Dominant Resource Fairness with Time Constraints in Cloud Computing

Chunyan Zhu

College of Computer Science and Engineering, Qujing Normal University,

Qujing Yunnan China 655011

chunyan8248@126.com

Keywords: Dominant resource fairness; Time constraints; Cloud computing

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. While in reality users are not all present in the system simultaneously. Dominant Resource Fairness(DRF) satisfies SI , PE AND EF, but violates SP. We have introduced Dominant Resource Fairness with Time Constraints (TDRF), a fair sharing model that generalizes max-min fairness to multiple resource types. TDRF has lots of good properties, it satisfies DSI, PE and SP. Users can exchange resources which are user don not need for resources which are user need by exchange mechanism. TDRF think about the time factor 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. We construct TDRF mechanisms that provably satisfy properties, and analyze the performance.

Introduction

Cloud computing[9] has become a hot research applications and is a new computing model which is the development of distributed computing, parallel computing and grid computing. 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.

Nevertheless, some aspects of realistic computing system are beyond the current scope of fair division theory. Indeed, it is typically not the case that all the users are present in the system at any given time; users may arrive and depart, and the system must be able to adjust the allocation of resources. For example, if one user arrives before another, the first user should intuitively have priority. What does fairness mean in this concept? We introduce the concepts that are necessary to answer this question, and design the mechanisms this satisfy our proposed desiderata.

The number of machines and applications running in cloud computing system is steadily increasing, leading to a diverse set of applications running over heterogeneous hardware. This has resulted in applications having constraints on the machines they can run on. For instance, a DNS server might need to run on machines that have a public IP address. While the nature of the constraints might vary, they can usually be classified into two categories: hard constraints and soft constraints. A job cannot run if its hard constraints are violated.

Problem Definition

2.1 Basic Setting

Let U={1,…,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, $k_i^{c_j}$ ($\forall i \in U, \forall c_j \in C$) is c_j type resource of user i contribute.

2.2 Exchange mechanism

User i contributes resources $k_i = (0.5,0.1,0.3)$ with $c_i = (c_2, c_3)$ and user j contributes resources $k_i = (0.1,0.8,0.3)$ with $c_j = (c_1, c_3)$. User i can change the resources which user i cannot use with user j. User i can get resources $k_i' = (0,0.6,0.3)$ and user j gets resources $k_j' = (c_1, c_2, c_3) = (0.6,0,0.3)$. So users increase the resources which are they can use. user i' resource c_2 increased to 0.6 and user j' resource c_1 increased to 0.6 by exchange mechanism. Algorithm 1 displays the exchange mechanism. Algorithm need traverse each user, so the complexity of the algorithm is $O(|U|)$.

2.3 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.

Consider a system with of 9CPUS, 9GB RAM, and two users, where user A runs tasks with demand vector<1CPU, 2GB>, and user B runs tasks with demand vector<3CPUs, 1GB> each. Each task from A consumers 1/9 of the total CPUs and 2/9 of the total memory, so user A's dominant resource is memory. Each task from B consumers 3/9 of the total CPUs and 1/9 of the total memory, so user B's dominant resource is CPU. Let x and y be the number of tasks allocated by DRF to users A and B, respectively. Then user A receives<x CPU, 2x GB>, and user B gets <3y CPU, y GB>. The dominant shares of users A and B are 2x/9=3y/9. The DRF allocation is then given by the solution to the following optimization problem:

Max(x,y)

Subject to x+3y≤9, 2x+y≤9, x/9=3y/9.

Solving this problem yields x=3 and y=2. Thus, user A gets<3 CPU,6 GB> and user B gets<6CPU,2GB>.

2.4 Dominant Resource Fairness with Time Constraints (TDRF)

DRF primary purpose is the fairness of dominant resources, but it think about the time length of the user stays in the system. Each user joins the system with contributing resources. If users stay the system more longer, the resources which user contributes will be use more longer by other users. So we must think about the time factor. We assume t^{init} is the initial time and $t^{\max}$ is the running time in the system. User i joins the system at time t^i. The TDRF allocation is then given by the solution to the following optimization problem:

Max(x,y)

Subject to x+3y≤9, 2x+y≤9, $\frac{2x}{9} \times \frac{t^A - t^{init}}{t^{max} - t^{init}} = \frac{3y}{9} \times \frac{t^B - t^{init}}{t^{max} - t^{init}}$

Solving this problem yields x=4 and y=1. Thus, user A gets<4 CPU,8 GB> and user B gets<3CPU,1GB>. User A who joins the system earlier than B can get more allocation resources in the TDRF then DRF. If users stay the system more longer, the resources which user contributes will be use more longer by other users. TDRF encourages users to stay longer in the system.

ALGORITHM2 CDRF pseudo-code

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

2: $U = (u_1, u_2, \cdots, u_n)$: the set of users

3: $k' = (k_1', k_2', \cdots, k_n')$: user i get k_i' by exchange mechanism

4: $R = (R_1, R_2, \cdots, R_n)$;the total resources in the system

5: $s = (s_1, s_2, \cdots, s_n)$: user i's dominant shares, initially 0

6: $t = \{t^1, t^2, \cdots, t^n\}$: user i joins the system at time t^i

7: $D = \{D_1, D_2, \cdots, D_n\}$: resources given to user i, initially 0

8: $C = \{c_1, c_2, \cdots, c_n\}$: consumed resources, initially 0

9: Run Algorithm 1

10: For each $\forall i \in U$

11: $D_i = D_i + k_i'$ →update consumed vector

12: $C = C + D_i$ →update i' allocation vector

13: $s_i = \max_{j=1}^{r} \{D_{i,j} / R_j\} \times \dfrac{t^i - t^{\min}}{t^{\max} - t^{\min}}$

12: end each

14: pick user i with lowest dominant share s_i

14: $J_i \leftarrow$ demand of user i' next task

15: if $C = C + J_i \le R$ then

16: $C = C + J_i$ →update consumed vector

17: $D_i = D_i + J_i$ →update i' allocation vector

18: $s_i = \max_{j=1}^{r} \{D_{i,j} / R_j\} \times \dfrac{t^i - t^{\min}}{t^{\max} - t^{\min}}$

19: else

20: return →the cluster is full

21: END

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' （$k_i' \le k_i$）.

（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.

Theorem 1. TRDF satisfies the DSI property

Proof. DRF satisfies SI, while CDRF does not satisfies SI. TDRF takes with time-related allocation strategy, so user who stays longer gets more allocation resources. An arbitrary user i gets resources k_i' by algorithm 1 which is the exchange mechanism. The system will give resources k_i' to user i in algorithm 2 which is TDRF, so TDRF satisfies the DSI property.

Theorem 2. TRDF satisfies SP property.

Proof. Let C_i be the set of machines that user i can actually use. Let assume user i lies about her constraints by claiming that she can use a set of machines C_i'. Let A_i and A_i' be the allocations of user i under constraints C_i and C_i'. If user i lies about her constraints, then

$C_i \neq C_i'$ and $A_i' \bigcap C_i \neq \phi$. Which means that user i would get no benefit by lying and may hurt other users, but won't get any benefit since she cannot use these machines.

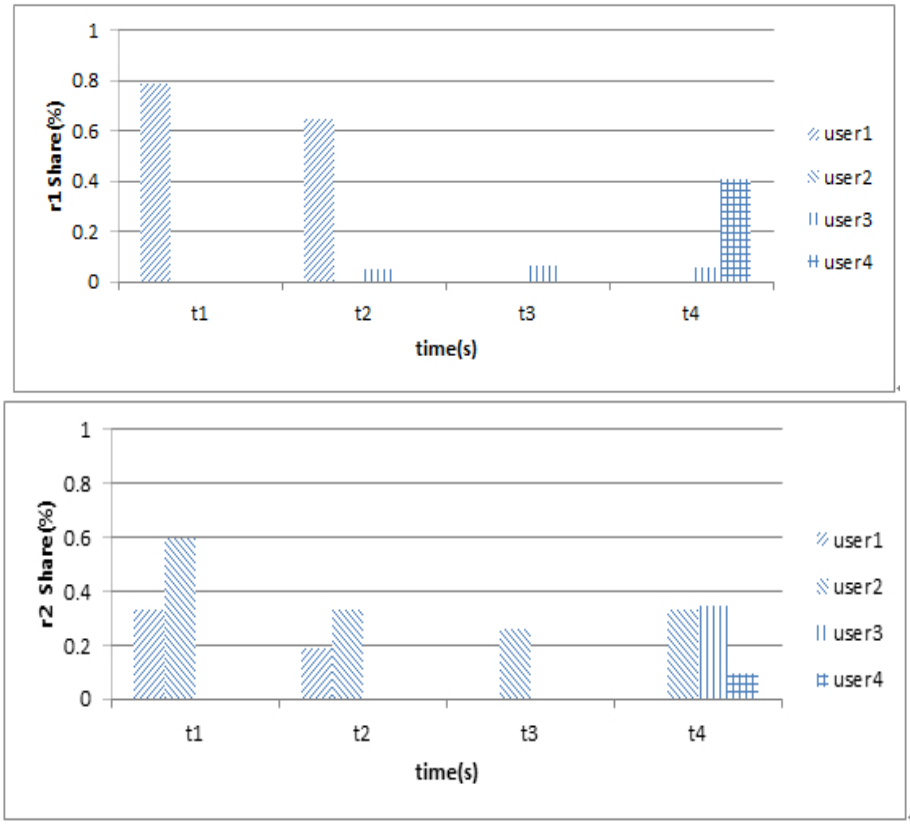
Theorem 3. TRDF satisfies PE property and TEF property.

Proof. The nature of the TDFR 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.

Experimental Results

In this section, we evaluate the performance of TDRF. The total resources are R=(r_1, r_2, r_3)=(1,1,1) in the system. User 1 joins the system at time t_1 and leaves at time t_3 with $k_1 = (0.5,0.2,0.4)$. User 2 joins the system at time t_1 with $k_2 = (0.4,0.3,0.5)$. User 3 joins the system at time t_2 with $k_3 = (0.1,0.6,0.2)$. User 4 joins the system at time t_4 with $k_2 = (0.2,0.2,0.5)$. User 1 uses (0.2,0.1,0) per task with $c_1 = \{r_1, r_2\}$, user 2 users (0,0.2,0.2) per task with $c_2 = \{r_2, r_3\}$, user 3 users (0.1,0,0.3) per task with $c_3 = \{r_1, r_3\}$ and user 4 users s(0.2,0.2,0) per task with $c_4 = \{r_1, r_2\}$. Figure 1 shows the r_1, r_2, r_3 and dominant resource allocation given to each user as a function of time.

User 1 and User 2 join the system at t_1. User 1 gets resources $K_1' = (0.9,0.2,0)$ and user 2 gets resources $K_2' = (0,0.3,0.9)$ by resources exchange mechanism at time t_1. User 1 leaves the system at time t_3 and user 3 joins the system at time t2. User 2 gets resources $K_2' = (0,0.3,0.9)$ and user 3 gets resources $K_3' = (0.1,0,0.5)$ by resources exchange mechanism at time t_3. When user 4 joins the system at time t4, user 4 gets resources $K_4' = (0.7,0.2,0)$.



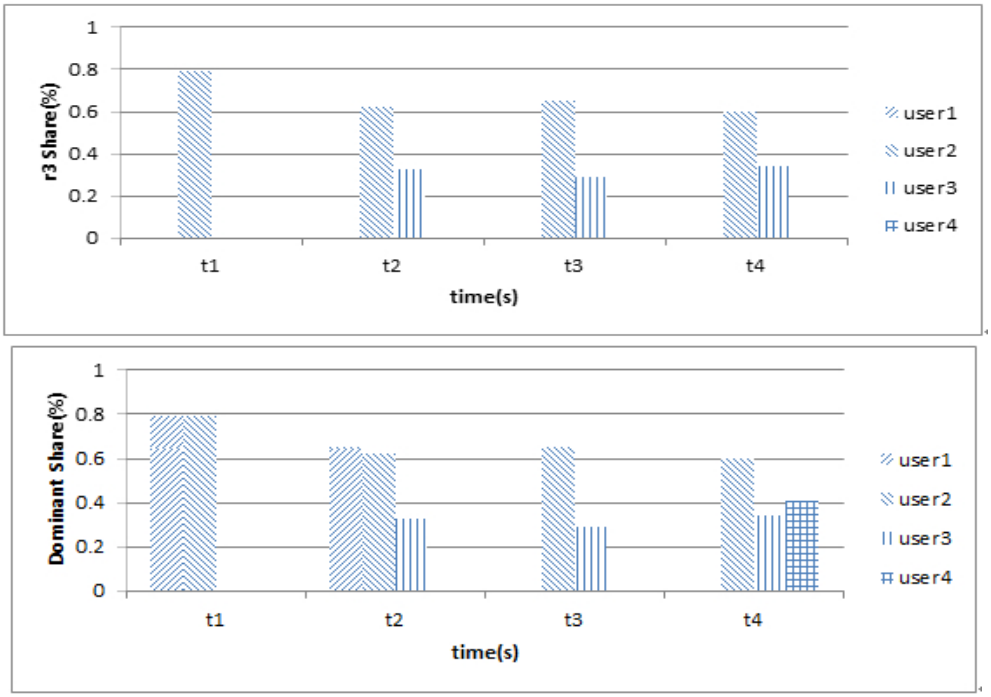


Fig. 1. Different resources and dominant share for four users

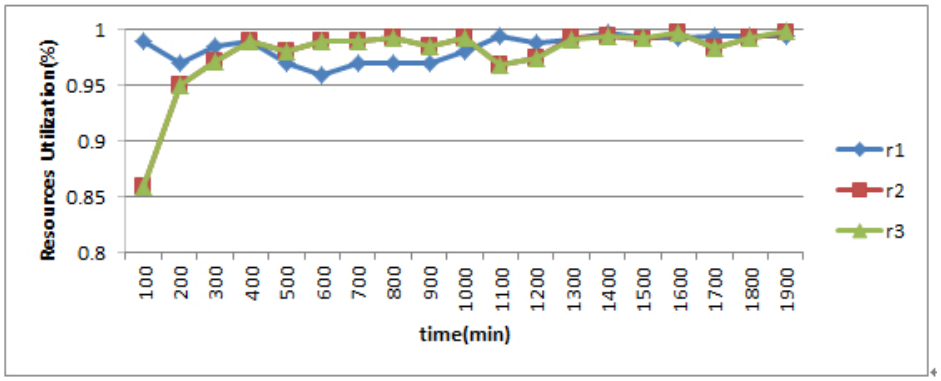


Fig .2. Resources share

Resource utilization: We next evaluate the resource utilization of the proposed TDRF algorithm. We take infinite users to join the system at different times. Fig.2 depicts the time series of resources utilization.

Conclusion and Future Work

In this paper, we study the problem which is users arrive at the different times. We have introduced Dominant Resource Fairness with Time Constraints(TDRF), a fair sharing model that generalizes max-min fairness to multiple resource types. Users can exchange resources which are user don not need for resources which are user need by exchange mechanism. TDRF think about the time factor 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. DRF satisfies SI , PE AND EF, but violates SP. TDRF has lots of good properties, it satisfies DSI , PE and SP. While in reality users are not all present in the system simultaneously, users which arrive at system at different times are allocated resources unfair. Any user is allocated fair resources in TDRF.

As for future work, we use TDRF in the real system (e.g., Hadoop, yarn) and Dynamic adjustment time constraints.

Acknowledgements

The work was supported by Chinese Natural Science Foundation Grant No.11361048,

References

[1] A.Ghodsi, M, Zaharia, S.Shenker and I.Stoica. EuroSys Vol. 23, (2013),p.78

[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. Cloud computing and emerging IT platforms:vision,hype,and reality for delivering computing as the 5th utility. Future Generation Computer System, Vol.25, (2009),p.599

[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)

[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

[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. [S.I.]:National Institute of Standards and Techology,(2011)