Fairness-Efficiency Scheduling for Cloud Computing with Soft Fairness Guarantees

Shanjiang Tang, Ce Yu, Yusen Li

Abstract—Fairness and efficiency are two important metrics for users in modern data center computing system. Due to the heterogeneous resource demands of CPU, memory, and network I/O for users' tasks, it cannot achieve the strict 100% fairness and the maximum efficiency at the same time. Existing fairness-efficiency schedulers (e.g., Tetris) can balance such a tradeoff elastically by relaxing fairness constraint for improved efficiency using the knob. However, their approaches are *unaware* of fairness degradation under different knob configurations, which makes several drawbacks. First, it cannot tell how much *relaxed* fairness can be guaranteed given a knob value. Second, it fails to meet several essential properties such as sharing incentive. To address these issues, we propose a new fairness-efficiency scheduler, *QKnober*, to balance the fairness and efficiency elastically and flexibly using a tunable fairness knob. QKnober is a *fairness-sensitive* scheduler that can maximize the system efficiency while guaranteeing the θ-soft fairness by modeling the whole allocation as a combination of *fairness-oriented* allocation and *efficiency-oriented* allocation. Moreover, QKnober satisfies fairness properties of sharing incentive, envy-freeness and pareto efficiency given a proper knob value. We have implemented QKnober in YARN and evaluated it using both testbed and simulated experiments. The results show that QKnober can achieve good performance and fairness.

Index Terms—Multi-Resource Allocation, Fairness, Efficiency, Hadoop

Appendix A
Proof of Theorem 1

Proof: For any two users $i, j \in [1, n]$, we have

$$\begin{aligned}
&|\frac{s_i}{w_i} - \frac{s_j}{w_j}| \leqslant \max_{1\leqslant i,j\leqslant n}\{|\frac{s_i}{w_i} - \frac{s_j}{w_j}|\}\\
&= \max_{1\leqslant i,j\leqslant n}\{|\frac{s_i^{max}\cdot\rho + s_i^{'}}{w_i} - \frac{s_j^{max}\cdot\rho + s_j^{'}}{w_j}|\}\\
&= \max_{1\leqslant i,j\leqslant n}\{|(\frac{s_i^{max}}{w_i} - \frac{s_j^{max}}{w_j})\cdot\rho + (\frac{s_i^{'}}{w_i} - \frac{s_j^{'}}{w_j})|\} \qquad (18)\\
&= \max_{1\leqslant i,j\leqslant n}\{|(\frac{s_i^{'}}{w_i} - \frac{s_j^{'}}{w_j})|\} = \max_{1\leqslant i\leqslant n}\frac{s_i^{'}}{w_i} - \min_{1\leqslant j\leqslant n}\frac{s_j^{'}}{w_j}\\
&= \max_{1\leqslant i\leqslant n}\{\frac{\mathbf{U}_i^{'}}{\mathbf{D}_i\cdot w_i}\cdot\max_{1\leqslant k\leqslant m}\{\frac{d_{i,k}}{r_k}\}\} - \min_{1\leqslant j\leqslant n}\{\frac{\mathbf{U}_j^{'}}{\mathbf{D}_j\cdot w_j}\cdot\max_{1\leqslant k\leqslant m}\{\frac{d_{j,k}}{r_k}\}\}.
\end{aligned}$$

Case $\#1$**:** When $\rho = 1$, we have $s_i^{'} = 0$ for $\forall i \in [1, n]$ discussed in Section 4.2.1 of the main file. In that case, it holds $|\frac{s_i}{w_i} - \frac{s_j}{w_j}| = 0$ for any user $i, j \in [1, n]$.

Case $\#2$**:** When $0 \leqslant \rho < 1$, according to the soft fairness definition, our proof turns to be seeking for an upper bound θ such that $\max_{1\leqslant i\leqslant n}\{\frac{\mathbf{U}_i^{'}}{\mathbf{D}_i\cdot w_i}\cdot\max_{1\leqslant k\leqslant n}\{\frac{d_{i,k}}{r_k}\}\} - \min_{1\leqslant j\leqslant n}\{\frac{\mathbf{U}_j^{'}}{\mathbf{D}_j\cdot w_j}\cdot\max_{1\leqslant k\leqslant n}\{\frac{d_{j,k}}{r_k}\}\} \leqslant \theta$ for all feasible allocations under the resource capacity vector $\mathbf{R}^{'}$.

In the efficiency-oriented resource allocation, a feasible allocation $\mathbf{U}^{'} = \langle\mathbf{U}_1^{'}, \cdots, \mathbf{U}_n^{'}\rangle$ ceases when at least one resource is fulfilled, i.e.,

$$\max_{1\leqslant k\leqslant m}\{\sum_{i=1}^{n} u_{i,k}^{'}/r_k^{'}\} = 1. \qquad (19)$$

Additionally, for all feasible allocations, the maximum value of $\mathbf{U}_i^{'}$ exists for user i when it possesses all the resource capacity vector $\mathbf{R}^{'}$ exclusively. In that case, all other users have no resource allocations, i.e., $\forall j \neq i \in [1, n], \mathbf{U}_j^{'} = 0$. We then get the maximum value of $\mathbf{U}_i^{'}$ according to Formula (5) and (19) as follows:

$$\mathbf{U}_i^{'} = \frac{\mathbf{D}_i}{\max_{1\leqslant k\leqslant m}\frac{d_{i,k}}{r_k - \rho\cdot\sum_{j=1}^{n} N_k(\mathbf{U}_k^{max})\cdot d_{j,k}}}. \qquad (20)$$

The upper bound θ of Formula (18) for all feasible allocations can then be computed, i.e.,

$$\begin{aligned}
\theta &= \max_{1\leqslant i\leqslant n}\{\frac{\mathbf{U}_i^{'}}{\mathbf{D}_i\cdot w_i}\cdot\max_{1\leqslant k\leqslant n}\{\frac{d_{i,k}}{r_k}\}\} - \min_{1\leqslant j\leqslant n}\{\frac{\mathbf{U}_j^{'}}{\mathbf{D}_j\cdot w_j}\cdot\max_{1\leqslant k\leqslant n}\{\frac{d_{j,k}}{r_k}\}\}\\
&= \max_{1\leqslant i\leqslant n}\{\frac{\mathbf{U}_i^{'}}{\mathbf{D}_i\cdot w_i}\cdot\max_{1\leqslant k\leqslant n}\{\frac{d_{i,k}}{r_k}\}\}\\
&= \max_{1\leqslant i\leqslant n}\{\frac{\max_{1\leqslant k\leqslant n} d_{i,k}/r_k}{\max_{1\leqslant k\leqslant m}\frac{w_i\cdot d_{i,k}}{r_k - \rho\cdot\sum_{j=1}^{n} N_k(\mathbf{U}_k^{max})\cdot d_{j,k}}}\}
\end{aligned}$$

□

Appendix B
Proof of Theorem 2

Proof: Let's start with the exclusively non-sharing case, where each user i schedules tasks under its own partition of the system resource, i.e., $\frac{w_i}{\sum_{j=1}^{n} w_j}\mathbf{R}$. In this case, the allocation stops when at least one resource is saturated, i.e., $\max_{1\leqslant k\leqslant m}\{d_{i,k}\cdot N_i(\overline{\mathbf{U}}_i)/(r_k\cdot\frac{w_i}{\sum_{j=1}^{n} w_j})\} = 1$ for each user i. Therefore,

$$N_i(\overline{\mathbf{U}}_i) = 1/\max_{1\leqslant k\leqslant m}\{d_{i,k}/(r_k\cdot\frac{w_i}{\sum_{j=1}^{n} w_j})\}.$$

- *S.J. Tang, C. Yu are with the College of Intelligence and Computing, Tianjin University, Tianjin 300072, China.*
E-mail: {tashj, yuce}@tju.edu.cn.
(Corresponding authors: Shanjiang Tang and Ce Yu.)
- *Yusen Lee is with the School of Computing, Nankai University, Tianjin 300071, China.*
E-mail: liyusen@nbjl.nankai.edu.cn.

Now we consider the sharing case for each user i. According to Formula (8), (14) and (11), we have

$$\begin{aligned} N_i(\mathbf{U}_i) &= \frac{s_i}{\max_{1\leqslant k\leqslant m}\{d_{i,k}/r_k\}} = \frac{s_i^{max}\rho + s_i^{'}}{\max_{1\leqslant k\leqslant m}\{d_{i,k}/r_k\}} \\ &= \frac{w_i/\sum_{j=1}^{n} w_j + s_i^{'}}{\max_{1\leqslant k\leqslant m}\{d_{i,k}/r_k\}} \geqslant N_i(\overline{\mathbf{U}}_i). \end{aligned}$$

□

Appendix C Proof of Theorem 3

Proof: By contradiction, let's assume that user i envies the allocation result of user j under QKnober allocation policy. Then for user i, it must have

$$N_i(\mathbf{U}_i) < N_i(\mathbf{U}_j). \tag{21}$$

We consider the following two cases:

Case 1: $\frac{\mathbf{D}_i}{|\mathbf{D}_i|} = \frac{\mathbf{D}_j}{|\mathbf{D}_j|}$: according to the fairness requirement of Formula (15), Formula (14) and (8), we have $\frac{s_i^{'}}{w_i} = \frac{s_j^{'}}{w_j} \Leftrightarrow \frac{s_i}{w_i} = \frac{s_j}{w_j} \Leftrightarrow \frac{N_i(\mathbf{U}_i)\cdot\max_{1\leqslant k\leqslant m}\frac{d_{i,k}}{r_k}}{w_i} = \frac{N_j(\mathbf{U}_j)\cdot\max_{1\leqslant k\leqslant m}\frac{d_{j,k}}{r_k}}{w_j}$. By exchanging the allocation between user i, j, subject to the fairness constraint of Formula (15), we should have $\frac{N_i(\mathbf{U}_j)\cdot\max_{1\leqslant k\leqslant m}\frac{d_{i,k}}{r_k}}{w_i} = \frac{N_j(\mathbf{U}_i)\cdot\max_{1\leqslant k\leqslant m}\frac{d_{j,k}}{r_k}}{w_j}$. Then it follows that $\frac{N_i(\mathbf{U}_i)\cdot\max_{1\leqslant k\leqslant m}\frac{d_{i,k}}{r_k}}{w_i} / \frac{N_i(\mathbf{U}_j)\cdot\max_{1\leqslant k\leqslant m}\frac{d_{i,k}}{r_k}}{w_i} = \frac{N_j(\mathbf{U}_j)\cdot\max_{1\leqslant k\leqslant m}\frac{d_{j,k}}{r_k}}{w_j} / \frac{N_j(\mathbf{U}_i)\cdot\max_{1\leqslant k\leqslant m}\frac{d_{j,k}}{r_k}}{w_j}$ i.e.,

$$N_i(\mathbf{U}_i)/N_i(\mathbf{U}_j) = N_j(\mathbf{U}_j)/N_j(\mathbf{U}_i). \tag{22}$$

Moreover, $\frac{\mathbf{D}_i}{|\mathbf{D}_i|} = \frac{\mathbf{D}_j}{|\mathbf{D}_j|} \Rightarrow \frac{d_{i,k}}{d_{j,k}} = \frac{|\mathbf{D}_i|}{|\mathbf{D}_j|}, \forall k \in [1, m]$. According to Formula (4), we have $N_i(\mathbf{U}_j) = \min_{1\leqslant k\leqslant m}\{u_{j,k}/d_{i,k}\} = N_j(\mathbf{U}_j)\cdot\min_{1\leqslant k\leqslant m}\{d_{j,k}/d_{i,k}\} = N_j(\mathbf{U}_j)\cdot|\mathbf{D}_j|/|\mathbf{D}_i|$. Hence,

$$N_i(\mathbf{U}_j)/N_j(\mathbf{U}_j) = |\mathbf{D}_j|/|\mathbf{D}_i|. \tag{23}$$

Similarly, we have

$$N_j(\mathbf{U}_i)/N_i(\mathbf{U}_i) = |\mathbf{D}_i|/|\mathbf{D}_j|. \tag{24}$$

According to Formula (22), (23) and (24), we have $\frac{N_i(\mathbf{U}_i)}{N_i(\mathbf{U}_j)} = \frac{N_i(\mathbf{U}_j)\cdot|\mathbf{D}_i|/|\mathbf{D}_j|}{N_i(\mathbf{U}_i)\cdot|\mathbf{D}_i|/|\mathbf{D}_j|} = \frac{N_i(\mathbf{U}_j)}{N_i(\mathbf{U}_i)} \Rightarrow N_i(\mathbf{U}_i) = N_i(\mathbf{U}_j)$, which contradicts the assumption of Formula (21).

Case 2: $\frac{\mathbf{D}_i}{|\mathbf{D}_i|} \neq \frac{\mathbf{D}_j}{|\mathbf{D}_j|}$: According to Formula (4), we have

$$N_i(\mathbf{U}_j) = \min_{1\leqslant k\leqslant m}\{\frac{u_{j,k}}{d_{i,k}}\} = N_j(\mathbf{U}_j)\cdot\min_{1\leqslant k\leqslant m}\{\frac{d_{j,k}}{d_{i,k}}\}. \tag{25}$$

and

$$N_j(\mathbf{U}_i) = \min_{1\leqslant k\leqslant m}\{\frac{u_{i,k}}{d_{j,k}}\} = N_i(\mathbf{U}_i)\cdot\min_{1\leqslant k\leqslant m}\{\frac{d_{i,k}}{d_{j,k}}\}. \tag{26}$$

According to Formula (6), there are

$$\begin{aligned} \epsilon_i(\mathbf{U}_j) &= N_i(\mathbf{U}_j)\cdot\sum_{k=1}^{m} d_{i,k}/r_k = N_j(\mathbf{U}_j)\cdot\min_{1\leqslant k\leqslant m}\{\frac{d_{j,k}}{d_{i,k}}\}\cdot\sum_{k=1}^{m} d_{i,k}/r_k \\ &= \epsilon_j(\mathbf{U}_j)\cdot\min_{1\leqslant k\leqslant m}\{\frac{d_{j,k}/r_k}{d_{i,k}/r_k}\}\cdot\frac{\sum_{k=1}^{m} d_{i,k}/r_k}{\sum_{k=1}^{m} d_{j,k}/r_k}. \end{aligned} \tag{27}$$

Moreover, since $\frac{\mathbf{D}_i}{|\mathbf{D}_i|} \neq \frac{\mathbf{D}_j}{|\mathbf{D}_j|}$, the following two conditions must hold:

$$\begin{aligned} i).\quad & \min_{1\leqslant k\leqslant m}\{\frac{d_{j,k}/r_k}{d_{i,k}/r_k}\} \leqslant \frac{d_{j,k'}/r_{k'}}{d_{i,k'}/r_{k'}} \\ & \Rightarrow \frac{d_{j,k'}}{r_{k'}} \geqslant \frac{d_{i,k'}}{r_{k'}}\cdot\min_{1\leqslant k\leqslant m}\{\frac{d_{j,k}/r_k}{d_{i,k}/r_k}\}, \forall k^{'} \in [1, m]. \end{aligned} \tag{28}$$

$$\begin{aligned} ii).\quad & \min_{1\leqslant k\leqslant m}\{\frac{d_{j,k}/r_k}{d_{i,k}/r_k}\} < \frac{d_{j,k_1}/r_{k'}}{d_{i,k_1}/r_{k_1}} \\ & \Rightarrow \frac{d_{j,k_1}}{r_{k_1}} > \frac{d_{i,k_1}}{r_{k_1}}\cdot\min_{1\leqslant k\leqslant m}\{\frac{d_{j,k}/r_k}{d_{i,k}/r_k}\}, \exists k_1 \in [1, m]. \end{aligned} \tag{29}$$

According to Formula (28) and (29), we have

$$\begin{aligned} & \min_{1\leqslant k\leqslant m}\{\frac{d_{j,k}/r_k}{d_{i,k}/r_k}\}\cdot\frac{\sum_{k=1}^{m} d_{i,k}/r_k}{\sum_{k=1}^{m} d_{j,k}/r_k} < \\ & \min_{1\leqslant k\leqslant m}\{\frac{d_{j,k}/r_k}{d_{i,k}/r_k}\}\cdot\frac{\sum_{k=1}^{m} d_{i,k}/r_k}{\min_{1\leqslant k\leqslant m}\{\frac{d_{j,k}/r_k}{d_{i,k}/r_k}\}\cdot\sum_{k=1}^{m} d_{i,k}/r_k} = 1. \end{aligned}$$

Hence, we have $\epsilon_i(\mathbf{U}_j) < \epsilon_j(\mathbf{U}_j)$ according to Formula (27). Similarly, it follows that $\epsilon_j(\mathbf{U}_i) < \epsilon_i(\mathbf{U}_i)$. Then after swapping the resource allocation, we have $\epsilon_i(\mathbf{U}_j) + \epsilon_j(\mathbf{U}_i) < \epsilon_i(\mathbf{U}_i) + \epsilon_j(\mathbf{U}_j)$, which violates the efficiency maximization requirement in Section 4.1 of the main file and hereby the assumption is false.

Finally, according to Case 1 and 2, we now can safely make the conclusion that QKnober policy is envy-freeness. □

Appendix D Proof of Theorem 4

Proof: We prove this theorem by contradiction to suppose that the resulting allocation $\mathbf{U} = \langle\mathbf{U}_1, \cdots, \mathbf{U}_n\rangle$ of QKnober is *not* pareto efficient. Then there must exist an alternative allocation $\breve{\mathbf{U}} = \langle\breve{\mathbf{U}}_1, \cdots, \breve{\mathbf{U}}_n\rangle$ such that $N_i(\mathbf{U}_i) \leqslant N_i(\breve{\mathbf{U}}_i)$ for $\forall i \in [1, m]$ and $\exists j \in [1, m], N_i(\mathbf{U}_j) < N_j(\breve{\mathbf{U}}_j)$. Similar to DRF mentioned in Section 4.1 of the main file, QKnober follows the progressive filling and the allocation terminates when at least one resource is saturated. It means that for the allocation $\mathbf{U}$ of QKnober, it holds

$$\max_{1\leqslant k\leqslant m}\{\sum_{i=1}^{n}\frac{u_{i,k}}{r_k}\} = \max_{1\leqslant k\leqslant m}\{\sum_{i=1}^{n}\frac{N_i(\mathbf{U}_i)\cdot d_{i,k}}{r_k}\} = 1.$$

Thus,

$$\max_{1\leqslant k\leqslant m}\{\sum_{i=1}^{n}\frac{N_i(\breve{\mathbf{U}}_i)\cdot d_{i,k}}{r_k}\} > \max_{1\leqslant k\leqslant m}\{\sum_{i=1}^{n}\frac{N_i(\mathbf{U}_i)\cdot d_{i,k}}{r_k}\} = 1.$$

which is not a feasible allocation and indicates that the assumption does not hold. Therefore, QKnober is pareto efficient. □

Appendix E Discrete Resource Allocation

In the previous section, we have implicitly assumed one *'super-computer'* with all big resources that can be allocated in arbitrarily small units. However, in practice, it is more likely to have a data center cluster consisting of many small computing nodes, which are allocated to tasks in discrete amounts. We refer to these two scenarios as the *continuous*, and the *discrete* scenario, respectively. We now take a look at how fairness is affected in the discrete scenario.

Consider a cluster consisting of K computing nodes, where the resource capacity of the i^{th} machine is $\mathbf{R}_i = \langle r_{i,1}, \cdots, r_{i,m}\rangle$ and $\mathbf{R} = \sum_{1\leqslant i\leqslant K}\mathbf{R}_i$. We assume that any task can be scheduled on every computing node. We further assume that each user has strictly positive demands. With these assumptions, we have the following conclusion,

Theorem 1: In the discrete scenario, QKnober is a θ-soft fairness policy where the difference between the allocations of any two users is bounded by

$$\theta = \begin{cases} \max_{1\leqslant i^{'}\leqslant n}\frac{\max_{1\leqslant k\leqslant m} d_{i^{'},k}/r_k}{w_{i^{'}}} + \\ \max_{1\leqslant i\leqslant n}\{\frac{\max_{1\leqslant k\leqslant n} d_{i,k}/r_k}{\max_{1\leqslant k\leqslant m}\frac{w_i\cdot d_{i,k}}{r_k - \rho\cdot\sum_{j=1}^{n}N_k(\mathbf{U}_k^{max})\cdot d_{j,k}}}\}, & (0\leqslant\rho<1). \\ \max_{1\leqslant i^{'}\leqslant n}\frac{\max_{1\leqslant k\leqslant m} d_{i^{'},k}/r_k}{w_{i^{'}}}. & (\rho=1) \end{cases}$$

Proof: In Section 4.2.1 of the main file, we have theoretically shown that QKnober can guarantee $\bar{s_i} = s_i^{max}\cdot\rho$ soft fairness in the fairness-oriented allocation under the *continuous* scenario by assuming the number of tasks can be *partial* value and one supercomputer containing all computing resources. However, in the *discrete* scenario, both task number and machine number are *discrete integer*, indicating that it is hard or even impossible to achieve the *exact* value of $s_i^{max}\cdot\rho$ soft fairness in its fairness-oriented allocation. Instead, we can seek to guarantee a soft fairness $\bar{s_i}$ in the fairness-oriented allocation under the discrete scenario satisfying that

$$s_i^{max}\cdot\rho \leqslant \bar{s_i} \leqslant s_i^{max}\cdot\rho + \max_{1\leqslant i^{'}\leqslant n}\{\max_{1\leqslant k\leqslant m} d_{i^{'},k}/r_k\}. \tag{30}$$

for each user i. Hence it holds,

$$0 \leqslant |\frac{\bar{s_i}}{w_i} - \frac{\bar{s_j}}{w_j}| \leqslant \max_{1\leqslant i^{'}\leqslant n}\frac{\max_{1\leqslant k\leqslant m} d_{i^{'},k}/r_k}{w_{i^{'}}}.$$

Case #1: When $\rho = 1$, the system performs pure fairness allocation only for the whole cluster resources. In that case, $s_i^{'} = 0$ for $\forall i \in [1, n]$. Then

$$\begin{aligned} |\frac{s_i}{w_i} - \frac{s_j}{w_j}| &\leqslant \max_{1\leqslant i,j\leqslant n}\{|\frac{s_i}{w_i} - \frac{s_j}{w_j}|\} \\ &\leqslant \max_{1\leqslant i,j\leqslant n}\{|(\frac{\bar{s_i}}{w_i} - \frac{\bar{s_j}}{w_j}) + (\frac{s_i^{'}}{w_i} - \frac{s_j^{'}}{w_j})|\} \\ &\leqslant \max_{1\leqslant i^{'}\leqslant n}\frac{\max_{1\leqslant k\leqslant m} d_{i^{'},k}/r_k}{w_{i^{'}}} \end{aligned}$$

Case #2: When $0 \leqslant \rho < 1$, for any two users $i, j \in [1, n]$ that

$$\begin{aligned} |\frac{s_i}{w_i} - \frac{s_j}{w_j}| &\leqslant \max_{1\leqslant i,j\leqslant n}\{|\frac{s_i}{w_i} - \frac{s_j}{w_j}|\} \\ &= \max_{1\leqslant i,j\leqslant n}\{|(\frac{\bar{s_i}}{w_i} - \frac{\bar{s_j}}{w_j}) + (\frac{s_i^{'}}{w_i} - \frac{s_j^{'}}{w_j})|\} \\ &\leqslant \max_{1\leqslant i,j\leqslant n}\{|\frac{\bar{s_i}}{w_i} - \frac{\bar{s_j}}{w_j}|\} + \max_{1\leqslant i,j\leqslant n}\{|\frac{s_i^{'}}{w_i} - \frac{s_j^{'}}{w_j}|\} \\ &\leqslant \max_{1\leqslant i^{'}\leqslant n}\frac{\max_{1\leqslant k\leqslant m} d_{i^{'},k}/r_k}{w_{i^{'}}} + \max_{1\leqslant i,j\leqslant n}\{|\frac{s_i^{'}}{w_i} - \frac{s_j^{'}}{w_j}|\} \\ &= \max_{1\leqslant i^{'}\leqslant n}\frac{\max_{1\leqslant k\leqslant m} d_{i^{'},k}/r_k}{w_{i^{'}}} + \max_{1\leqslant i\leqslant n}\frac{s_i^{'}}{w_i} - \min_{1\leqslant j\leqslant n}\frac{s_j^{'}}{w_j} \\ &= \max_{1\leqslant i^{'}\leqslant n}\frac{\max_{1\leqslant k\leqslant m} d_{i^{'},k}/r_k}{w_{i^{'}}} + \max_{1\leqslant i\leqslant n}\{\frac{\mathbf{U}_i^{'}}{\mathbf{D}_i\cdot w_i}\cdot\max_{1\leqslant k\leqslant m}\{\frac{d_{i,k}}{r_k}\}\} \\ &\quad - \min_{1\leqslant j\leqslant n}\{\frac{\mathbf{U}_j^{'}}{\mathbf{D}_j\cdot w_j}\cdot\max_{1\leqslant k\leqslant m}\{\frac{d_{j,k}}{r_k}\}\} \end{aligned} \tag{31}$$

In terms of the soft fairness definition, our proof aims to find an upper bound for Formula (31). Let $\eta = \max_{1\leqslant i\leqslant n}\{\frac{\mathbf{U}_i^{'}}{\mathbf{D}_i\cdot w_i}\cdot\max_{1\leqslant k\leqslant m}\{\frac{d_{i,k}}{r_k}\}\} - \min_{1\leqslant j\leqslant n}\{\frac{\mathbf{U}_j^{'}}{\mathbf{D}_j\cdot w_j}\cdot\max_{1\leqslant k\leqslant m}\{\frac{d_{j,k}}{r_k}\}$. It is equivalent to seeking for an upper bound for η in the efficiency-oriented allocation.

According to Formula (30), the upper bound of $\mathbf{R}^{'}$ idle resources for efficiency-oriented allocation occurs when $\bar{s_i} = s_i^{max}\cdot\rho$ given the total resource capacity $\mathbf{R}$. Moreover, in the efficiency allocation, the maximum value of η is achieved when a user i possesses all idle resources $\mathbf{R}^{'}$ and other users have no resource allocations (i.e., $\forall j \neq i \in [1, n], \mathbf{U}_j^{'} = 0$). In this case, we have $\eta = \max_{1\leqslant i\leqslant n}\{\frac{\mathbf{U}_i^{'}}{\mathbf{D}_i\cdot w_i}\cdot\max_{1\leqslant k\leqslant m}\{\frac{d_{i,k}}{r_k}\}\}$. Moreover, in the discrete scenario, the maximum value of η in the efficiency-oriented allocation cannot be larger than that in the continuous scenario. Thereby, we have

$$\mathbf{U}_i^{'} \leqslant \frac{\mathbf{D}_i}{\max_{1\leqslant k\leqslant m}\{\frac{d_{i,k}}{r_k^{'}}\}} = \frac{\mathbf{D}_i}{\max_{1\leqslant k\leqslant m}\frac{d_{i,k}}{r_k - \rho\cdot\sum_{j=1}^{n}N_k(\mathbf{U}_k^{max})\cdot d_{j,k}}}. \tag{32}$$

in the discrete scenario by modifying Formula (20). Then, it holds

$$\begin{aligned} \eta &= \max_{1\leqslant i\leqslant n}\{\frac{\mathbf{U}_i^{'}}{\mathbf{D}_i\cdot w_i}\cdot\max_{1\leqslant k\leqslant m}\{\frac{d_{i,k}}{r_k}\}\} \\ &\leqslant \max_{1\leqslant i\leqslant n}\{\frac{\max_{1\leqslant k\leqslant n} d_{i,k}/r_k}{\max_{1\leqslant k\leqslant m}\frac{w_i\cdot d_{i,k}}{r_k - \rho\cdot\sum_{j=1}^{n}N_k(\mathbf{U}_k^{max})\cdot d_{j,k}}}\} \end{aligned}$$

Therefore, the upper bound θ for Formula(31) is

$$\begin{aligned} |\frac{s_i}{w_i} - \frac{s_j}{w_j}| &\leqslant \max_{1\leqslant i^{'}\leqslant n}\frac{\max_{1\leqslant k\leqslant m} d_{i^{'},k}/r_k}{w_{i^{'}}} + \eta \\ &\leqslant \max_{1\leqslant i^{'}\leqslant n}\frac{\max_{1\leqslant k\leqslant m} d_{i^{'},k}/r_k}{w_{i^{'}}} + \\ &\quad \max_{1\leqslant i\leqslant n}\{\frac{\max_{1\leqslant k\leqslant n} d_{i,k}/r_k}{\max_{1\leqslant k\leqslant m}\frac{w_i\cdot d_{i,k}}{r_k - \rho\cdot\sum_{j=1}^{n}N_k(\mathbf{U}_k^{max})\cdot d_{j,k}}}\} \end{aligned}$$

□

Appendix F Overhead Evaluation

Recall from our system implementation in Section 6.2 that the task scheduling logic (e.g., fairness-oriented allocation plus

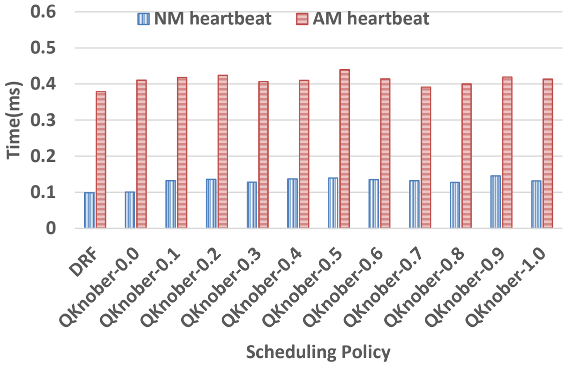

Fig. 1: The average processing time (overheads) to handle heartbeats from the NM and AM for DRF and QKnober.

efficiency-oriented allocation) of QKnober is more complicated than that in YARN. This section evaluates the computational overhead of QKnober with the aforementioned four workloads under different knob configurations. Specifically, we consider the time taken by the Resource Manager (RM) to deal with a *heartbeat* request from the Node Manager (NM) and from the Application Master (AM). In YARN, the RM performs the actual resource allocation during the NM heartbeat. AM is responsible for asking RM for resource allocation. At an AM ask heartbeat, the RM updates the accumulative asks from the AM and reacts with any tasks in the past asks that have been satisfied in the NM heartbeat.

Figure 1 presents the overhead results for DRF and QKnober to process heartbeats from the NM and AM under different knob values, respectively. It shows that the heartbeats processing time for both NM and AM is minor compared with the Hadoop workloads that often takes hours or days to complete [32]. Second, QKnober performs heartbeats a bit slower than DRF. This is because QKnober has more complex task scheduling mechanism than DRF. Third, for QKnober, the heartbeats processing time is much close under different knob configurations. It indicates that the knob configuration has no too much impact on the overhead contribution.

Acknowledgement

This work is sponsored by the National Natural Science Foundation of China (61972277) and Tianjin Natural Science Foundation (18JCZDJC30800). Ce Yu is supported by the Joint Research Fund in Astronomy (U1731243, U1931130) under cooperative agreement between the National Natural Science Foundation of China (NSFC) and Chinese Academy of Sciences (CAS).

References

[1] Apache hive performance benchmarks. In *https://issues.apache.org/jira/browse/HIVE-396*, 2009.

[2] Apache tpc-h benchmark on hive. In *https://issues.apache.org/jira/browse/HIVE-600*, 2009.

[3] Facebook workload traces. In *https://github.com/SWIMProjectUCB/SWIM/wiki/Workloads-repository.*, 2009.

[4] Google cluster data. In *https://code.google.com/p/googleclusterdata/*, 2011.

[5] Glpk (gnu linear programming kit). In *https://www.gnu.org/software/glpk/*, 2012.

[6] Puma datasets. In *http://web.ics.purdue.edu/f̃ahmad/datasets.htm.*, 2012.

[7] Faraz Ahmad, Seyong Lee, Mithuna Thottethodi, and T. N. Vijaykumar. Puma: Purdue mapreduce benchmarks suite. In *ECE Technical Reports*, 2012.

[8] Arka A. Bhattacharya, David Culler, Eric Friedman, Ali Ghodsi, Scott Shenker, and Ion Stoica. Hierarchical scheduling for diverse datacenter workloads. In *Proceedings of the 4th Annual Symposium on Cloud Computing*, SOCC '13, pages 4:1–4:15, New York, NY, USA, 2013. ACM.

[9] R. P. Brent. Efficient implementation of the first-fit strategy for dynamic storage allocation. *ACM Trans. Program. Lang. Syst.*, 11(3):388–403, July 1989.

[10] Paul C Chu and John E Beasley. A genetic algorithm for the multidimensional knapsack problem. *Journal of heuristics*, 4(1):63–86, 1998.

[11] E. Danna, S. Mandal, and A. Singh. A practical algorithm for balancing the max-min fairness and throughput objectives in traffic engineering. In *INFOCOM'12*, pages 846–854, March 2012.

[12] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. In *OSDI'04*, pages 10–10, Berkeley, CA, USA, 2004. USENIX Association.

[13] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008.

[14] Christina Delimitrou and Christos Kozyrakis. Quasar: Resource-efficient and qos-aware cluster management. In *ASPLOS '14*, pages 127–144, New York, NY, USA, 2014. ACM.

[15] Danny Dolev, Dror G. Feitelson, Joseph Y. Halpern, Raz Kupferman, and Nathan Linial. No justified complaints: On fair sharing of multiple resources. In *ITCS '12*, pages 68–75, NY, USA, 2012. ACM.

[16] Lars George. *HBase: the definitive guide*. " O'Reilly Media, Inc.", 2011.

[17] Ali Ghodsi, Vyas Sekar, Matei Zaharia, and Ion Stoica. Multi-resource fair queueing for packet processing. In *SIGCOMM '12*, pages 1–12, NY, USA, 2012. ACM.

[18] Ali Ghodsi, Matei Zaharia, Benjamin Hindman, Andy Konwinski, Scott Shenker, and Ion Stoica. Dominant resource fairness: Fair allocation of multiple resource types. In *NSDI'11*, pages 323–336, Berkeley, CA, USA, 2011. USENIX Association.

[19] Robert Grandl, Ganesh Ananthanarayanan, Srikanth Kandula, Sriram Rao, and Aditya Akella. Multi-resource packing for cluster schedulers. In *SIGCOMM'14*, pages 455–466. ACM, 2014.

[20] Benjamin Hindman, Andy Konwinski, Matei Zaharia, Ali Ghodsi, Anthony D. Joseph, Randy Katz, Scott Shenker, and Ion Stoica. Mesos: A platform for fine-grained resource sharing in the data center. In *NSDI'11*, pages 295–308, Berkeley, CA, USA, 2011. USENIX Association.

[21] Carlee Joe-Wong, Soumya Sen, Tian Lan, and Mung Chiang. Multiresource allocation: Fairness-efficiency tradeoffs in a unifying framework. *IEEE/ACM Trans. Netw.*, 21(6):1785–1798, December 2013.

[22] Ian Kash, Ariel D. Procaccia, and Nisarg Shah. No agent left behind: Dynamic fair division of multiple resources. In *AAMAS '13*, pages 351–358, 2013.

[23] Haikun Liu and Bingsheng He. Reciprocal resource fairness: Towards cooperative multiple-resource fair sharing in iaas clouds. In *SC '14*, pages 970–981, Piscataway, NJ, USA, 2014. IEEE Press.

[24] David C. Parkes, Ariel D. Procaccia, and Nisarg Shah. Beyond dominant resource fairness: Extensions, limitations, and indivisibilities. *ACM Trans. Econ. Comput.*, 3(1):3:1–3:22, March 2015.

[25] Charles Reiss, Alexey Tumanov, Gregory R. Ganger, Randy H. Katz, and Michael A. Kozuch. Heterogeneity and dynamicity of clouds at scale: Google trace analysis. In *SoCC '12*, pages 7:1–7:13. ACM, 2012.

[26] S. Tang, Z. Niu, B. He, B. Lee, and C. Yu. Long-term multi-resource fairness for pay-as-you use computing systems. *IEEE Transactions on Parallel and Distributed Systems*, 29(5):1147–1160, 2018.

[27] Shanjiang Tang, Qifei Chai, Ce Yu, Yusen Li, and Chao Sun. Balancing fairness and efficiency for cache sharing in semi-external memory system. 2020.

[28] Shanjiang Tang, BingSheng He, Shuhao Zhang, and Zhaojie Niu. Elastic multi-resource fairness: Balancing fairness and efficiency in coupled cpu-gpu architectures. In *SC '16*, pages 75:1–75:12. IEEE Press, 2016.

[29] Shanjiang Tang, Ce Yu, Chao Sun, Jian Xiao, and Yinglong Li. Qknober: A knob-based fairness-efficiency scheduler for cloud computing with qos guarantees. In Claus Pahl, Maja Vukovic, Jianwei Yin, and Qi Yu, editors, *Service-Oriented Computing*, pages 837–853, Cham, 2018.

[30] A. Thusoo, J.S. Sarma, N. Jain, Zheng Shao, P. Chakka, Ning Zhang, S. Antony, Hao Liu, and R. Murthy. Hive - a petabyte scale data warehouse using hadoop. In *ICDE'10*, pages 996–1005, March 2010.

[31] Hal R Varian. Equity, envy, and efficiency. *Journal of economic theory*, 9(1):63–91, 1974.

[32] Vinod Kumar Vavilapalli, Arun C. Murthy, Chris Douglas, Sharad Agarwal, Mahadev Konar, Robert Evans, Thomas Graves, and Lowe. Apache hadoop yarn: Yet another resource negotiator. In *SOCC '13*, pages 5:1–5:16, New York, NY, USA, 2013. ACM.

[33] Hui Wang and Peter Varman. Balancing fairness and efficiency in tiered storage systems with bottleneck-aware allocation. In *FAST'14*, pages 229–242, Berkeley, CA, USA, 2014. USENIX Association.

[34] Wei Wang, Chen Feng, Baochun Li, and Ben Liang. On the fairness-efficiency tradeoff for packet processing with multiple resources. In *CoNEXT '14*, pages 235–248, New York, NY, USA, 2014. ACM.

[35] Wei Wang, Baochun Li, and Ben Liang. Dominant resource fairness in cloud computing systems with heterogeneous servers. In *INFOCOM, 2014 Proceedings IEEE*, pages 583–591, April 2014.

[36] Wei Wang, Baochun Li, Ben Liang, and Jun Li. Multi-resource fair sharing for datacenter jobs with placement constraints. In *SC '16*, pages 86:1–86:12, Piscataway, NJ, USA, 2016. IEEE Press.

[37] Wei Wang, Shiyao Ma, Bo Li, and Baochun Li. Coflex: Navigating the fairness-efficiency tradeoff for coflow scheduling. In *INFOCOM'17*.

[38] Matei Zaharia, Dhruba Borthakur, Joydeep Sen Sarma, Khaled Elmeleegy, Scott Shenker, and Ion Stoica. Delay scheduling: A simple technique for achieving locality and fairness in cluster scheduling. In *EuroSys'10*, pages 265–278, New York, NY, USA, 2010. ACM.

[39] Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. Spark: cluster computing with working sets. In *Hot Cloud'10*, volume 10, page 10, 2010.

[40] Seyed Majid Zahedi and Benjamin C. Lee. Ref: Resource elasticity fairness with sharing incentives for multiprocessors. In *ASPLOS '14*, pages 145–160. ACM, 2014.