy to maximize system utilization while providing fairness in the allocations of the competing clients. We evaluate the performance of our method and result show that our method can tradeoff the efficiency and fairness.

Problem Definition

2.1 Dominant Resource Fairness(DRF)

DRF is an allocation policy for multiple resources that meets all four of the required properties. For every user, DRF computes the share of each resource allocated to that user. The maximum among all shares of a user is called that user's dominant share, and the resource corresponding to the dominant share is called the dominant resource. Different users may have different dominant

resources. DRF seeks to maximize the smallest dominant share in the system, then the second-smallest, and so on.

2.2 Basic Setting

Let $U = \{1,2,\cdots,N\}$ be the set of cloud users. For every user i, we normalize the resource demand vector to d_i, where d_{ir} is the fraction of resource r required by each task of user i over the entire system. For simplicity, we assume positive demands for all users, $d_{ir} > 0, \forall i \in U, r \in R$. Let x_i be the number of tasks processed on the server for user i. User i joins the system at time t^i. Each user i contributes k_i resources to a common pool of machines, her weight is k_i. $C_i(t)$ is the contribution to the system form time t^i to t. User at time t gets the resources which are $A_i(t) = \{A_{i1}, A_{i2}, \cdots, A_{im}\}$. $B_i(t)$ is the number of tasks request and $x_i(t)$ is the number of tasks actual of user i at time t.

$$I = \{x_1, x_2, \cdots, x_m\}, \quad \forall x_i \in I, \quad i \in U, \quad x_i(t) = B_i(t) < \frac{k_i}{d_i}. \tag{1}$$

C_i satisfies

$$\sum_{i \in I} c_i(t) = \sum_{i \in U - I} (x_i(t) - k_i), \quad c_i(t) \le k_i - B_i(t) \cdot d_i, \quad c_i(t) \ge 0, \quad c_i = \int_{t^i}^{t} c_i(t)dt. \tag{2}$$

$$A_i(t) = \sum_{j=1}^{m} A_{ij}(t) = \sum_{j=1}^{m} x_i(t) \cdot d_{ij}. \tag{3}$$

$$A_i = \int_{t^i}^{t} A_i(t)dt. \tag{4}$$

If user has greater contribution to the system, the greater the value of $C_i(t)$. (1) defines the set of I which have users who do not need to require more resources than their bring($x_i \cdot d_i \le k_i$). System can be distributed the unused resources to other users who need resources($x_i \cdot d_i > k_i$) and improve the overall efficiency.

2.3 Fairness-Efficiency Tradeoff

Fair allocation has been widely studied in economics and computer science[12,13]. Generally speaking, the notion of fairness may pertain to mechanisms like bargaining and their relationship to ethical issues. Fairness offers predictable isolation among users. It can ensure that in the server the user receives in system is at least 1/n of all resources. Efficiency servers as another important metric measuring. High resource utilization naturally translates into high throughput. This is pf particular importance to cloud computing. Both fairness and efficiency can be achieved at the same time in traditional single-resource. Before discussing the tradeoff between fairness and efficiency, we shall first clarify how the notion of fairness it to be defined, and how efficiency it to be measured quantitatively.

A strict DRF schedule allows each flow to receive the same dominant service, but DRF at all times may not be possible in practice. In addition to fairness, efficiency is another important concern for a multi-resource scheduling algorithm, but has received no significant attention before. Perhaps the most widely adopted efficiency measure is system throughput, whose conventional definition is the rate of completions[11]. While this performance metric is well defined for single-resource systems, extending its definition to multiple types of resources. We define the efficiency measure is resource utilization in this paper. It displays the utilization rates of all resources. We define the fairness is each user can get resources which less than their need based on their contribution to the system.

2.4 Contribute Resource Fairness

User joins the system with a few resources. If user i and user j have $C_i(t) \ge C_j(t)$, we think user i will get more resources than user j when user i and j need more resources than their bring according to intuition ($A_i(t) \ge A_j(t)$). If the resources of user bring made more contribution in

the system (the more users use) and when her need more resources, her can get more resources than others. The users with few tasks want stay the system by this mechanism in order to get more resources when they need. System can improve the overall effectiveness and attract users to join by this mechanism.

$$\max \quad x=\{x_1,x_2,\cdots,x_n\} \text{ , s.t.}$$

$$\begin{cases} x_i(t)=B_i(t), if\ B_i(t)\le \min\left\{\dfrac{k_i}{d_i}\right\} \\ eles\ \sum_{i=1}^{n} x_i(t)\cdot d_i \le \sum_{i=1}^{n} k_i, x_i(t)\ge \min\left\{\dfrac{k_i}{d_i}\right\} \end{cases} \tag{5}$$

$$\min \quad \sum_{i=1}^{n} k_i - \sum_{i=1}^{n} x_i(t)\cdot d_i \text{ , s.t.}$$

$$\begin{cases} \left[x_i-\min\left(\dfrac{k_{ij}}{d_{ij}}\right)\right]\cdot d_i \ge \left[x_k-\min\left(\dfrac{k_{kj}}{d_{kj}}\right)\right]\cdot d_j \\ c_i \ge c_k, B_i(t)\cdot d_i > k_i, B_k(t)\cdot d_k > k_k \end{cases} \tag{6}$$

User gets resources less than her need according to (5) to avoid waste of resources. The more contribute to the system, the more resources get. If user needs the amount of resources more than her bring and her will get over resources from the unused resources of others. Users get over resources based on contribution value. This allocation method is a combination optimization problem and is NP problem according to (2)(5)(6).

Attenuation mechanism periodically updates the user's contribution value in order to prevent the user with high contribution value to enjoy the resources and not contribute resources. We define the attenuation factor to update the user's contribution value in (7).

$$c_i = c_i(1-\alpha). \tag{7}$$

2.5 Fairness Properties

The following are important and desirable properties of a fairness:

(1)Sharing incentive(SI) [6]. Each user should be better off sharing the cluster. Each user should not be able to allocate more tasks in a cluster partition consisting of $\frac{1}{n}$ of all resources.

(2)Dynamic sharing incentive(DSI). Assume each user i contributes resources k_i to a common pool of machines, and that each can use at least the resources k_i ($B_i \cdot d_i > k_i$). If $B_i \cdot d_i \le k_i$, user can get the resources $B_i \cdot d_i$.

(3)Strategy-proofness (SP) [6]. Users should not be able to benefit by lying about their resource demands. This provides incentive compatibility, as a user cannot improve her allocation by lying.

(4)Pareto efficiency(PE) [6]. It should not be possible to increase the allocation of a user without decreasing the allocation of at least another user. This property is important as it leads to maximizing system utilization subject to satisfying the other properties.

(5)Envy-freeness(EF). A user should not prefer the allocation of another user. If user $B_i(t) > \min\left\{\dfrac{k_i}{d_i}\right\}$, so user i does not envy user j and user j satisfies

$\forall j \in U$, $A_j(t)-k_j \le A_i(t)-k_i$, $c_i \ge c_j$.

Theorem 1. CRF satisfies the DSI property

Proof. DRF satisfies SI, while CDRF does not satisfy SI. TDRF takes with time-related allocation

strategy, so user who stays longer gets more allocation resources. CRF is not satisfies the SI property. For example, there are (3) users. $U=\{u_1,u_2,u_3\}$, $c_1>c_2$, $d_1=\{0.1,0.2\}$, $d_2=\{0.2,0.1\}$, $d_3=\{0.1,0.1\}$, $k_1=k_2=k_3=\{0.5,0.5\}$, $B_1=B_2=7$, $B_3=1$. We have $x_1=5$, $x_1=4$ according to (5)(6) . So

$A_{11}=0.5\geq\sum_{i=1}^{3}k_{11}/3=0.5$, $A_{12}=1\geq\sum_{i=1}^{3}k_{11}/3=0.5$. But $A_{21}=0.8\geq\sum_{i=1}^{3}k_{11}/3=0.5$, $A_{22}=0.4\geq\sum_{i=1}^{3}k_{11}/3=0.5$.

So CRF is not satisfies the SI property. CRF satisfies the SI property by (5)(6).

Theorem 2. CRF satisfies SP property.

Proof. If user i get more resources at time t_1 by deceptive way, the others contribution value will increase who lend resource to user i. At time t_2, user i needs more resources with low contribution value and other users have high contribution value, user i may get few resource who has no advantage to compete with other users. For example, user i get more resources at time $t_1(A_i'>A_i\geq k_i)$, we assume $j,\exists j\in U, B_j(t)\cdot d_j<k_j$ and user i get over resources from user j. At time t_2, because $B_i(t_2)\cdot d_i>k_i$, $B_j(t_2)\cdot d_j>k_j$, and $c_j>c_i$, we have $A_j'(t_2)>A_i'(t_2)$, $A_i'(t_2)<A_i(t_2)$, $A_j'(t_2)>A_j(t_2)$, ccording to (2) (3) (5)(6). So user i gets less resource by deceptive way and it leads to that user j gets more resources. So CRF satisfies the SP property.

Theorem 3. CRF satisfies PE property.

Proof. The nature of the CRF is still DRF. DRF satisfies PE property. Assume user i can increase her dominant share, without decreasing the dominant share of anyone else. User i has at least one saturated resource. There are two cases. If no other user is using the saturated resource, then user i cannot increase her dominant share. If anyone is using the saturated resource, then user i increase her dominant share result in decreasing the allocation of at least one user sharing the same saturated resource. So user I cannot increase her dominant share or increase her dominant by decrease the allocation of other users who have the same dominant share.

Theorem 4. CRF satisfies EF property.

Proof. If user i satisfies $x_i(t)=B_i(t)$, so this satisfies demand of user i and user i will not envies resources of others

If user i satisfies $x_i(t)<B_i(t)$ and $\forall j\in U$, $A_j(t)-k_j>A_i(t)-k_i$, we have $c_j>c_i$ according (5). It represents the contribution of user j more the user i and user i has no reason to envy user j. If $\exists j\in U$, $A_j(t)-k_j>A_i(t)-k_i$, $c_j\leq c_i$, it contradicts with (5). We cannot have that user j gets resources more user i and contribution of user i more user j.

2.6 Online Algorithm Design

We present an online algorithm. When new user joins the system, her will priority gets resources to run task. We set up this mode, because new user joins the system with contribution value 0 and system prevents starvation phenomenon(because contribution value of other users bigger than her, lending to her cannot get enough resources to run any task).

ALGORITHM1 Contribute Resources Fairness pseudo-code

1: $K=(k_1,k_2,\cdots,k_n)$:each user i contributes k_i resources

2: $U=\{u_1,u_2,\cdots,u_k\}$: the set of users

3: $d=\{d_1,d_2,\cdots,d_n\}$: user i needs resources per task

4: $x=(x_1(t),x_2(t),\cdots,x_n(t))$number of tasks actual at time t

5: $c=(c_1,c_2,\cdots,c_n)$ contribution value of user i

6: while do

7: Pick user i who is new user or have the max c_i

8: If $\sum_{k=1}^{n} x_k(t) \cdot d_k + d_i \le \sum_{k=1}^{n} k_k$ and satisfies (5)(6)

9: Then

10: Update $x_i(t)$

11: Update c_j, $\forall j \in U$

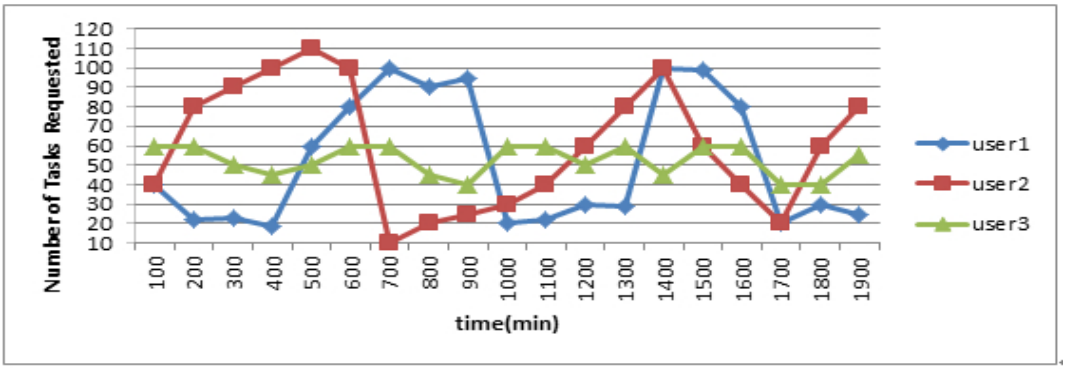
12: End If

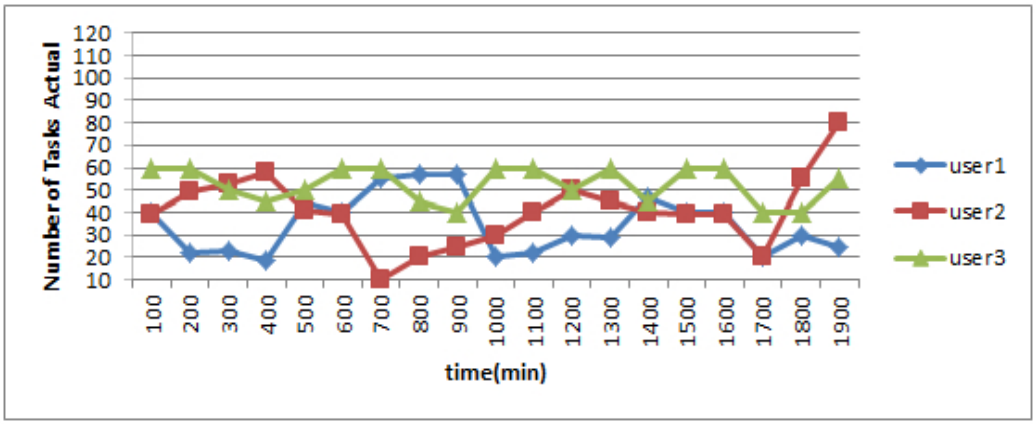
13: end while

Experimental Results

In this section, we evaluate the performance of CRF. The total resources are $R = (r_1, r_2)$ in the system. User 1 joins the system with $k_1 = (6,6)$. User 2 joins the system with $k_2 = (6,6)$. User 3 joins the system with $k_3 = (6,6)$, User 1 uses (0.1,0.2) per task who is a web application with heavy CPU and accessed by a large number of users sometime ,user 2 users (0.2,0.1) per task who is use high memory and regular uses resources, user 3 users (0.1,0.1) per task who is smooth uses resources. Figure 1 shows the number of tasks, contribution value, CPU and memory allocation given to each user as a function of time. The more contribution resources, the more resources value. If user need more resources to run more tasks, her will be get more resources based her higher contribution value by CRF. System encourages users to contribute unused resources and stay longer.

We next evaluate the resource utilization of the proposed TDRF algorithm. We take infinite users to join the system with different number of tasks requested. Fig.2 depicts the time series of resources utilization.





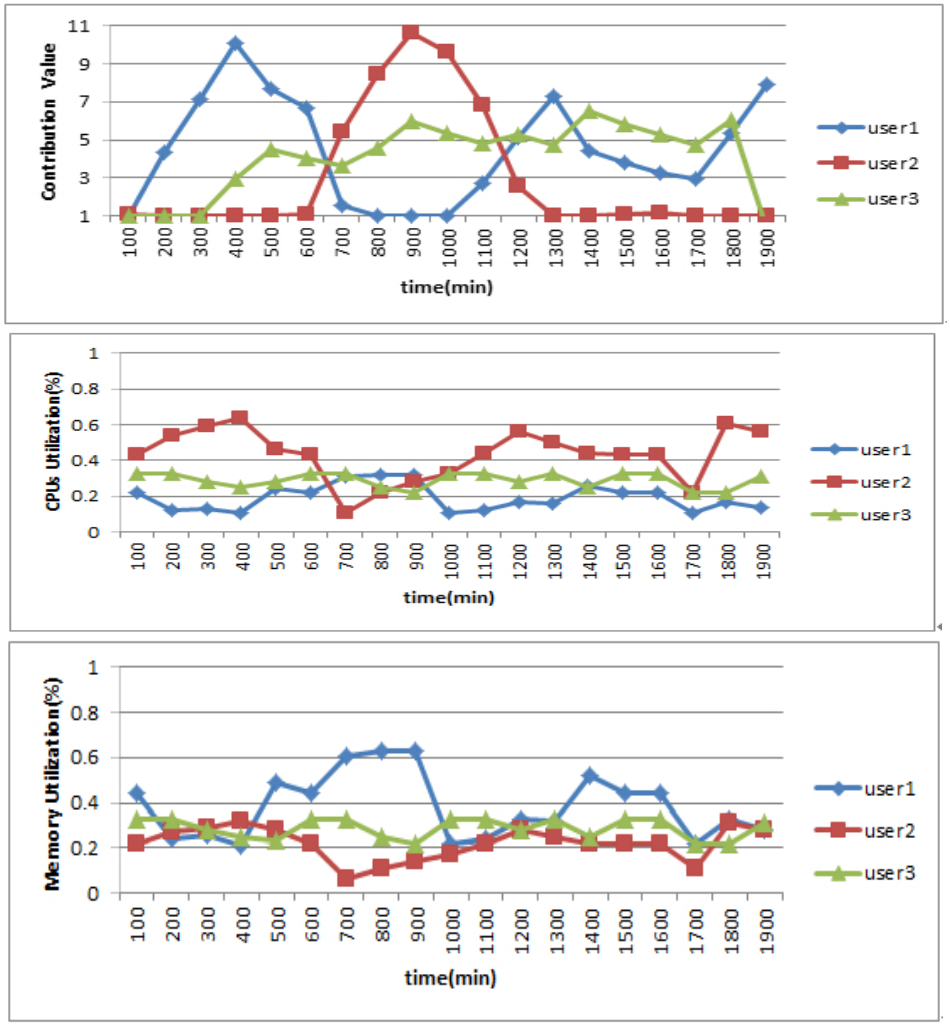


Fig. 1. Different resources and number of tasks for three users

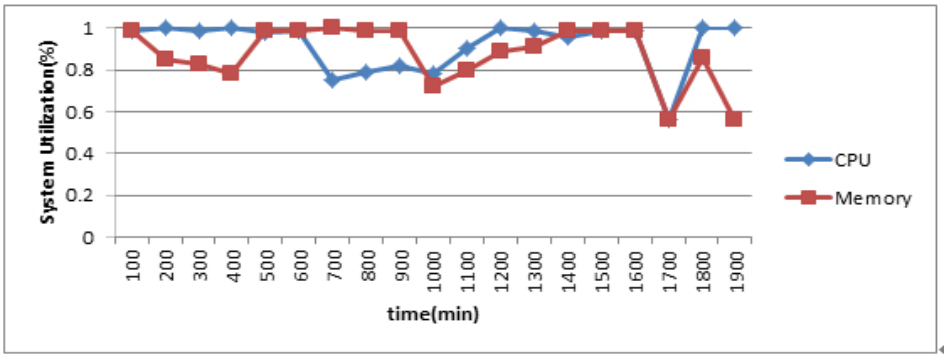


Fig .2. System Utilization

Conclusion and Future Work

In this paper, we study the problem which is how to measure the contribution of different users. We have introduced Contribution Resource Fairness (CRF), a fair sharing model that generalizes max-min fairness to multiple resource types. User can contribute unused resources to the others who need more resources. CRF think about the using mutual resources and allocating resources based on the contribution value to improve the overall effectiveness of the system. If users stay the system more longer, the resources which user contributes will be use more longer by other users and contribution value of her will be more bigger. So she can get more resources when she need by CRF.DRF satisfies SI , PE AND EF, but violates SP. CRF has lots of good properties, it satisfies DSI ,EF, PE and SP. As for future work, we use CRF in the real system (e.g., Hadoop, yarn) and

Dynamic adjustment contribution constraints.

Acknowledgements

The work was supported by Chinese Natural Science Foundation Grant No.11361048, Qujing Normal University Natural Science Foundation Grant No.2012QN016.

References

[1] A. Ghodsi, M, Zaharia, S.Shenker and I.Stoica. Choosy:Max-Min Fair Sharing for Datacenter Jobs with Constraints, EuroSys 2013,April (2013)

[2] J.Dean and S.Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. OSDI'04, (2004)

[3] M. Zaharia, M. Chowdhury, M.J.Franklin, S.Shenker, I.Stoica. Spark: Cluster Computing with Working Sets. HotCloud Vol.10,(2010),p.10

[4] R. Buyya,C. S.Yeo C S. Cloud computing and emerging IT platforms:vision,hype,and reality for delivering computing as the 5th utility[J]. Future Generation Computer System,Vol.25(2),(2009),p. 599-616.

[5] M. Isard, M. Budiu, Y. Yu, A. Birrell, and D.Fetterly, "Dryad: distributed data-parallel programs from sequential building blocks," in Proc. EuroSys,(2007)

[6] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica, "Dominant resource fairness: Fair allocation of multiple resource types," in Proc. USENIX NSDL,(2011)

[7] B. Hindman, A. Konwinski, M.Zahria, A. Ghodis, A.D.Joseph, R.Katz, S.Shenker and I.Stoica, Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center, NSDI 2011, March (2011)

[8] M. Isard, V. Prabhakaran, J. Currey, U. Wieder, K. Talwar, A. Goldberg. Quincy: Fair Scheduling for Distributed Computing Clusters, In SOSP, Vol.9, (2009),p.261-267

[9] M.Ambrust, A.Fox, R.Griffith, Above the Clouds: A Berkeley View of Cloud Computing[EB/OL].(2011-01-25).http://www.eecs.berkeley.edu/pubs/techrpts/2009/EECS-(2009)

[10] J. Mell, T. Grance. The NIST definition of cloud computing[R]. [S.I.]:National Institute of Standards and Techology, (2011)

[11] M.B.Harchol. Performance Modeling and Design of Computer System: Queueing Theory in Action. Cambridge University Press,(2013)

[12] S.J.Brams and A.D. Taylor, F.Division: From Cake-Cutting to Dispute Resolution. Cambidge University Press, Cambridge, UK,.(1996)

[13] A.V. Goldberg and J. Hartline, "Envy-free auctions for digital goods". In 4th ACM Conf. Electronic Commerce,(2003).p.29