Fair scheduling of multiple resource types over multiple and heterogeneous resource pools*

G. Kesidis, Y. Wang, B. Urgaonkar
School of EECS, PSU, State College, PA
{gik2,ymw5093,buu1}@psu.edu

J. Khamse-Ashari and I. Lambadaris
SCE Dept, Carleton University, Ottawa, Canada
{jalalkhamseashari,ioannis}@sce.carleton.ca

September 8, 2016

CSE Dept., PSU, Technical Report, CSE-16-009
http://www.cse.psu.edu/research/publications/tech-reports/2016/CSE-16-009.pdf

Abstract

We consider the problem of scheduling a group of heterogeneous, distributed processes to a group of heterogeneous resource pools (physical servers, clusters of processors/cores) each with different types of resources. The processes specify their preference-constraints [20] and linearly elastic resource needs for heterogeneous resource pools. Our setting is a neutral, public cloud datacenter [6]. We extend the results of [12] on proportional and Max-Min Fairness (MMF) to this problem setting. A compromise is sought between a kind of proportional fairness across all resource pools, which may lead to not fully booked resources, and weighted Dominant Resource Fairness (DRF) [5] within each pool. We also give example models of subadditive elastic demand.

1 Introduction

Various parties in the cloud-computing ecosystem offer distributed container orchestration frameworks, including scheduling services (as *e.g.*, via YARN), targeting private cloud, public cloud, or hybrid: *e.g.*, Mesos, Google's Kubernetes, VMWare's vSphere, Amazon's Elastic Container Service, IBM's

*This research was supported in part by NSF CNS 1526133, DARPA XD3, and AFOSR DDDAS grants.

Bluemix Container Service, Microsoft's Azure Container Service. These platforms improve usability and reduce costs. They can also be used for scheduling distributed workloads. Based on extensive prior work on scheduling and load balancing, one can pose frameworks that:

- are proactive or reactive or have elements of both,
- involve task scheduling, longer-term resource scheduling, or have elements of both,
- are arranged in a hierarchical structure with a centralized task dispatcher/router at the root of the hierarchy.

Suppose an informed central dispatcher divides a process's jobs (as they are spawned) into tasks/threads and assigns/schedules them to available (virtual) cores, where scheduling considers constraints owing to cache affinities of cores executing dependent processes (or threads of the same process) and synchronization delays at execution barriers (joins). In Spark (a general purpose middleware for parallel computing), small-sized tasks (threads) are scheduled, workers pull-in new tasks when they're idle, and tasks may be rescheduled when they're deemed stragglers by the dispatcher. As such, this reactive approach with small-sized tasks is suited for workers with uncertain execution speeds and for poorly characterized or uncharacterized workloads.

IT resource scheduling naturally occurs on a slower time-scale and thus requires knowledge of available resources and workload characterizations corresponding to IT resource demands in the near future. So for the *proactive* resource scheduling described in the following, there is a basic assumption of offline characterization of resource needs and availability. This assumption is supported implicitly in *e.g.*, [5, 17, 3] and explicitly in *e.g.*, [15, 4]. Resource needs from such characterizations could obviously be based on quality-of-solution and quality-of-service requirements (particularly involving response/execution times) and may be time-varying, *e.g.*, [10, 4], though we do not consider time-varying requirements herein.

Moreover, processes running in public clouds may experience high-frequency effective capacity variation [11, 16] due to competition for unvirtualized resources such as memory bandwidth and lowest-level CPU/core caches *e.g.*, [22], hardware heterogeneity affecting performance, or deterministic token-bucket regulation of IT resources as in Amazon EC2 burstable instances [13]. How tenants could adapt to such variations in effective capacity, which may not violate public-cloud service level agreements (SLAs), is also beyond the scope of the present work.

We consider a plurality of heterogeneous, distributed tenants (whole application processes and their component threads) whose resource needs have been characterized in the form of a service level agreement with a public cloud. We consider how a public cloud could consolidate/schedule these resource requirements on a plurality of heterogeneous pools of different resource types, *i.e.*, physical servers or portions thereof[1]. The tenant/process resource demands are assumed specified as elastic (as [5]) and potentially congest available resources. In this context, we've previously argued that future neutrality regulations of a public cloud may require that resource allocations to tenants be "fair" in a congested regime [6][2].

Fair allocation of distributed resources to different parties is a classical problem, *e.g.*, [1, 12]. The problem of max-min fair (MMF) resource allocation was recently considered for multiple pools of the same resource (*e.g.*, network IO, CPU) in *e.g.*, [20, 3, 8, 7] under fixed demands with specified pool preferences and tenant priorities/weights. We are herein interested in a highly heterogeneous setting both in terms of demand and supply. There is a well-known trade-off between resource utilization and fairness to tenants/users, see Section III.C of [12] and Section 6.5.2 of [1] (there in the context of flows through a network).

This paper is organized as follows. In Section 2, we describe MMF and DRF and discuss prior work on scheduling of processes spanning multiple pools (servers) of multiple resource types [17, 3]. In Section 3, we describe our proposed scheduler to jointly consider multiple heterogeneous pools of resources each with different resource types and elastic demand. Some illustrative numerical examples are given in Section 4 to show the compromise between DRF and a kind of proportional fairness. In Section 5, we discuss subadditive workload models that are more general than "proportionately" elastic. We conclude with a summary and discussion of future work in Section 6. In the Appendix, we provide proofs for our extensions of the results of [12, 7].

[1] within one physical server, a cluster of cores (processors, CPUs) with common cache memory and separate disk and network IO resources

[2] The scheduling frameworks considered herein could also apply to a private or to a large tenant scheduling their plural applications on containers residing in plural Virtual Machines (VMs) or containers, in which case some of the "economic" issues considered in [5, 14] may apply. Beyond neutrality regulations in the public cloud setting, tenants simply "get what they pay for" and "use it or lose it" regarding any reserved resources.

2 Background: MMF and DRF

2.1 Multiple pools of one resource type - MMF

Given different resources i of common type in the amount $s_i > 0$, processes n with weights/priorities $\phi_n > 0$, and preference Booleans $\delta_{n,i} \in \{0,1\}$, define process n's weighted resource share as

$$F_n \quad = \quad \sum_i s_i \alpha_{n,i} \Big/ \phi_n \ , \tag{1}$$

for non-negative allocation parameters α satisfying

$$\forall i, \ \sum_n \alpha_{n,i} \delta_{n,i} = 1 \quad \text{and} \quad \alpha_{n,i} > 0 \ \Rightarrow \ \delta_{n,i} = 1. \tag{2}$$

An allocation α is said to be max-min fair if for each process $n \in N$, F_n cannot be increased while without decreasing F_m for some process m for which $F_m \le F_n$ [1, 12, 20]. That is, a server should be allocated only to processes that have the minimum normalized throughput among processes connected to it: if $F_n > F_m$ and $\delta_{m,i} = 1$, then $\alpha_{n,i} = 0$.

Claim 1. *Optimizing the Lagrangian*

$$L_0 = \sum_n \phi_n g(F_n) + \sum_i v_i (1 - \sum_n \delta_{n,i} \alpha_{n,i}) + \sum_{n,i} u_{n,i} \alpha_{n,i} \delta_{n,i}$$

leads to MMF allocations, where g is strictly concave and increasing and Lagrange multipliers $v, u \ge 0$ associated with the simplex constraints for the α variables (2).

A proof is given in [7]. A definition of MMF is given in the Appendix below. Note how the feasibility conditions give a convex (simplex) domain for α. Thus, the MMF allocation exists and is unique. A primal-dual or projected-gradient method could be used in practice to find the MMF allocation. Here, MMF results in 100% resource utilization. Also see Lemma 3 and Corollary 2 of [12] - in their model and that of [1], the allocation of resources to flows is the minimum of that at each network node, where here it's the sum.

In [20], the preference parameters δ were motivated based on service quality requirements of the corresponding packet-flows. Now consider scheduling of multiple cores/processors. Recently, [18] focused on the coupling between CPU and co-processor resources in one pool. The δ parameters could be used to indicate how simultaneously executing processes n (or threads of the same processes running in containers of a multiple-VM tenant) would naturally have affinity to cores that share caches. Issues of "atomizing" or threading workloads in order to assign them to processors are discussed in *e.g.*, [17, 7].

For an illustrative example, assume three equally weighted processes $N = \{g, h, l\}$, i.e., $\forall n \in N$, $\phi_n = 1/3$, two cores $S = \{1, 2\}$ with capacities $s_1 = 2.5$GHz and $s_2 = 1.7$GHz, and preferences $\delta_{n,i} = 1$ only if $n = g, h$ or $i = 2$ (*i.e.*, $\delta_{l,1} = 0$). At max-min fair allocation, $\alpha_{g,1} = \alpha_{h,1} = 0.5$ and $\alpha_{g,2} = \alpha_{h,2} = 0.15/1.7$, $\alpha_{l,2} = 1.4/1.7$. But, if the the second core is capacity reduced to $s_2 = 1$GHz, then the max-min fair allocations are $\alpha_{g,1} = \alpha_{h,1} = 0.5$, but $\alpha_{g,2} = \alpha_{h,2} = 0$, $\alpha_{l,2} = 1$. So for fully utilized cores, MMF gives proportional fairness ($\forall n$, $F_n \propto \phi_n$) when $s_2 = 1.7$, but not when $s_2 = 1$GHz. See Fig. 2 of [12] for another example, again with allocation to a process/flow being the minimum, not sum, of that given at each node associated with the flow[3].

Accordingly, we have the following claim, also see [2, 9].

Claim 2. *A necessary condition so that MMF implies proportional fairness is based on the maximum normalized service of any process:*

$$\min_{n \in N} \phi_n^{-1} \sum_i \delta_{n,i} s_i - \sum_i (1 - \delta_{n,i}) s_i \Bigg/ \sum_{n' \neq n} \phi_{n'} \geq 0.$$

2.2 Single pool, multiple resource types, linear resource demands, resource congestion - weighted DRF

For a single pool (i) of resources of different types (r), Dominant Resource Fairness (DRF) can be used to distribute resources [5] as "Nash bargaining" (choice of boundary Nash equilibrium under congested resources) with desirable properties such as Pareto optimality and strategy-proofness. DRF can be extended to include process priorities (ϕ), but preferences (δ) across different resource pools, as [20, 19], are not considered. DRF assumes a model of resource demands (xd) that are all proportionate to incident workload (x). That is, $x_{n,i}$ is the incident workload of process n in pool i.

First consider a single resource pool indexed i. Suppose that every process n has demand parameters d so that its Dominant Resource (DR) is

[3]For proportional fairness, the objective $\sum_n \phi_n g(F_n)$ in Claim 1 is replaced by $\sum_n \log(F_n)$ [12].

[5]

$$\rho(n,i) \ := \ \arg\max_r d_{n,r}/s_{i,r} \tag{3}$$

The fraction of resource r allocated to process n in pool i is

$$\alpha_{n,i,r} \ := \ \frac{x_{n,i}d_{n,r}}{s_{i,r}} \text{ and } x_{n,i} > 0 \ \Rightarrow \ \delta_{n,i} = 1, \tag{4}$$

i.e., α, x are variables that are directly related through parameters d, s. If *all* processes are thus flexible (no upper bound on their incident workloads x) and all processes have demand for at least one common resource (*i.e.*, $\forall n$, $d_{n,r} > 0$ for r =CPU, memory), then weighted Dominant Resource Fairness (DRF) is achieved in pool i when the weighted Dominant Resource Shares (DRSs) $\{\alpha_{n,i,\rho(n,i)}\phi_n\}_{n:\delta_{n,i}=1}$ are equal.

As such, the DRF is distinct from proportional fairness, *e.g.*, [12] and Claim 5 and Corollary 3 in the Appendix.

The following claim is an immediate consequence of the following proportionate workload model: $\forall r, n$,

$$\begin{aligned}\alpha_{n,i,r} \ &= \ \frac{d_{n,r}/s_{i,r}}{d_{n,\rho(n,i)}/s_{i,\rho(n,i)}}\alpha_{n,i,\rho(n,i)} \\ &=: \ b_{n,i,r}\alpha_{n,i,\rho(n,i)} \ \leq \ \alpha_{n,i,\rho(n,i)}.\end{aligned} \tag{5}$$

Claim 3. *The unique weighted DRF allocation for pool i under congestion is, $\forall n$ s.t. $\delta_{n,i} = 1$,*

$$\alpha_{n,i,\rho(n,i)} \ = \ \left(\phi_n \max_r \sum_{n':\delta_{n',i}=1} b_{n',i,r}/\phi_{n'}\right)^{-1},$$

and $\alpha_{n,i,\rho(n,i)} = 0$ if $\delta_{n,i} = 0$.

Proof. [5, 14] Fix index of resource pool, i. Recall by assumption that $\forall r$, $x_{n,i} = \alpha_{n,i,r}s_{i,r}/d_{n,r}$ when $\delta_{n,i} = 1$. The DRF solution is by definition constant DR allocations/shares: $\forall n$, $A_i := \alpha_{n,i,\rho(n,i)}\phi_n$. Congested (highest utilization) allocation A_i is found through tight resource-capacity constraints:

$$\begin{aligned}1 \ &= \ \max_r \sum_n \alpha_{n,i,r}\phi_n/\phi_n \\ &= \ \max_r \sum_n b_{n,i,r}\alpha_{n,i,\rho(n,i)}\phi_n/\phi_n \\ \Rightarrow A_i \ &= \ 1/\max_r \sum_n b_{n,i,r}/\phi_n.\end{aligned}$$

□

Generally, DRF may *not* result in 100% utilization of all resources.

2.3 Multiple pools of multiple resource types

Now consider multiple heterogeneous pools of different resource types. In this section, elastic demand according to (5) is assumed, unless otherwise noted.

Under DRFH [17], the "global" DR of each process n is just based on the total amount of each resource r across all heterogeneous resource-pools is computed, $\sum_i s_{i,r}$:

$$R(n) \quad := \quad \arg\max_r d_{n,r} / \sum_i s_{i,r}. \tag{6}$$

DRFH's unconstrained objective is

$$\max_\alpha \min_n \frac{1}{\phi_n} \sum_i \min_r \frac{\alpha_{n,i,r} s_{i,r}}{d_{n,r}} \cdot \frac{d_{n,R(n)}}{\sum_i s_{i,R(n)}}, \tag{7}$$

interpreted as maximizing the minimum global DRS (here weighted by ϕ_n). Global DR could be modified for scenarios with preferences for resource pools. The approach we advocate in Section 3 below simply sums the DRS of each resource pool for each process, thus extending [12, 7]. For the proportionate demand model (4),(5), see Corollaries 2 (max-min fairness, but subject to a special condition (33)) and 3 (proportional fairness) in the Appendix below.

Now suppose that resource allocations α are not constrained proportionate to d. The following is an extension of the approach described in [3]. Define the variables

$$M_n \quad = \quad \min_{i,r} \frac{\alpha_{n,i,r} s_{i,r}}{d_{n,r}}, \tag{8}$$

i.e., $\alpha_{n,i,r} s_{i,r} / d_{n,r}$ is here not defined as constant ($= x_{n,i}$) in r. Also considering preferences δ, weights ϕ and multiple resource types, we can formulate the objective as maximizing utilization,

$$\max_\alpha \sum_{n,i,r} \alpha_{n,i,r} / \phi_n \tag{9}$$

such that the M_n/ϕ_n are all equal, *i.e.*,

$$\begin{aligned} 0 &= \text{var}(M/\phi) := \frac{1}{N}\sum_n (M_n/\phi_n)^2 - \left(\frac{1}{N}\sum_n (M_n/\phi_n)\right)^2, \\ 0 &< \alpha_{n,i,r} \Rightarrow 1 = \delta_{n,i}. \end{aligned}$$

Alternatively, we can use the following unconstrained objective,

$$\max_\alpha \sum_n M_n/\phi_n \tag{10}$$

where the inequality constraints on the M_n variables can be used, $M_n \leq \alpha_{n,i,r} s_{i,r}/d_{n,r}$, for purposes of optimization (instead of definition (8).

Related optimizations are addressed in the Appendix: Claim 4 for max-min fairness (generalizing Theorem 1 of [7], but subject to a special condition (27)) and Claim 5 on proportional fairness (generalizing Lemma 2 of [12]).

In [3], a two-phase approach is also disclosed. First, the DRF solution for each pool (requiring $\delta_{n,i} > 0 \Rightarrow \alpha_{n,i} > 0$) and then the remaining resources are allocated in such a way that the process' DRs ρ do not change [3]. A neutral, public cloud may not allocate remaining resources in a manner not consistent with tenant-cloud SLAs. That is, the manner in which remaining resources are allocated needs to be consistent with elastic resource demands as stipulated in SLAs and otherwise "fair" [6], or remaining resources are simply unallocated for the sake of neutrality and in anticipation of future demand that may be able to exploit them.

3 Multiple pools of different resource types

Recall that Equations (3)-(5) define dominant resources for each resource pool i. In the following, take the total weighted resource share of each process n to be

$$F_n = \sum_i \alpha_{n,i,\rho(n,i,x_{n,i})} \Big/ \phi_n \tag{11}$$

Consider the optimization objective (again, similar to [12]) that spans all resource pools:

$$\max_\alpha \Omega := \max_\alpha \sum_n \phi_n g(F_n) \tag{12}$$

subject to the compact resource-capacity constraints

$$\forall i, r,\ s_{i,r} \geq \sum_n x_{n,i}\delta_{n,i}d_{n,r} \quad \text{and} \quad \forall n, i,\ x_{n,i} \geq 0. \tag{13}$$

Note that Ω is continuous and strictly concave and the domain given by (13) is compact. So, simply by Weierstrass's Extreme Value Theorem, there exists a unique maximum for Ω subject to (13).

This optimization is addressed in the Corollaries of the Appendix below regarding MMF and proportional fairness. The optimization does not necessarily achieve MMF (MMF requires condition (33) below) nor DRF.

For proportional fairness, we can constrain our optimization problem to solutions with zero variance of the total weighted resource shares

$$0 \ = \ \text{var}(F_n) := \frac{1}{N}\sum_n (F_n)^2 - \left(\frac{1}{N}\sum_n F_n\right)^2, \tag{14}$$

where n is the the total number of processes. Similarly for DRF, we can constrain

$$0 \ = \ \sum_i V_i, \tag{15}$$

where the variance of DRS's in pool i is

$$\begin{aligned} V_i &= \text{var}_n(\alpha_{n,i,\rho(n,i)}) \\ &= \frac{1}{\tilde{\Delta}_i}\sum_n \left(\alpha_{n,i,\rho(n,i)}\right)^2 - \left(\frac{1}{\tilde{\Delta}_i}\sum_n \alpha_{n,i,\rho(n,i)}\right)^2, \\ \& \ \tilde{\Delta}_{n,i} &= \sum_n \mathbf{1}\{\alpha_{n,i,\rho(n,i)} > 0\}. \end{aligned}$$

Existence of a unique maximizing solution $\{x_{n,i}\}$ of (12) such that (13) continues to hold if (14) or (15) are added. Achieving both DRF and proportional fairness (or MMF) may not be possible.

Numerically, optimization is complicated by discontinuity of the variance terms V at the origin[4]. This discontinuity may be relaxed but the resulting constraint term $-\sum_i V_i$ will generally not be concave[5]. Optimization is also

[4]To see why, note that simply $\lim_{c\downarrow 0} \frac{a+b+c}{3} \neq \frac{a+b}{2}$.

[5]See, *e.g.*, [21] regarding continuous relaxation, *e.g.*, by using arctan or tanh, with sufficiently high derivative at the origin, instead of the step in the $\tilde{\Delta}$ terms. Heuristically, a projected Armijo gradient with random restart could be employed, and the α (or x) variables set to zero when they become sufficiently small, or a mixed-integer programming approach could be used.

simplified if $\forall n, i$: (i) $\rho(n,i)$ are fixed independent of workload intensities $x_{n,i}$, and (ii) $\delta_{n,i} = 1 \Rightarrow \alpha_{n,i,\rho(n,i)} > 0$ (the latter condition simply avoiding the discontinuity issue just raised). Accordingly, this optimization can be solved by a simple primal-dual or projected-gradient method. Note that the second simplification means positive allocation in all "preferred" resource pools (*i.e.*, $\delta_{n,i} = 1 \Leftrightarrow \alpha_{n,i,\rho(n,i)} > 0$), avoiding the discontinuity of the variance V terms at the origin. In the example of Section 2.1, this condition is satisfied when $s_2 = 1.7$GHz but not when $s_2 = 1$GHz.

Alternatively, we may modify the optimization objective Ω of (12) to

$$\sum_n \phi_n g(F_n) - t_{\text{PF}} \text{var}(F_n) - t_{\text{DRF}} \sum_i V_i, \tag{16}$$

where $t_{\text{PF}}, t_{\text{DRF}} \geq 0$ are parameters (rather than a Lagrange multipliers) trading-off proportional fairness across resource pools (servers) and DRF within pools, with respect to allocations F_n.

4 Illustrative numerical examples

Consider a system with 2 heterogeneous pools $S = \{1, 2\}$, each pool preserves 2 resources, *i.e.*, $R = \{1, 2\}$. Take the resource capacities as

$$\boldsymbol{s} = \begin{bmatrix} 9 & 18 \\ 18 & 9 \end{bmatrix}. \tag{17}$$

Let $N = \{1, 2, 3\}$ be the set of equally weighted processes ($\phi_1 = \phi_2 = \phi_3$) sharing the resources.

In addition to measures of DRF $\sqrt{\sum_i V_i}$ and proportional fairness (PF) $\sqrt{\text{var}(F_n)}$, we consider the total utilizations of different resources $(i, r) \in S \times R$,

$$c_{i,r} \;\; = \;\; \sum_{n \in N} x_{n,i} d_{n,r} / s_{i,r} = \sum_{n \in N} \alpha_{n,i,r}.$$

The objective function $g(\cdot)$ in (16) is logarithm. The optimization procedure was randomly initialized by a feasible allocation of the DRSs.

4.1 Example with No Preference Restrictions

First consider the case where there are no preference restrictions ($\forall n, i$, $\delta_{n,i} = 1$), and resource demands

$$\boldsymbol{d} = \begin{bmatrix} 1 & 3 & 3 \\ 4 & 1 & 4 \end{bmatrix}^{\text{T}} \tag{18}$$

The process and result of optimizing (16) with $t_{\mathrm{PF}} = 0 = t_{\mathrm{DRF}}$ (*i.e.*, optimizing just Ω of (12)), subject to capacity constraints, is depicted in Figures 1-4. We see here that PF and DRF are jointly reached but that not all resources are fully utilized. Regarding Figures 3 and 4, the found dominant resources $\rho(n,i)$ and dominant resource shares $\alpha_{n,i,\rho(n,i)}$ are

$$\boldsymbol{\rho} = \begin{bmatrix} 2 & 1 & 1 \\ 2 & 1 & 2 \end{bmatrix}^{\mathrm{T}} \qquad \boldsymbol{\alpha}_{\mathrm{DRS}} = \begin{bmatrix} 0.4 & 0.4 & 0.4 \\ 0.375 & 0.375 & 0.375 \end{bmatrix}^{\mathrm{T}}$$

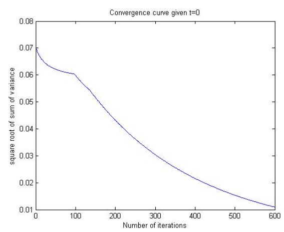

Figure 1: $\sqrt{\sum \tilde{V}_i}$

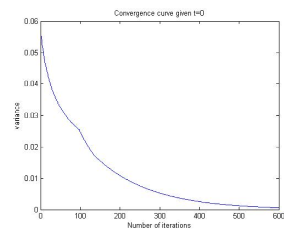

Figure 2: $\sqrt{\mathrm{var}(F_n)}$

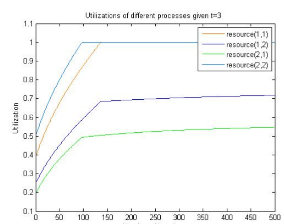

Figure 3: $c_{i,r}$

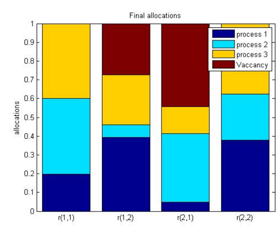

Figure 4: Final Allocations

4.2 Examples with Heterogeneous Preferences

For resource capacities (17), consider resource demands and pool-preferences,

$$\boldsymbol{d} = \begin{bmatrix} 3 & 1 & 2.5 \\ 4 & 1 & 4 \end{bmatrix}^{\mathrm{T}} \quad \boldsymbol{\delta} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}^{\mathrm{T}}$$

i.e., process $n = 1$ does not use resource pool $i = 1$.

4.2.1 No DRF or PF Penalty

The process and results of optimizing (16) with $t_{\mathrm{DRF}} = 0 = t_{\mathrm{PF}}$ are given in Figures 5-8. We see that here PF and DRF are *not* reached (and again not all resources are fully utilized). The DRs of different processes in the same resource pool are no longer equalized since preferences of processes lead to multiple bottlenecks.

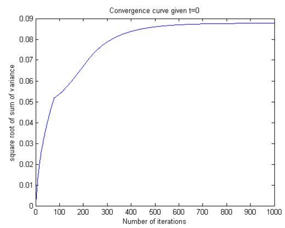

Figure 5: $\sqrt{\sum \tilde{V}_i}$

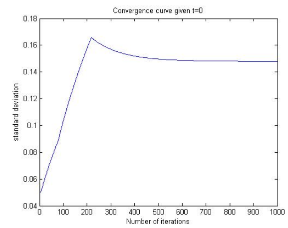

Figure 6: $\sqrt{\mathrm{var}(\mathrm{F_n})}$

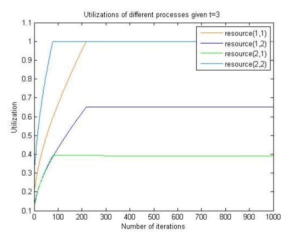

Figure 7: $c_{i,r}$

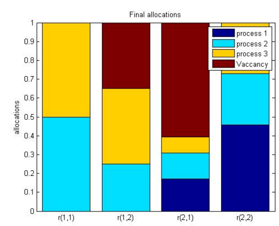

Figure 8: Final Allocations

The found DRs and DRSs are

$$\boldsymbol{\rho} = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \end{bmatrix}^{\mathrm{T}} \qquad \boldsymbol{\alpha}_{\mathrm{DRS}} = \begin{bmatrix} 0 & 1/2 & 1/2 \\ 0.4574 & 0.2713 & 0.2713 \end{bmatrix}^{\mathrm{T}}$$

Note that $\alpha_{1,2,\rho(1,2)}$ can be increased ($\Rightarrow$ F_1 increases) by decreasing $\alpha_{2,2,\rho(2,2)}$ or $\alpha_{3,2,\rho(3,2)}$ while $F_1 < \min\{F_2, F_3\}$ and some resources in pool 2 are fully booked, thus violating the definition of MMF in particular, *cf.* the Appendix. The distinction between MMF and PF is pointed out in *e.g.*, [12] and in the example of Section 2 above (where resource bottlenecks are caused by different preference parameters $\delta_{n,i}$).

4.2.2 DRF Penalty

Now suppose only $t_{\mathrm{PF}} = 0$ in (16). The process and results of Figures 9-12 for large $t_{\mathrm{DRF}} = 900$ show that DRF is reached (y-axis in multiples of 10^{-3})

but not PF, and the DRSs are

$$\boldsymbol{\alpha}_{\text{DRS}} = \begin{bmatrix} 0 & 1/2 & 1/2 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}^{\text{T}}$$

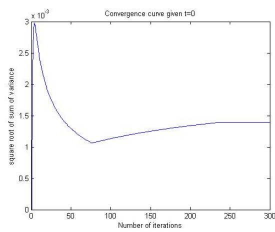

Figure 9: $\sqrt{\sum \tilde{V}_i}$

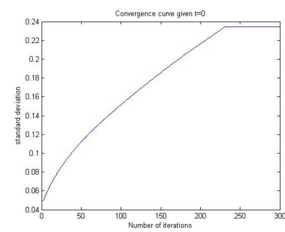

Figure 10: $\sqrt{\text{var}(\text{F}_{\text{n}})}$

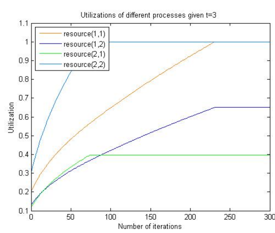

Figure 11: $c_{i,r}$

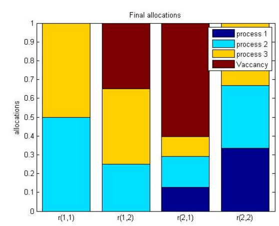

Figure 12: Final Allocations

4.2.3 PF Penalty

For demand matrix (18) and preference matrix

$$\boldsymbol{\delta} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}^{\text{T}}$$

we also optimized (16) with $t_{\text{DRF}} = 0$ but large PF penalty $t_{\text{PF}} = 900$. The process and results are given in Figures 13-16. The converged allocation is PF across all pools and, interestingly, shows a high utilization rate. Here the DRs and DRSs are

$$\boldsymbol{\rho} = \begin{bmatrix} 2 & 1 & 1 \\ 2 & 1 & 2 \end{bmatrix}^{\text{T}} \qquad \boldsymbol{\alpha}_{\text{DRS}} = \begin{bmatrix} 0.6000 & 0.0000 & 0.6000 \\ 0.2257 & 0.8229 & 0.2257 \end{bmatrix}^{\text{T}}$$

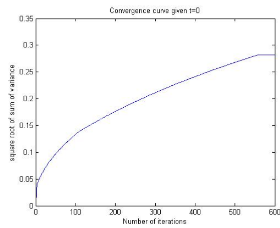

Figure 13: $\sqrt{\sum \tilde{V}_i}$

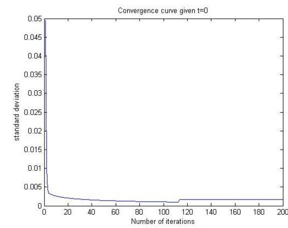

Figure 14: $\sqrt{\mathrm{var}(\mathrm{F_n})}$

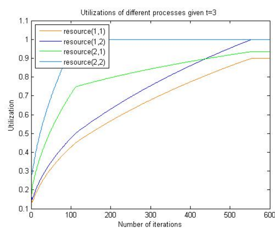

Figure 15: $c_{i,r}$

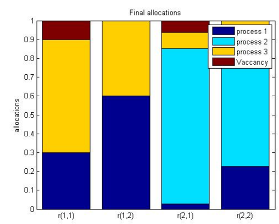

Figure 16: Final Allocations

5 Discussion: More general workload models for a single resource pool

Suppose a locally executed (resource-pool/server indexed i) application process (indexed n) which primarily requires CPU ($s_{i,1}$ cores) and disk IO ($s_{i,2}$ bit/s) and has execution time (Quality of Service or QoS) requirement T_n. Also suppose it requires resource allocations $(\alpha_{n,i,1}, \alpha_{n,i,2})$ for workload (dataset) size $x_{n,i}$ such that

$$\left(\frac{1}{\alpha_{n,i,1}s_{i,1}} + \frac{1}{\alpha_{n,i,2}s_{i,2}}\right) x_{n,i} \quad \leq \quad T_n, \tag{19}$$

where we've obviously assumed that disk IO is not pipelined with CPU activity (see below). *E.g.,* the application could be an annotated search of a textual/document dataset of size $x_{n,i}$, where the disk IO is required to load the data into memory. So, the set of feasible $(\alpha_{n,i,1}, \alpha_{n,i,2}) \in [0,1]^2$ allocations is convex. More specifically, for each fixed workload $x_{n,i}$, the feasible allocations $(\alpha_{n,i,1}, \alpha_{n,i,2})$ given by (19) are on or above a hyperbola. For this example application, a scarcity in one resource type (CPU or disk IO) can be relieved the other.

With pipelining using $\alpha_{n,i,3}s_{i,3}$ ($< x_{n,i}$) allocated RAM, the term $x_{n,i}/(\alpha_{n,i,1}s_{i,1})$

in (19) could be reduced to $\alpha_{n,i,3}s_{i,3}/(\alpha_{n,i,1}s_{i,1})$. This results in the CPU requirement being proportionate to demand $x_{n,i}$ as in Section 2.2:

$$\alpha_{n,i,1} \quad = \quad \left(T_n - \frac{\alpha_{n,i,3}s_{i,3}}{\alpha_{n,i,2}s_{i,2}}\right)^{-1} \frac{x_{n,i}}{s_{i,1}}.$$

The required allocations $D(x) \triangleq \alpha s$ are said to be *subadditive* in applied workload $x \geq 0$ when

$$\forall x^{(1)}, x^{(2)} \geq 0,\ D(x^{(1)} + x^{(2)}) \quad \leq \quad D(x^{(1)}) + D(x^{(2)}).$$

Since generally D is continuous, increasing and $D(0) = 0$, concavity of D implies subadditivity.

For another simple illustrative example, consider an M/M/1 queue with mean job service rate D and workload (mean job arrival rate) $x < D$, so that the mean sojourn time (queueing delay plus service time) is $Q = 1/(D - x)$. For required QoS $q = Q$, we see here that $D(x) = x + q^{-1}$ is affine and thus subadditive. Alternatively, define $D > x$ so that the probability that that the sojourn time of a job exceeds θ is at most $\mu \in (0, 1)$, *i.e.*, $(x/D)^{\theta D} = \mu$ where the required QoS is parameterized $\underline{q} = (\theta, \mu)$. One can directly show that D is concave ($D'' \leq 0$) and $D(0) \geq 0$ ($\forall x \geq 0,\ D(x) > x$), hence subadditive.

In a tenant/public-cloud SLA, the specific desired resource allocations for a specific workload intensity could be stipulated, or the cloud could make choices according to an SLA expressed for elastic resource requirements (as (19)), possibly with elastic demand (variable x). When the cloud makes such choices on behalf of the tenants, issues of fairness and neutrality ensue [6] - to this end the weighted total allocation quantities F_n may be used to define fair resource allocations. That is, the cloud itself would have a neutral resource allocation objective as a function of the F_n that it would optimize to find the allocations $\{\alpha_{n,i,r}\}$ for all processes n.

Generalizing (19) (again without pipelining), we may model the QoS requirement as

$$x_{n,i} \sum_r \frac{1}{D_{n,r}(x_{n,i})} \quad \leq \quad T_n, \tag{20}$$

for subadditive $D_{n,r} \geq 0$. If we model, *e.g.*, $D_{n,r}(x) = c_{n,r}x^{a_{n,r}}$ where $c_{n,r} > 0$ and $0 < a_{n,r} < 1$, then $x_{n,r}/D_{n,r}(x_{n,r})$ is concave and so the feasible set $(D_{n,r}(x))_r$ is *not* generally convex, which may complicate optimization.

Alternatively, suppose that we generalize (5) as: $\forall n, i, r$,

$$\alpha_{n,i,r} \quad = \quad \frac{1}{s_{i,r}} D_{n,r}(D^{-1}_{n,\rho(n,i)}(\alpha_{n,i,\rho(n,i)} s_{i,\rho(n,i)})), \tag{21}$$

where under (21), the DR of process n depends on its incident workload

$$\forall r,\ x_{n,i} \quad = \quad D^{-1}_{n,r}(\alpha_{n,i,r} s_{i,r}) \tag{22}$$

$$\rho(n, i, x_{n,i}) \quad = \quad \arg\max_r \frac{D_{n,r}(x_{n,i})}{s_{i,r}} = \arg\max_r \alpha_{n,i,r}. \tag{23}$$

So here the demands for different types of resources are "independent" given the workload intensity x.

For instance, consider resource pool i and suppose that for process n, $D_{n,r}(x_{n,i}) = d_{n,r} x_{n,i} + c_{n,r}$ for parameters $d_{n,r} > 0, c_{n,r} \geq 0$. Thus $\forall r$, $\alpha_{n,i,r}$ satisfies

$$D^{-1}_{n,r}(\alpha_{n,i,r} s_{i,r}) = (\alpha_{n,i,r} s_{i,r} - c_{n,r})/d_{n,r} = x_{n,i}.$$

Process n's DR in pool i is

$$\rho(n, i, x_{n,i}) = \arg\min_r (d_{n,r} x_{n,i} + c_{n,r})/s_{i,r} = \arg\min_r \alpha_{n,i,r},$$

which coincides with the definition of DR in [5] (and does not depend on $x_{n,i}$) when $\forall r$, $c_{n,r} = 0$.

Again note that if D_1, D_2 are both strictly subadditive (so concave) and increasing functions, a (nonempty) constraint set such as $\{v : s \geq D_1(D_2^{-1}(v))\}$ is generally *not* convex, which may complicate optimization. For the special case of affine D_1 and strictly concave D_2, the constraint set is convex.

Corollary 1. *For subadditive resource demands, the statement of Claim 3 holds with $D_{n,r}(x_{n,i})$ instead of $d_{n,r}$ in (5).*

6 Summary and future work

In summary, we considered a fair scheduling problem for heterogeneous processes n of different priorities (δ_n) with flexible demands ($x_n d_{n,r}$) for a plurality of different resource types (r) and with preferences ($\delta_{n,i}$) for specific heterogeneous resource pools (i). Max-min, proportional and dominant-resource fairness were jointly considered. We extended the results of [12, 7] to the heterogeneous, multi-resource case with resource-pool preferences. A scheduler was proposed that can trade-off MMF and DRF and its properties were illustrated with numerical examples.

In future work, we will consider pricing for different resources, which may vary from pool to pool. Given prices, we can consider issues such as revenue maximizing allocations for the cloud, in addition to "fair" allocations with desirable properties such as Pareto optimality and strategy-proofness (as possessed by DRF for a single resource pool [5]).

We will also explore optimization using subadditive resource demand models.

References

[1] D. Bertsekas and R. Gallager. *Data Networks, 2nd Ed.* Prentice Hall, 1992.

[2] A. Chandra, M. Adler, P. Goyal, and P. Shenoy. Surplus Fair Scheduling: A Proportional-share CPU Scheduling Algorithm for Symmetric Multiprocessors. In *Proc. USENIX Conf. on Operating System Design & Impl. (OSDI)*, 2000.

[3] M. Chowdhury, Z. Liu, A. Ghodsi, and I. Stoica. HUG: Multi-resource fairness for correlated and elastic demands. In *Proc. USENIX NSDI*, March 2016.

[4] C. Delimitrou and C. Kozyrakis. HCloud: Resource-Efficient Provisioning in Shared Cloud Systems. In *Proc. ASPLOS*, Atlanta, April 2016.

[5] 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 NSDI*, 2011.

[6] G. Kesidis, N. Nasiriani, B. Urgaonkar, and C. Wang. Neutrality in Future Public Clouds: Implications and Challenges. In *Proc. USENIX HotCloud*, June 2016.

[7] J. Khamse-Ashari, G. Kesidis, I. Lambadaris, B. Urgaonkar, and Y. Zhao. Constrained Max-Min Fair Scheduling of Variable-Length Packet-Flows to Multiple Servers. In *Proc. IEEE GLOBECOM*, Washington, DC, Dec. 2016.

[8] J. Khamse-Ashari, G. Kesidis, I. Lambadaris, B. Urgaonkar, and Y. Zhao. Max-Min Fair Scheduling of Variable-Length Packet-Flows to Multiple Servers by Deficit Round-Robin. In *Proc. CISS, Princeton*, March 2016.

[9] J. Khamse-Ashari, G. Kesidis, I. Lambadaris, B. Urgaonkar, and Y. Zhao. Efficient and Fair Scheduling of Placement Constrained Threads on Heterogeneous Multi-Processors. In *preprint*, Sept. 2016.

[10] I. Manotas, R. Zhang, M. Li, R. Tewari, and D. Hildebrand. Resource Usage Characterization for Social Networks Analytics on Spark. In *Proc. ACM Supercomputing (SC15)*, 2015 (poster).

[11] X. Meng, C. Isci, J. Kephart, L. Zhang, E. Bouillet, and D. Pendarakis. Efficient resource provisioning in compute clouds via VM multiplexing. In *Proc. ACM ICAC*, Washington, DC, 2010.

[12] J. Mo and J. Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Trans. Networking*, Vol. 8, No. 5:pp. 556–567, 2000.

[13] N. Nasiriani, C. Wang, G. Kesidis, and B. Urgaonkar. A Measurement-based Study of Effective Capacity Dynamism on Amazon EC2. Technical Report CSE-16-05, CSE Dept, PSU, May 2016, http://www.cse.psu.edu/research/publications/tech-reports/2016/CSE-16-005.pdf.

[14] D. Parkes, A. Procaccia, and N. Shah. Beyond dominant resource fairness: Extensions, limitations, and indivisibilities. In *Proc. ACM EC*, Valencia, Spain, 2012.

[15] Z. Shen, S. Subbiah, X. Gu, and J. Wilkes. CloudScale: Elastic Resource Scaling for Multi-Tenant Cloud Systems. In *Proc. ACM SOCC*, Cascais, Portugal, 2011.

[16] C. Wang, B. Urgaonkar, R. Birke, A. Gupta, L. Chen, Q. Wang, G. Kesidis, and N. Nasiriani. A case for dynamic pricing with effective capacity modulation in the cloud. In *Proc. IEEE Int'l Conf. Auto. Comp. (ICAC)*, Wurzburg, July 2016.

[17] W. Wang, B. Li, and B. Liang. Dominant resource fairness in cloud computing systems with heterogeneous servers. In *Proc. IEEE INFOCOM*, 2014.

[18] W. Wang, Y. Tan, Q. Wu, and Y. Zhang. Multiple resources scheduling for diverse workloads in heterogeneous data centers. In *Proc. Int'l Conf. on Comp. Sci. and Net. Tech. (ICCSNT)*, 2015.

[19] K.-K. Yap. *Using All Networks Around Us.* PhD thesis, Stanford University, 2013.

[20] K.-K. Yap, T.-Y. Huang, Y. Yiakoumis, S. Chinchali, N. McKeown, and S. Katti. Scheduling packets over multiple interfaces while respecting user preferences. In *Proc. ACM CoNEXT*, Dec. 2013.

[21] I. Zang. Discontinuous Optimization by Smoothing. *Mathematics of Operations Research*, 6(1):140–152, Feb. 1981.

[22] X. Zhang, E. Tune, R. Hagmann, R. Jnagal, V. Gokhale, and J. Wilkes. CPI^2: CPU performance isolation for shared compute clusters. In *Proc. of the 8th ACM European Conference on Computer Systems*, 2013.

Appendix: Generalization of Theorem 1 of [7] (MMF) and of Lemma 2 of [12] (PF)

Define the preference sets and total workloads

$$N_i = \{n \mid \delta_{n,i} = 1\} \quad \text{and} \quad x_n = \sum_i x_{n,i}\delta_{n,i}. \tag{24}$$

In this appendix, we assume the demand model (4),(5) and strictly concave and increasing g with $g(0) = 0$. Define the optimization problem

$$\max_x \sum_n \phi_n g\left(x_n/\phi_n\right) \tag{25}$$

such that

$$\forall i, r, \ \sum_{n \in N_i} x_{n,i} B_{n,i,r} \leq 1 \quad \text{and} \quad \forall n, i \ x_{n,i} \geq 0, \tag{26}$$

where $B_{n,i,r} := d_{n,r}/s_{i,r}$. Regarding fully booked resources in pool i under allocations $x = \{x_{n,i}\}$, also define

$$R_i \ := \ \{(x, r) \mid \sum_{n \in N_i} x_{n,i} B_{n,i,r} = 1\}.$$

Definition 1. *A feasible allocation* $\{x_{n,i}\}$ *satisfying (26) is said to be* (x/ϕ)*-max-min fair (MMF) if:*

$$\delta_{l,i} = 1, x_{m,i} > 0, \;\&\; \exists r \; s.t. \sum_{n \in N_i} x_{n,i} B_{n,i,r} = 1$$

imply that

$$\frac{x_l}{\phi_l} \;\; \geq \;\; \frac{x_m}{\phi_m}.$$

So if $\{x_{n,i}\}$ is MMF, $x_{m,i} > 0$ ($\Rightarrow \delta_{m,i} = 1$), $\delta_{l,i} = 1$, and $x_l/\phi_l < x_m/\phi_m$ then $x_{l,i} = 0$, otherwise x_l/ϕ_l could be increased and x_m/ϕ_m through resource reallocation at pool i (particularly if no resource r is fully booked in pool i). Also, if $\{x_{n,i}\}$ is MMF and $x_{m,i}, x_{l,i} > 0$ then $x_{m,i}/\phi_m = x_{l,i}/\phi_l$.

Claim 4. *A solution* $x = \{x_{n,i}\}$ *of the optimization (25) s.t. (26) has at least one resource* r *is fully booked in each pool* i*. In addition,* $\{x_{n,i}\}$ *is uniquely* (x/ϕ)*-MMF if also*

$$\delta_{m,i} = 1 = \delta_{l,i} \;\&\; (x, r) \in R_i \quad \Rightarrow \quad d_{m,r} = d_{l,r}. \tag{27}$$

Remark: Regarding (27), note that $d_{m,r} = d_{l,r} \Rightarrow B_{m,i,r} = B_{l,i,r}$. Also, (27) would obviously be satisfied if

$$\delta_{m,i} = 1 = \delta_{l,i} \quad \Rightarrow \quad \forall r, \; d_{m,r} = 1 = \delta_{l,r}.$$

Proof. Define the Lagrangian to be maximized over x and over Lagrange multipliers $\lambda, \nu \geq 0$:

$$\begin{aligned} L \;\; = \;\; & \sum_n \phi_n g(x_n/\phi_n) + \sum_{i,r} \lambda_{i,r}(1 - \sum_{n \in N_i} x_{n,i} B_{n,i,r}) \\ & + \sum_{i, n \in N_i} \nu_{n,i} x_{n,i}. \end{aligned}$$

The first-order optimality condition,

$$\begin{aligned} & \forall i, n \in N_i, \; \delta_{n,i} = 1, \\ 0 = \frac{\partial L}{\partial x_{n,i}} \;\; = \;\; & g'(x_n/\phi_n) - \sum_r \lambda_{i,r} B_{n,i,r} + \nu_{n,i}, \end{aligned} \tag{28}$$

and g strictly increasing imply

$$\forall i, n \in N_i, \;\; \sum_r \lambda_{i,r} B_{n,i,r} > \nu_{n,i} \geq 0. \tag{29}$$

So, $\forall i$, $\exists r$ s.t. $\lambda_{i,r} > 0$. Thus, complementary slackness is

$$\forall i, r,\ \lambda_{i,r}(1 - \sum_{n \in N_i} x_{n,i} B_{n,i,r}) = 0 \tag{30}$$

$$\Rightarrow \quad \forall i, \exists r \text{ s.t.} \sum_{n \in N_i} x_{n,i} B_{n,i,r} = 1, \tag{31}$$

i.e., in every pool i, one resource r (which may depend on i) is fully booked. So, the set of fully booked resources in pool i under allocations $x = \{x_{n,i}\}$ can be characterized by $\{r \mid \lambda_{i,r} > 0\}$. Now by (28), uniquely

$$\begin{aligned} \forall i, n \in N_i,\ \frac{x_n}{\phi_n} &= (g')^{-1}\left(\sum_r \lambda_{i,r} B_{n,i,r} - \nu_{n,i}\right) \\ &= (g')^{-1}\left(\sum_{r:\lambda_{i,r}>0} \lambda_{i,r} B_{n,i,r} - \nu_{n,i}\right). \end{aligned}$$

Now consider two processes m and l and resource pool (server) i such that $x_{m,i} > 0$ and $\delta_{m,i} = 1 = \delta_{l,i}$. So, complementary slackness

$$\forall i, n \in N_i,\ \nu_{n,i} x_{n,i} = 0, \tag{32}$$

implies $\nu_{m,i} = 0$. Thus, because $(g')^{-1}$ is strictly decreasing (g strictly concave),

$$\begin{aligned} \frac{x_m}{\phi_m} &= (g')^{-1}\left(\sum_{r:\lambda_{i,r}>0} \lambda_{i,r} B_{m,i,r}\right) \\ &\leq (g')^{-1}\left(\sum_{r:\lambda_{i,r}>0} \lambda_{i,r} B_{l,i,r} - \nu_{l,i}\right) = \frac{x_l}{\phi_l}, \end{aligned}$$

where we have used (27) for the inequality. Because of this and (31), $\{x_{n,i}\}$ is MMF. □

Note that we can restate (11) as

$$F_n = \phi_n^{-1} \sum_{i:\delta_{n,i}=1} x_{n,i} \max_r B_{n,i,r} = \phi_n^{-1} \sum_{i:\delta_{n,i}=1} \max_r \alpha_{n,i,r}$$

The following corollary is proved just as the previous claim.

Corollary 2. *A solution $x = \{x_{n,i}\}$ of the optimization problem objective*

$$\max_x \sum_n \phi_n g(F_n) \quad \textit{s.t.} \quad (26)$$

is such that at least one resource r is fully booked in each pool i. Also, x is a uniquely F-MMF if in addition

$$\begin{aligned} \delta_{m,i} = 1 = \delta_{l,i} \ \& \ (x,r) \in R_i \quad \Rightarrow & \\ \frac{B_{m,i,r}}{\max_{r'} B_{m,i,r'}} \quad = \quad & \frac{B_{l,i,r}}{\max_{r'} B_{l,i,r'}}. \end{aligned} \tag{33}$$

Note that if there is a single resource type r in each pool (as [20, 7]), then (33) trivially holds. See Theorem 1 of [7].

In the final step of the proof of Claim 4, we see the need to divide by the weight ϕ in the argument of the objective function $g(\cdot)$. For weighted proportional fairness, we change objective (25) to

$$\max_x \sum_n \phi_n g_a(x_n), \tag{34}$$

i.e., without dividing by ϕ_n in the argument of g_a [12], and for parameter $a > 0$ specifically take

$$g_a(X) \quad = \quad \begin{cases} \log(X) & \text{if } a = 1 \\ (1-a)^{-1} X^{1-a} & \text{else} \end{cases}$$

i.e., $g'_a(X) = 1/X^a$, again see [12]. Obviously, in the case of $g = \log$ ($a = 1$), whether the factor ϕ is in the argument of g is immaterial.

Claim 5. *A solution x^* of the optimization (34) s.t. (26) is uniquely (weighted) (ϕ, a) x-proportional fair,* i.e., *for any other feasible solution x,*

$$\Phi(x, x^*) := \sum_n \phi_n \frac{x_n - x_n^*}{(x_n^*)^a} \quad \leq \quad 0.$$

Proof. The Lagrangian here is

$$\begin{aligned} L \quad = \quad & \sum_n \phi_n g_a(x_n) + \sum_{i,r} \lambda_{i,r} (1 - \sum_{n \in N_i} x_{n,i} B_{n,i,r}) \\ & + \sum_{i, n \in N_i} \nu_{n,i} x_{n,i} \delta_{n,i} \end{aligned}$$

where, again, the Lagrange multipliers $\lambda, \nu \geq 0$. A first-order optimality condition is

$$\begin{aligned} \forall i, n \in N_i,\ 0 &= \frac{\partial L}{\partial x_{n,i}}(x^*) \\ &= \phi_n g_a'(x_n^*) - \sum_r \lambda_{i,r} B_{n,i,r} + \nu_{n,i}. \end{aligned} \tag{35}$$

Multiplying (35) by $x_{n,i} - x_{n,i}^*$ and summing over i and $n \in N_i$ gives

$$\begin{aligned} 0 &= \sum_n \phi_n g_a'(x_n^*)(x_n - x_n^*) + \sum_{i,n \in N_i} \nu_{n,i}(x_{n,i} - x_{n,i}^*) \\ &\quad - \sum_{i,r} \sum_{n \in N_i} \lambda_{i,r} B_{n,i,r}(x_{n,i} - x_{n,i}^*) \end{aligned}$$

where the first term is $\Phi(x, x^*)$ and recall (24). Thus, by complementary slackness (30) and (32),

$$\Phi(x, x^*) = \sum_{i,r} \lambda_{i,r} \left(\sum_{n \in N_i} x_{n,i} B_{n,i,r} - 1 \right) - \sum_{i,n \in N_i} \nu_{n,i} x_{n,i}.$$

Finally, since x is a feasible solution, *i.e.*, (26), $\Phi(x, x^*) \leq 0$. □

Note that $\{x_n^* = \sum_i x_{n,i}^*\}_n$ is unique though $x^* = \{x_{n,i}^*\}_{n,i}$ may not be.
The following corollary is proved just as the previous claim.

Corollary 3. *A solution x^* of the optimization problem*

$$\max_x \sum_n \phi_n \log(F_n) \quad s.t. \quad (26)$$

is uniquely $(\phi, 1)$ F_n-proportional fair, i.e.*, for any other feasible x,*

$$\sum_n \phi_n \frac{F_n - F_n^*}{F_n^*} \quad \leq \quad 0.$$

That is, $\{F_n^*\}$ is unique though $x^* = \{x_{n,i}^*\}_{n,i}$ may not be.

Note that to generalizing Lemma 3 and Corollary 2 of [12] in this way requires an assumption like (27) or (33).