Proximal optimization for resource allocation in distributed computing systems with data locality

Diego Goldsztajn, Fernando Paganini and Andres Ferragut
Universidad ORT Uruguay

***Abstract*— We consider resource allocation questions for computing infrastructures with multiple server instances. In particular, the joint optimization of active service capacity, load balancing between clusters of servers, and task scheduling at each cluster, under conditions of data locality which imply different service rates for different cluster locations.**

Building on previous work, we formulate a convex optimization problem, and use Lagrange duality to decompose it between the different decision variables. We include regularization terms from proximal methods to obtain continuous control laws for load balancing and scheduling, and optimize the remaining variables through primal-dual gradient dynamics. We prove convergence of the resulting control laws to the desired optimal points, and demonstrate its behavior by simulations.

I. Introduction

Convex optimization has proven to be a powerful tool for solving resource allocation problems in telecommunication networks [15]. These continuous variable methods, with ingredients from control and dynamics, provide a macroscopic view that is difficult to attain within the more classical discrete models of queueing theory. Of particular importance are convex optimization methods for allocation decisions at multiple layers, where useful decompositions are found via Lagrange duality [4].

In recent years, the focus has shifted away from the communication substrate and into higher layers, in particular the now prevalent cloud computing infrastructures. Rather than packets and flows, the discrete entities are server instances, but resource allocation issues are still of crucial importance: right sizing of the active service capacity [8], load balancing of tasks between server locations [10], and scheduling of server instances [16]. Most of these recent references employ stochastic queueing theory tools.

Is there a role for convex optimization in this area? Our recent exploration [12] reveals that optimization decompositions are very natural in this context: in particular, the recent proposal [16] of Join-the-Shortest-Queue (JSQ) load balancing with MaxWeight (MW) scheduling, for MapReduce systems under data locality, can be naturally justified by Lagrange duality. And within this framework, the additional degree of freedom of sizing the server capacity is naturally included through primal-dual gradient dynamics [1], [6].

One difficulty with the approach taken in [12], framed in continuous time, is that JSQ and MW exhibit discontinuous switching around the optimal values. In this paper we address this issue by introducing quadratic regularization terms taken from the literature in proximal methods [13]. With this addition, load balancing and scheduling now evolve continuously, and still reach the same optimal solution. Establishing this fact is our main result, which requires extensions to the theory of saddle point dynamics.

The paper is organized as follows. Section II describes the model for the cloud computing infrastructure under consideration, and reviews the results from [12] on resource allocation via convex optimization. The regularization is introduced in Section III, showing how a partial minimization reduces the problem to an equivalent, smooth reduced Lagrangian. Section IV presents the corresponding saddle point gradient dynamics and its convergence proof. Some ideas on implementation of these laws without prior knowledge of the load are presented in Section V, together with illustrative simulations. Conclusions are given in Section VI, and some proofs are relayed to the Appendix.

II. Model description and prior work

We consider a distributed computing system with several clusters of servers, indexed by $j = 1, \dots, N$. Each cluster may be regarded as occupying a different geographical location or, alternatively, as a group of servers within a single facility that share a common physical rack. The number of active servers at cluster j is denoted by s_j, and it may be changed over time: for example, by switching servers between an active and a sleep mode.

All instances at the same cluster have the same standard processing capacity, and thus are interchangeable for each arriving task. However, service differentiation may appear between different cluster/task matchings, due to the *data locality* issue. Namely, the amount of data a server must retrieve from a remote cluster, to perform a certain task, will impact its processing speed. We model this by classifying tasks into different types: $i = 1, \dots, M$, as in [16] the type of a task reflects the location of all data required to perform the task. We let μ_{ij} denote the units of type i tasks that one server from cluster j can perform per unit of time; the matrix $\mu \in \mathbb{R}_+^{M \times N}$ is assumed given.

The arrival rate of type i tasks is denoted by λ_i; let $\lambda \in \mathbb{R}_+^M$ be the vector of these rates, which is typically unknown, and often time varying. The decision variables to be controlled over time are the following:

- The number of active servers s_j across clusters. The vector of cluster sizes is denoted by $s \in \mathbb{R}_+^N$ and its selection is called the *right sizing* problem.

This work was partially supported by ANII–Uruguay under grant FCE_1_2017_1_136748.

E-mail: goldsztajn@ort.edu.uy.

- The rates λ_{ij} in which traffic of type i is split between clusters, subject to the conservation constraint

$$\sum_{j=1}^{N} \lambda_{ij} = \lambda_i \quad \forall\ i = 1, \ldots, M.$$

  $\Lambda \in \mathbb{R}_+^{M \times N}$ is the matrix formed by the rates λ_{ij}, its selection is called the *load balancing* problem.
- The aggregate service rate r_{ij} that tasks of type i receive from cluster j. This means, the units of type i tasks that are performed at cluster j per unit of time. Providing this aggregate service requires the allocation of r_{ij}/μ_{ij} servers. The constraint on the aggregate server resources at each cluster is:

$$\sum_{i=1}^{M} \frac{r_{ij}}{\mu_{ij}} = s_j \quad \forall\ j = 1, \ldots, N.$$

  The matrix with entries r_{ij} is denoted by $R \in \mathbb{R}_+^{M \times N}$ and its selection is called the *scheduling* problem.

In the *fluid* models to be considered in this paper, all the above variables are taken to be real-valued. This is a suitable approximation for large-scale systems. In a similar vein, we use a fluid model for the queue q_{ij} of type i tasks present at cluster j; its evolution is given by

$$\dot{q}_{ij} = [\lambda_{ij} - r_{ij}]^+_{q_{ij}}\,. \tag{1}$$

This is simply flow balance plus a saturation to maintain non-negative queues. We use the positive projection notation:

$$[z]^+_q = \begin{cases} z & \text{if} \quad q > 0 \quad \text{or} \quad z \geq 0, \\ 0 & \text{if} \quad q = 0 \quad \text{and} \quad z < 0; \end{cases}$$

the projection is called active in the latter case. A fundamental requirement of any useful control policy is that the queues remain *stable*, i.e. bounded in time.

If we let Q denote the matrix with entries q_{ij}, a compact matrix notation for equation (1) is

$$\dot{Q} = [\Lambda - R]^+_Q. \tag{2}$$

Notation $\langle \cdot, \cdot \rangle$ will be used for the standard inner product of matrices, and $||\cdot||$ for the corresponding Frobenius norm.

A. Optimization approach

Much of the previous literature on task-cluster assignment concerns the case where the service capacity of clusters is fixed: s_j is constant over time for all j. An interesting question in this case is to characterize the capacity region of the system: the space of arrival rate vectors λ for which there exist a load balancing Λ and a scheduling R that yield stable queues. It is shown in [16], using queueing theory tools, that a combination of JSQ load balancing and MW scheduling is throughput optimal: it stabilizes the full capacity region. The delay performance of this policy is also studied in [16] under heavy traffic assumptions.

Our model differs from the above in allowing the number of servers s_j to be controlled over time. The rationale is that there is a large underlying capacity that, if activated, could stabilize any practical arrival rate vector, but one would like to keep active only the "right" amount. This right sizing problem has been studied for a single server class [7], [8], [11], but to our knowledge the multi-class, multi-cluster setting had not been addressed prior to our recent work [12].

Typically there is a cost (e.g. energy) associated with active server instances, so we would like to stabilize the load in the cheapest way, exploiting the available degrees of freedom. The following optimization problem from [12] expresses this objective:

Problem 1: Minimize $c(s) = \sum_{j=1}^{N} c_j(s_j)$,

$$\text{subject to} \ \sum_{j=1}^{N} \lambda_{ij} = \lambda_i,\ i = 1, \ldots, M; \tag{3a}$$

$$\sum_{i=1}^{M} \frac{r_{ij}}{\mu_{ij}} = s_j,\ j = 1, \ldots, N; \tag{3b}$$

$$0 \leq \lambda_{ij} \leq r_{ij}\ \forall\ i, j. \tag{3c}$$

We assume that $c_j(s_j)$ are increasing, strictly convex, and differentiable with a locally Lipschitz derivative.

As in the cited precedent from communication networks [4], [15], we aim to use convex optimization to obtain useful decompositions of this problem.

B. JSQ/MW policy via duality and gradient dynamics

We briefly review our results in [12], using duality to decompose the problem into suitable load balancing, scheduling and right sizing policies. Introduce the Lagrangian

$$L(\Lambda, R, s, Q) := \sum_j c_j(s_j) + \sum_{i,j} q_{ij}\,(\lambda_{ij} - r_{ij})\,,$$

where only the constraints $\lambda_{ij} - r_{ij} \leq 0$ from Problem 1 have been dualized. We have named the Lagrange multipliers q_{ij} because they will correspond to queues in the gradient dynamics to be introduced below.

The remaining constraints in (3) remain implicit; minimizing now over the variables (Λ, R) where $\Lambda \geq 0$ satisfies (3a), $R \geq 0$ satisfies (3b), leads to the reduced Lagrangian

$$\begin{aligned} \bar{L}(s, Q) &:= \min_{\Lambda, R} L(\Lambda, R, s, Q) \\ &= c(s) + \min_{\Lambda} \left\{ \sum_{i,j} q_{ij} \lambda_{ij} \right\} - \max_{R} \left\{ \sum_{i,j} q_{ij} r_{ij} \right\}. \end{aligned}$$

This partial minimization defines instantaneous load balancing and scheduling rules, decoupled in Λ and R. In particular, the second term is minimized by any traffic split $\Lambda^*(Q)$ that sends all type i tasks to the clusters j for which q_{ij} is minimum; if the latter correspond to queues, this is exactly the Join the Shortest Queue load balancing policy. Similarly, the last term is maximized by the schedules $R^*(s, Q)$ that assign the servers of cluster j to the tasks i which maximize $\mu_{ij} q_{ij}$, a Max-Weight scheduling policy.

To complete the optimization we used in [12] the saddle point gradient dynamics [1], [6] for the reduced Lagrangian, which are shown to have the form:

$$\dot{s}_j = \left[-\frac{\partial \bar{L}}{\partial s_j}\right]_{s_j}^{+} = \left[\max_i\{\mu_{ij}q_{ij}\} - c_j'(s_j)\right]_{s_j}^{+}, \tag{4a}$$

$$\dot{q}_{ij} = \left[\frac{\partial \bar{L}}{\partial q_{ij}}\right]_{q_{ij}}^{+} = \left[\lambda_{ij}^* - r_{ij}^*\right]_{q_{ij}}^{+}. \tag{4b}$$

Here $\Lambda^*(Q)$ and $R^*(s,Q)$ are as discussed before, the right-hand side of (4b) is a supergradient of $\bar{L}$, and the positive projections ensure non-negative cluster sizes and multipliers.

If we compare equations (1) and (4b), we verify that the Lagrange multiplier q_{ij} corresponds to the queue of type i tasks at cluster j, justifying the preceding references to JSQ/MW as in [16]. The stability of (4) is proved in [12].

An undesirable feature of the JSQ/MW in this continuous formulation is that they exhibit "chattering" around the equilibrium point: for instance, when queues are approximately equal, the load balancing $\Lambda^*(Q)$ will switch rapidly between them, as illustrated in the simulation traces of Fig. 1; a similar behavior occurs with $R^*(s,Q)$.

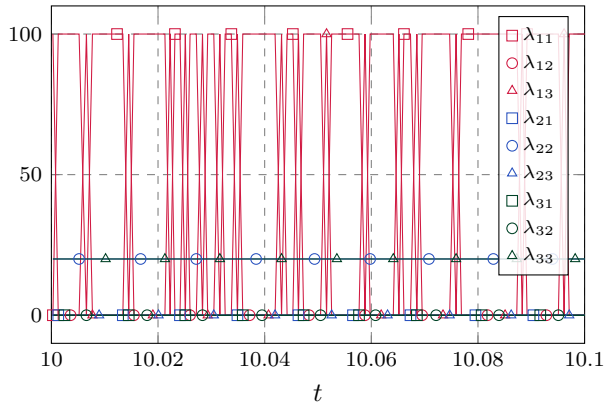

Fig. 1. JSQ load balancing $\Lambda^*(Q)$ once (s,Q) has reached an equilibrium.

These discontinuities make the mathematical analysis challenging[1], so we would like to find a smoother alternative in which the variables Λ and R settle into well defined equilibrium values. This motivates our next proposal, which consists of formulating an equivalent optimization problem where the reduced Lagrangian has continuous gradients.

III. Proximal regularization

Proximal methods have a long history in optimization [14], and have gained recent popularity for non-smooth or distributed problems [5], [13]. There are various such methods, all based on adding quadratic regularization terms to the cost. Our regularization below is of a similar nature to the one carried out in [9] for multipath routing problems. Specifically, we consider the modified problem with additional variables $\alpha, \beta \in \mathbb{R}^{M\times N}$:

[1]To some extent this limitation is due to the fluid model; in discrete terms, the alternation of jobs between queues is less problematic.

Problem 2: Minimize

$$c(s) + \frac{1}{2}\left|\left|\Lambda - \alpha\right|\right|^2 + \frac{1}{2}\left|\left|R - \beta\right|\right|^2$$

$$\text{subject to } \sum_{j=1}^{N} \lambda_{ij} = \lambda_i,\ i = 1,\dots,M; \tag{5a}$$

$$\sum_{i=1}^{M} \frac{r_{ij}}{\mu_{ij}} = s_j,\ j = 1,\dots,N; \tag{5b}$$

$$0 \le \lambda_{ij} \le r_{ij}\ \forall i,j. \tag{5c}$$

Since the constraints (5) are the same as (3), not involving the new variables, this is clearly equivalent to Problem 1. More precisely, both problems have the same infimum, and (Λ, R, s) is an optimum of (1) if and only if $(\Lambda, R, s, \alpha, \beta)$ is an optimum of (2) with $\alpha = \Lambda$ and $\beta = R$.

As in the previous section, we will derive saddle point dynamics for a reduced Lagrangian of this problem. In this case, the gradients of the reduced Lagrangian will be continuous, avoiding the chattering exhibited by JSQ/MW.

A. Reduced Lagrangian

As before, let us dualize Problem 2 with respect to the constraints $\lambda_{ij} - r_{ij} \le 0$, introducing the multipliers q_{ij}. The corresponding Lagrangian is

$$\begin{aligned} L(\Lambda, R, s, \alpha, \beta, Q) &:= c(s) + \sum_{i,j} q_{ij}\left(\lambda_{ij} - r_{ij}\right) \\ &+ \frac{1}{2}\sum_{i,j}\left(\lambda_{ij} - \alpha_{ij}\right)^2 + \frac{1}{2}\sum_{i,j}\left(r_{ij} - \beta_{ij}\right)^2. \end{aligned} \tag{6}$$

We now perform the partial minimization in Λ and R, with the aim of obtaining instantaneous load balancing and scheduling policies as before. The reduced Lagrangian is

$$\bar{L}(s,\alpha,\beta,Q) := \min_{\Lambda,R} L(\Lambda, R, s, \alpha, \beta, Q).$$

Completing the squares in the right-hand side of (6), and taking the minimum with respect to Λ and R, we see that

$$\begin{aligned} \bar{L}(s,\alpha,\beta,Q) = c(s) &+ \sum_{i,j}\left[(\alpha_{ij} - \beta_{ij})q_{ij} - q_{ij}^2\right] \\ &+ \min_{\Lambda}\left\{\frac{1}{2}\sum_{i,j}\left(\alpha_{ij} - q_{ij} - \lambda_{ij}\right)^2\right\} \\ &+ \min_{R}\left\{\frac{1}{2}\sum_{i,j}\left(\beta_{ij} + q_{ij} - r_{ij}\right)^2\right\}. \end{aligned}$$

The first minimum is taken across all Λ in the set

$$\Delta_\Lambda := \left\{\Lambda \in \mathbb{R}_+^{M\times N} : \sum_{j=1}^{N}\lambda_{ij} = \lambda_i,\ \forall\ i = 1,\dots,M\right\}.$$

The minimizer may be regarded as the closest point in this set to $\alpha - Q$, in the Frobenius norm. Since Δ_Λ is closed and convex, it defines a projection

$$\pi_\Lambda(X) := \operatorname*{argmin}_{Y\in\Delta_\Lambda} \left|\left|X - Y\right|\right|.$$

Therefore, we now have the load balancing policy

$$\Lambda^*(\alpha, Q) := \pi_\Lambda(\alpha - Q). \tag{7}$$

Remark 1: Since the set Δ_Λ depends on λ, implementing the policy (7) would require knowledge of the arrival rates. This goes against our objective that the system should automatically scale to the incoming load, a deficiency which will be amended in Section V through a specific implementation proposal.

Introducing the notation $\circ$ for the Hadamard (componentwise) product of matrices, the constraints in R take the form $R \in \Delta_R(s) \circ \mu$, with

$$\Delta_R(s) = \left\{ X \in \mathbb{R}_+^{M\times N} : \sum_{i=1}^{M} x_{ij} = s_j \ \forall\ j = 1, \dots, N \right\}.$$

Hence, the optimal schedule R^* is the point in $\Delta_R(s) \circ \mu$ that lies closest to $\beta + Q$. Again, the sets $\Delta_R(s) \circ \mu$ are closed and convex, so the projections

$$\pi_R(s, X) := \operatorname*{argmin}_{Y \in \Delta_R(s)\circ\mu} ||X - Y||$$

are well defined for each s. Then, the optimal schedule is

$$R^*(s, \beta, Q) := \pi_R(s, \beta + Q). \tag{8}$$

Consider now the real-valued functions

$$d_\Lambda(X) := \frac{1}{2}\,||X - \pi_\Lambda(X)||^2 \quad \text{and}$$
$$d_R(s, X) := \frac{1}{2}\,||X - \pi_R(s, X)||^2 \quad \forall\ X \in \mathbb{R}^{M\times N}.$$

These compute the squared distance to Δ_Λ and $\Delta_R(s) \circ \mu$, respectively, and the reduced Lagrangian is given by

$$\bar{L}(s, \alpha, \beta, Q) = c(s) + \sum_{i,j} \left[(\alpha_{ij} - \beta_{ij})q_{ij} - q_{ij}^2\right] + d_\Lambda(\alpha - Q) + d_R(s, \beta + Q).$$

Theorem 2: $\bar{L}$ is convex in (s, α, β), concave in Q, and differentiable, with locally Lipschitz continuous gradients:

$$\frac{\partial \bar{L}}{\partial s} = \nabla c(s) + \nabla_s d_R(s, \beta + Q);$$
$$\frac{\partial \bar{L}}{\partial \alpha} = \alpha - \Lambda^*(\alpha, Q);$$
$$\frac{\partial \bar{L}}{\partial \beta} = \beta - R^*(s, \beta, Q);$$
$$\frac{\partial \bar{L}}{\partial Q} = \Lambda^*(\alpha, Q) - R^*(s, \beta, Q).$$

The proof is given in the Appendix, based on a study of projections over a scaled simplex.

B. Equivalence of saddle points

Before proposing a gradient dynamics to seek saddle points of $\bar{L}$, we show that these are in one-to-one correspondence with the primal-dual optima of (2).

Proposition 3: $(\hat\Lambda, \hat R, \hat s, \hat\alpha, \hat\beta, \hat Q)$ is a saddle point of L if and only if $(\hat s, \hat\alpha, \hat\beta, \hat Q)$ is a saddle point of $\bar{L}$ such that

$$\Lambda^*(\hat\alpha, \hat Q) = \hat\Lambda \quad \text{and} \quad R^*(\hat s, \hat\beta, \hat Q) = \hat R.$$

Here the word saddle point is used under the convention that the variables s and Q lie in $\mathbb{R}_+^N$ and $\mathbb{R}_+^{M\times N}$, respectively.

Proof: Suppose that $(\hat\Lambda, \hat R, \hat s, \hat\alpha, \hat\beta, \hat Q)$ is a saddle point of L. Then $\Lambda^*(\hat\alpha, \hat Q) = \hat\Lambda$ and $R^*(\hat s, \hat\beta, \hat Q) = \hat R$ because

$$L(\hat\Lambda, \hat R, \hat s, \hat\alpha, \hat\beta, \hat Q) = \min_{\Lambda, R} L(\Lambda, R, \hat s, \hat\alpha, \hat\beta, \hat Q).$$

To show that $(\hat s, \hat\alpha, \hat\beta, \hat Q)$ is a saddle point of $\bar{L}$, note that

$$\begin{aligned}\bar{L}(\hat s, \hat\alpha, \hat\beta, \hat Q) &= L(\hat\Lambda, \hat R, \hat s, \hat\alpha, \hat\beta, \hat Q)\\ &\le L(\Lambda^*(\alpha, \hat Q), R^*(s, \beta, \hat Q), s, \alpha, \beta, \hat Q)\\ &= \bar{L}(s, \alpha, \beta, \hat Q)\end{aligned}$$

for all (s, α, β), and that for all Q we have

$$\begin{aligned}\bar{L}(\hat s, \hat\alpha, \hat\beta, \hat Q) &= L(\hat\Lambda, \hat R, \hat s, \hat\alpha, \hat\beta, \hat Q)\\ &\ge L(\hat\Lambda, \hat R, \hat s, \hat\alpha, \hat\beta, Q) \ge \bar{L}(\hat s, \hat\alpha, \hat\beta, Q).\end{aligned}$$

Suppose now that $(\hat s, \hat\alpha, \hat\beta, \hat Q)$ is saddle point of $\bar{L}$, and let $\hat\Lambda := \Lambda^*(\hat\alpha, \hat Q)$ and $\hat R := R^*(\hat s, \hat\beta, \hat Q)$. We first note that

$$\begin{aligned}L(\hat\Lambda, \hat R, \hat s, \hat\alpha, \hat\beta, \hat Q) &= \bar{L}(\hat s, \hat\alpha, \hat\beta, \hat Q)\\ &\le \bar{L}(s, \alpha, \beta, \hat Q) \le L(\Lambda, R, s, \alpha, \beta, Q)\end{aligned}$$

for all $(\Lambda, R, s, \alpha, \beta)$. Also, since $\hat Q$ maximizes $\bar{L}(\hat s, \hat\alpha, \hat\beta, \cdot)$ over $\mathbb{R}_+^{M\times N}$, then we have

$$\left[\hat\Lambda - \hat R\right]_{\hat Q}^{+} = \left[\frac{\partial \bar{L}}{\partial Q}(\hat s, \hat\alpha, \hat\beta, \hat Q)\right]_{\hat Q}^{+} = 0.$$

As a result, for all $Q \in \mathbb{R}_+^{M\times N}$ we have

$$\langle Q - \hat Q, \hat\Lambda - \hat R\rangle = \langle Q - \hat Q, \hat\Lambda - \hat R - \left[\hat\Lambda - \hat R\right]_{\hat Q}^{+}\rangle \le 0;$$

here $\langle x - y, z - [z]_y^+\rangle \le 0$ for all $x, y \in \mathbb{R}_+^n$ and all $z \in \mathbb{R}^n$. From this inequality we conclude that

$$\begin{aligned}L(\hat\Lambda, \hat R, \hat s, \hat\alpha, \hat\beta, Q) &= L(\hat\Lambda, \hat R, \hat s, \hat\alpha, \hat\beta, \hat Q) + \langle Q - \hat Q, \hat\Lambda - \hat R\rangle\\ &\le L(\hat\Lambda, \hat R, \hat s, \hat\alpha, \hat\beta, \hat Q)\end{aligned}$$

for all $Q \in \mathbb{R}_+^{M\times N}$, which completes the proof. ∎

IV. Saddle point gradient dynamics

We may find the saddle points of $\bar{L}$ by means of gradient descent in (s, α, β) and gradient ascent in Q, as follows:

$$\dot s_j = \left[-c_j'(s_j) - \frac{\partial}{\partial s_j} d_R(s, \beta + Q)\right]_{s_j}^{+}, \tag{9a}$$
$$\dot\alpha_{ij} = \lambda_{ij}^*(\alpha, Q) - \alpha_{ij}, \tag{9b}$$
$$\dot\beta_{ij} = r_{ij}^*(s, \beta, Q) - \beta_{ij}, \tag{9c}$$
$$\dot q_{ij} = \left[\lambda_{ij}^*(\alpha, Q) - r_{ij}^*(s, \beta, Q)\right]_{q_{ij}}^{+}. \tag{9d}$$

Here we used the expressions from Theorem 2; the field is locally Lipschitz continuous, except for the positive projections at the boundary points $s_j = 0,\ q_{ij} = 0$, to ensure that cluster sizes and multipliers (as before, corresponding to queues) remain non-negative.

Comparing with (4), we have removed the most problematic discontinuities: the chattering caused by JSQ and MW. The switching caused by the positive projections remains, but there are tools for dealing with projected dynamical systems of this kind, particularly in the setting of gradient dynamics arising from optimization problems [3].

It is easily seen that any equilibrium point of (9) is a saddle point of $\bar{L}$; we establish the existence of such equilibria.

Proposition 4: For a separable cost $c(s) = \sum_j c_j(s_j)$, with increasing c_j, solutions to Problems 1 and 2 exist. Furthermore, the equilibrium set of (9) is non-empty and (s, α, β, Q) is an equilibrium if and only if $(\alpha, \beta, s, \alpha, \beta, Q)$ is a primal-dual optimum of Problem 2.

Proof: We first note that there exists some (Λ_0, R_0, s_0) that is feasible for (3). Moreover, it is clear that Slater's condition holds for the two formulations of the problem, and thus strong duality holds in either case.

The set of those (Λ, R, s) feasible for Problem 1, such that $s_j \leq s_{0j}$ for all j, is compact and non-empty, so Problem 1 has a minimizer, and the same happens with Problem 2.

By Proposition 3, we know that $(\Lambda, R, s, \alpha, \beta, Q)$ is a primal-dual optimum of (2) if and only if (s, α, β, Q) is a saddle point of $\bar{L}$, and thus an equilibrium point of (9), with

$$\Lambda = \Lambda^*(\alpha, Q) \quad \text{and} \quad R = R^*(s, \beta, Q).$$

It is clear that $\alpha = \Lambda$ and $\beta = R$ in this case; for instance, from (9b) and (9c). We have shown that Problem 2 has a solution if the c_j are increasing, and since Slater's condition holds for this problem, there exists a primal-dual optimum. Particularly, the equilibrium set of (9) is non-empty. ∎

A. Global asymptotic stability

We will prove that each trajectory of (9) converges to an equilibrium point.

Suppose that a primal-dual optimum $(\hat{\Lambda}, \hat{R}, \hat{s}, \hat{\alpha}, \hat{\beta}, \hat{Q})$ of Problem 2 exists, and consider the Lyapunov function

$$\begin{aligned} V(s, \alpha, \beta, Q) = &\frac{1}{2} ||s - \hat{s}||^2 + \frac{1}{2} ||\alpha - \hat{\alpha}||^2 \\ &+ \frac{1}{2} \left|\left|\beta - \hat{\beta}\right|\right|^2 + \frac{1}{2} \left|\left|Q - \hat{Q}\right|\right|^2. \end{aligned} \tag{10}$$

Proposition 5: V is monotone along the trajectories of the dynamics (9). Specifically, $\dot{V}(s, \alpha, \beta, Q) \leq 0$ if $(s, Q) \geq 0$.

This first result follows from first order convexity and concavity conditions for the reduced Lagrangian; the arguments are standard, and may be found, for instance, in [6]. In particular we conclude the stability of the saddle points in the broad Lyapunov sense.

To establish asymptotic stability, an argument based on a LaSalle invariance principle may be given, based on the characterization of the set where $\dot{V} = 0$. In this regard, the following lemma is *not* standard, and its proof rather technical; it will be reported elsewhere.

Lemma 6: $\dot{V}(s, \alpha, \beta, Q) = 0$ and $(s, Q) \geq 0$ imply:

(a) $s = \hat{s}$,
(b) $\Lambda^*(\alpha, Q) = \alpha$ and $R^*(s, \beta, Q) = \beta$,
(c) $\bar{L}(s, \alpha, \beta, Q) = \bar{L}(\hat{s}, \hat{\alpha}, \hat{\beta}, \hat{Q})$.

Theorem 7: Suppose that a primal-dual optimum of Problem 2 exists. Then each solution of (9) converges to an equilibrium (s, α, β, Q) and $(\alpha, \beta, s, \alpha, \beta, Q)$ is primal-dual optimal for Problem 2.

Proof: We use a LaSalle invariance principle for Caratheodory solutions[2] of saddle point dynamics with projections, stated in [3]. It requires that $\bar{L}$ has locally Lipschitz gradients, which we have established in Theorem 2. It states that the solutions of (9) converge to the largest invariant subset contained in the closure cl$E$, where

$$E = \left\{(s, \alpha, \beta, Q) : \dot{V}(s, \alpha, \beta, Q) = 0,\ (s, Q) \geq 0\right\}.$$

This set E is contained in the set F of those (s, α, β, Q) satisfying the conditions of Lemma 6; the latter set is closed, since the projections Λ^* and R^* are continuous, as well as $\bar{L}$, which is differentiable. Therefore cl$E \subset F$.

Let M denote the largest invariant subset of cl$E$; fix some $(s, \alpha, \beta, Q) \in M$ and consider the solution $\eta(t)$ starting at this point; we will use the notation $\eta = (\eta_s, \eta_\alpha, \eta_\beta, \eta_Q)$.

Since $M \subset F$, we know that η satisfies the conditions of Lemma 6 at each instant of time, so we conclude that $\eta_s \equiv \hat{s}$ from (a), and $\dot{\eta}_\alpha \equiv \dot{\eta}_\beta \equiv 0$ from (b), (9b) and (9c). In particular, we have $\eta_\alpha \equiv \alpha$ and $\eta_\beta \equiv \beta$. Furthermore,

$$c(\hat{s}) + \langle \eta_Q, \alpha - \beta \rangle \equiv \bar{L}(\eta) \equiv \bar{L}(\hat{s}, \hat{\alpha}, \hat{\beta}, \hat{Q}). \tag{11}$$

Here the last identity follows from (c), whereas the first is a consequence of

$$\alpha \equiv \eta_\alpha \equiv \Lambda^*(\eta_\alpha, \eta_Q) \quad \text{and} \quad \beta \equiv \eta_\beta \equiv R^*(\eta_s, \eta_\beta, \eta_Q).$$

Differentiating equation (11) with respect to t, we see that

$$\langle [\alpha - \beta]^+_{\eta_Q}, \alpha - \beta \rangle \equiv \langle \dot{\eta}_Q, \alpha - \beta \rangle \equiv 0.$$

It is easy to check that $\langle [x]^+_y, x \rangle = \left|\left|[x]^+_y\right|\right|^2$ for all vectors x and y. Therefore, we conclude that

$$||\dot{\eta}_Q||^2 \equiv \left|\left|[\alpha - \beta]^+_{\eta_Q}\right|\right|^2 \equiv 0.$$

This proves that M consists exclusively of equilibrium points of (9). It is easy to conclude from the continuity of Λ^* and R^* that the equilibrium set is closed, hence M is closed too.

Now let $\eta(t)$ be *any* trajectory of (9). Proposition 5 implies that $\eta(t)$ is bounded, so its omega-limit set is non-empty, and the LaSalle principle implies that $\eta(t)$ converges to the closed set M; so the omega-limit set is made of equilibrium points of (9), which are saddle points of $\bar{L}$. Now applying Proposition 5, the distance from $\eta(t)$ to any given equilibrium is non-increasing. As a result, the omega-limit set must contain exactly one equilibrium point. ∎

[2]These are the absolutely continuous functions that satisfy the differential equation almost everywhere with respect to the Lebesgue measure.

V. Implementation and simulation

From an implementation standpoint, the aim is to distribute the control between the different agents in the system: a central task *dispatcher* in charge of load balancing; and, at each cluster location, a controller of both task *scheduling* and server population *right sizing*. Looking at our dynamics (9), the variables (Λ, α) naturally reside at the load balancer, and the rest at the cluster locations. Feedback information of the queues Q to the load balancer is naturally required.

However, a difficulty already noted in Remark 1, is that to implement the control law $\Lambda^*(\alpha, Q)$ using equation (7), requires knowledge of the arrival rates λ_i of all traffic types. This goes against the desired automatic adaptation of our system to uncertainty or variability in the exogenous load.

Nevertheless, it is possible to suppress this dependence through a change of variables. To this end, define

$$\gamma_{ij} = \frac{\alpha_{ij}}{\lambda_i} \quad \text{and} \quad p^*_{ij}(\alpha, Q) = \frac{\lambda^*_{ij}(\alpha, Q)}{\lambda_i},$$

and let γ and $P^*(\alpha, Q)$ denote the respective matrices. Note that p^*_{ij} represents the *fraction* of type i traffic that is routed to cluster j when the load balancing Λ^* is used.

Introduce the notation Δ_a for the simplex

$$\Delta_a = \left\{ x \in \mathbb{R}^N_+ : \sum_{j=1}^N x_j = a \right\}. \tag{12}$$

For each matrix $X \in \mathbb{R}^{M\times N}$ let X_i denote its i-th row; from equation (7) we conclude that

$$\begin{aligned} P^*_i(\alpha, Q) &= \lambda_i^{-1} \operatorname*{argmin}_{x\in\Delta_{\lambda_i}} ||\alpha_i - Q_i - x|| \\ &= \operatorname*{argmin}_{y\in\Delta_1} ||\alpha_i - Q_i - \lambda_i y|| \\ &= \operatorname*{argmin}_{y\in\Delta_1} \left|\left|\gamma_i - \lambda_i^{-1} Q_i - y\right|\right|. \end{aligned} \tag{13}$$

Also, dividing both sides of (9b) by λ_i, we obtain

$$\dot\gamma_{ij} = p^*_{ij} - \gamma_{ij}; \tag{9b'}$$

this suggests using (γ, P) as control variables at the dispatcher, instead of (α, Λ). Now, the computation of P^* in (13) requires knowledge of the ratios q_{ij}/λ_i. For this purpose, introduce the variables:

- $q_i = \sum_j q_{ij}$, total number of type i tasks in the system.
- τ_i, mean queueing delay experienced by type i tasks; using Little's law, $\tau_i = q_i/\lambda_i$. In practice, we may tag some tasks and measure their queueing delay by requesting a timed report of entrance to a server. Then we can estimate τ_i as the running average of these measurements over some window.

Writing

$$\frac{q_{ij}}{\lambda_i} = \frac{q_{ij}}{q_i}\frac{q_i}{\lambda_i} = \frac{q_{ij}}{\sum_{j=1}^N q_{ij}}\tau_i,$$

we see that P^* in (13) can be computed with knowledge of (γ, Q, τ); therefore (13)-(9b') may replace (7)-(9b) as an implementation for the load balancer, without requiring knowledge of the arrival rates λ_i.

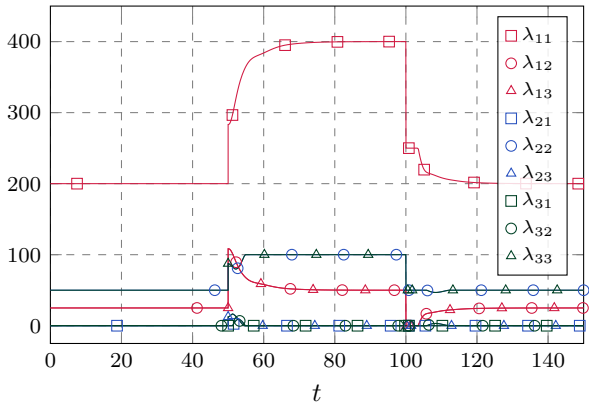

Fig. 2. Traffic split resulting from the routing probabilities P^*. Namely, the plot shows the rates $\lambda^*_{ij} = p^*_{ij}\lambda_i$.

On the cluster side, implementation is decentralized:

- The rates $r^*_{ij}(s, \beta, Q)$ and the dynamics (9c) only depend on information $(s_j,\ \beta_{ij},\ q_{ij})$ local to cluster j.
- The dynamics (9a) only depend on information that is local to cluster j. Indeed, the projection terms $\nabla_s d_R(s, \beta + Q)$ decouple across j.

The Appendix contains information on how to compute efficiently projections onto a simplex (see Proposition 8), as required for P^* and R^*; and also formulas for the derivatives $\partial_{s_j} d_R$ (see Proposition 9).

A. Simulation experiments

We implemented in Matlab the control rules for $(P, R, s, \beta, \gamma, Q)$. The chosen parameters were:

$$\mu = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix},$$

and the arrival rates were time-varying, of the form

$$\lambda(t) = \begin{bmatrix} 250 \\ 50 \\ 50 \end{bmatrix} \mathbf{1}_{t\in[0,50)\cup(100,150]} + \begin{bmatrix} 500 \\ 100 \\ 100 \end{bmatrix} \mathbf{1}_{t\in[50,100]}.$$

The traffic split trajectories $\lambda^*_{ij} = p^*_{ij}\lambda_i$, that result from the routing probabilities P^*, are shown in Fig. 2. We see that the rates λ^*_{ij} adopt a well defined equilibrium soon after the load changes. We note the absence of chattering, as compared to the situation of Fig. 1.

We also simulated a discrete system, using a continuous time Markov chain for arrival/service times, and estimating the mean queueing delays τ_i as explained above. More precisely, for each task type i, we measure the queueing delay of 1 out of 10 tasks and we take the average over a window of 30 tasks.

The evolution of cluster sizes is shown in Fig. 3. Here we see how the fluid dynamics provide an excellent macroscopic approximation to the discrete system's behavior, capturing both equilibrium values and transients. We further see, both in Fig. 2 and Fig. 3, how our control rules react quickly to variations of the load, stabilizing each new equilibrium point.

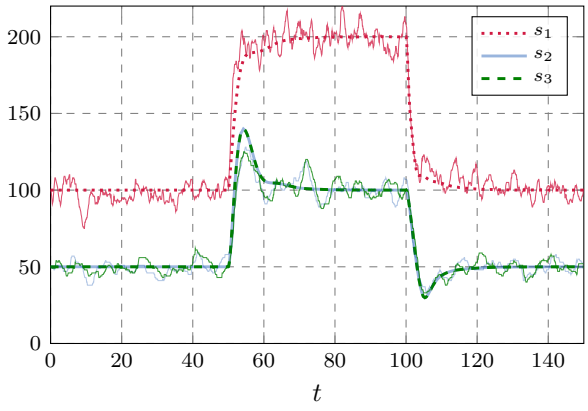

Fig. 3. Evolution of all cluster sizes s over time. Both the fluid dynamics and the discrete Markovian simulation are represented.

VI. Conclusions

This paper continues a line of research on using convex optimization tools to design controllers to manage simultaneously the load balancing, scheduling and capacity right-sizing for a cloud computing system. In particular, we used Lagrange duality to obtain useful decompositions of the problem, and saddle point gradient dynamics for its solution.

Our previous work motivated the introduction of switching policies, such as JSQ and MW, for load-balancing and scheduling, which are difficult to analyze with continuous time methods. Motivated by this, we introduced here a proximal regularization of the cost function, which led to a new, smooth alternative for the saddle point dynamics. The analysis has technical difficulties due to the appearance of projections onto a simplex and the fact that the cost function is not strictly convex; the latter complicates the asymptotic convergence proof for the saddle point dynamics.

We also described a procedure to implement these laws with available information, and tested in simulation the performance of these methods for situations outside our theory, such as time-varying loads.

Appendix

To prove Theorem 2 we must differentiate the functions $d_\Lambda(X)$ and $d_R(s,X)$ that measure the squared distance to a generalized simplex in matrix space.

A. Projection and squared distance to a simplex

We work first with the standard simplex in $\mathbb{R}^n$, Δ_a from (12) and its scaled version $\Delta_a \circ \mu$, where $\mu = (\mu_j) \in \mathbb{R}^n_{++}$ is a vector of fixed positive parameters, and $\circ$ denotes the componentwise product of vectors. Since $\Delta_a \circ \mu$ is compact and convex, the projection operator

$$\pi(a,x) := \operatorname*{argmin}_{y \in \Delta_a \circ \mu} ||x-y||$$

is well defined (gives a unique point). The squared distance function is given by

$$d(a,x) := \frac{1}{2}||x-\pi(a,x)||^2 = \frac{1}{2}\min_{y\in\Delta_a\circ\mu}||x-y||^2.$$

Introducing the convex indicator function

$$I_{\Delta_a\circ\mu}(x) := \begin{cases} \infty & \text{if} \quad x \notin \Delta_a \circ \mu, \\ 0 & \text{if} \quad x \in \Delta_a \circ \mu, \end{cases}$$

$d(a,x)$ can be identified with the Moreau-Yosida regularization (see [13]) of $I_{\Delta_a\circ\mu}(x)$; namely,

$$d(a,x) = \min_{y\in\mathbb{R}^n}\left\{ I_{\Delta_a\circ\mu}(y) + \frac{1}{2}||x-y||^2 \right\}.$$

This function is known to be differentiable in x, with gradient

$$\frac{\partial d}{\partial x}(a,x) = x - \pi(a,x). \tag{14}$$

We characterize first the projection operator in this case.

Proposition 8: For each $x \in \mathbb{R}^n$ and $a \geq 0$ we have

$$\pi(a,x)_i = \left[x_i - \theta(a,x)\mu_i^{-1}\right]^+,$$

where $\theta(a,x) \in \mathbb{R}$ satisfies

$$\sum_i \mu_i^{-2}[x_i\mu_i - \theta(a,x)]^+ = a. \tag{15}$$

Proof: The projection $\pi(x,a)$ is the vector y that minimizes $\frac{1}{2}||y-x||^2$, subject to $y_i \geq 0$ and $\sum_i \frac{y_i}{\mu_i} = a$. We write the Lagrangian with respect to the last constraint:

$$L(y,\theta) = \sum_i \left[\frac{(y_i-x_i)^2}{2} + \theta\frac{y_i}{\mu_i}\right] - \theta a.$$

Minimizing L with respect to each $y_i \geq 0$ yields

$$y_i = [x_i - \theta/\mu_i]^+ = \mu_i^{-1}[x_i\mu_i - \theta]^+; \tag{16}$$

imposing the constraint in y leads to (15). ∎

We wish to give a formula for $\theta(a,x)$ in (15)[3], extending those in [2], which apply to the case $a=1$ and $\mu = \mathbf{1}$ (unscaled). We will state the formula and sketch its derivation.

The main idea is that for a given θ, the nonzero terms in (15) are those with $z_i := x_i\mu_i \geq \theta$. As $a \geq 0$ grows, satisfying (15) requires lowering θ, and more terms will enter the sum in decreasing order in z_i. In particular, when $n-k$ nonzero terms are incorporated, solving for θ yields

$$\theta_k^\sigma(a,x) := \left(\sum_{r=k+1}^n \frac{1}{\mu_{\sigma(r)}^2}\right)^{-1}\left(-a + \sum_{r=k+1}^n \frac{x_{\sigma(r)}}{\mu_{\sigma(r)}}\right);$$

here σ is a permutation of $\{0,1,\ldots,n\}$ such that $z_{\sigma(r)}$ is monotonically increasing[4]. The overall formula is:

$$\theta(a,x) := \sum_{k=0}^{n-1} \theta_k^\sigma(a,x)\mathbf{1}_{z_{\sigma(k)} < \theta_k^\sigma(a,x) \leq z_{\sigma(k+1)}}. \tag{17}$$

An alternate formula can be obtained with the indicator functions applying to intervals in a. For this purpose, let $a_0(x) = +\infty$ and let $a_n^\sigma(x) \leq \cdots \leq a_1^\sigma(x)$ be such that

$$z_{\sigma(k)} < \theta_k^\sigma(a,x) \leq z_{\sigma(k+1)} \iff a_{k+1}^\sigma(x) \leq a < a_k^\sigma(x)$$

[3] $\theta(a,x)$ is uniquely defined by (15) except at $a=0$; here we will choose the smallest compatible value, namely $\theta(0,x) = \max_i(x_i\mu_i)$.

[4] For consistency, let $z_0 = -\infty$. The permutation σ may not be unique, but the expression (17) for $\theta(a,x)$ does not depend on this choice.

holds for all $k = 0, \dots, n-1$. Specifically,

$$a_k^\sigma(x) = \sum_{r=k}^{n} \left(\frac{x_{\sigma(r)}}{\mu_{\sigma(r)}} - \frac{z_{\sigma(k)}}{\mu^2_{\sigma(r)}} \right).$$

This leads to the expression below:

$$\theta(a,x) = \sum_{k=0}^{n-1} \theta_k^\sigma(a,x) \mathbf{1}_{a^\sigma_{k+1}(x) \le a < a^\sigma_k(x)}; \tag{18}$$

This formula is instrumental to the following result.

Proposition 9: The derivative of $a \longmapsto d(a,x)$ exists for each $a \ge 0$ and each $x \in \mathbb{R}^n$. Moreover, we have

$$\nabla_a d(a,x) = -\theta(a,x). \tag{19}$$

The proof is omitted due to space limitations. We now use this result to establish that $\nabla_a d$ is also a Lipschitz function.

Proposition 10: $\theta : \mathbb{R}_+ \times \mathbb{R}^n \longrightarrow \mathbb{R}$ is continuous and piecewise linear, and therefore locally Lipschitz.

Proof: Continuity of the function in (18) can be established by a careful consideration of the boundary between intervals, details are omitted. For piecewise linearity, for each $k = 0, \dots, n-1$, and each permutation σ, consider the set E_k^σ of all $(a,x) \in \mathbb{R}_+ \times \mathbb{R}^n$ such that

$$\begin{aligned} &\mu_{\sigma(1)} x_{\sigma(1)} \le \cdots \le \mu_{\sigma(n)} x_{\sigma(n)} \quad \text{and} \\ &a^\sigma_{k+1}(x) \le a < a^\sigma_k(x). \end{aligned}$$

The union of these $n! \times n$ polyhedrons is $\mathbb{R}_+ \times \mathbb{R}^n$ and $\theta(a,x) = \theta_k^\sigma(a,x)$ if $(a,x) \in E_k^\sigma$, for some permutation σ. This completes the proof because θ_k^σ is linear on E_k^σ. ∎

B. Proof of Theorem 2

Proof: Since $L(\,\cdot\,, Q)$ is convex, then

$$(s,\alpha,\beta) \longmapsto \bar{L}(s,\alpha,\beta,Q) = \min_{\Lambda,R} L(\Lambda,R,s,\alpha,\beta,Q)$$

is also convex. Moreover, if we fix $P, Q \in \mathbb{R}^{M \times N}$, and we let $\Lambda^*_Q = \Lambda^*(\alpha, Q)$ and $R^*_Q = R^*(s,\beta,Q)$, then

$$\begin{aligned} \bar{L}(s,\alpha,\beta,P) &= \min_{\Lambda,R} L(\Lambda,R,s,\alpha,\beta,P) \\ &\le L(\Lambda^*_Q, R^*_Q, s, \alpha, \beta, P) \\ &= \bar{L}(s,\alpha,\beta,Q) + \langle P - Q, \Lambda^*_Q - R^*_Q \rangle; \end{aligned} \tag{20}$$

Therefore, $\bar{L}(s,\alpha,\beta,\cdot)$ has a supergradient at any point, and is thus concave. In fact, differentiability to be shown below implies that $\Lambda^*_Q - R^*_Q$ is actually the gradient $\frac{\partial \bar{L}}{\partial Q}$.

The non-trivially differentiable components of $\bar{L}$ are the distance functions $d_\Lambda(\alpha - Q)$ and $d_R(s, \beta + Q)$; but these are covered by the results of the previous subsection.

Focusing first on $d_\Lambda(\cdot)$, we note that the constraints (3a) decouple over the rows of λ, and each row i involves the squared distance to a simplex (in this case, unscaled, and with constant $a = \lambda_i$). Hence, the gradient of d_Λ exists and from (14) it is given by

$$\nabla d_\Lambda(X) = X - \pi_\Lambda(X),$$

where $\pi_\Lambda(X)$ is defined, row by row, by the expressions of the preceding section, and is locally Lipschitz continuous.

For $d_R(s,X)$, the problem now decouples by *columns* of X; for each column j we have a *scaled* simplex $\Delta_a \circ \mu^{(j)}$ (here $\mu^{(j)}$ is the j-th column of the matrix μ), and the sum $a = s_j$ is now a variable. We can still follow (14) to compute

$$\nabla_X d_R(s,X) = X - \pi_R(s,X),$$

and (19) to compute $\nabla_s d_R(s,X)$; both are locally Lipschitz.

We have already found in (20) the expression for the gradient of $\bar{L}$ with respect to Q; we can also verify the given formulas for the gradients in α, β. For instance:

$$\begin{aligned} \frac{\partial \bar{L}}{\partial \beta} &= -Q + \nabla_X d_R(s, \beta + Q) \\ &= -Q + \beta + Q - \pi_R(s, \beta + Q) = \beta - R^*(s,\beta,Q). \end{aligned}$$

∎

References

[1] K. Arrow, L. Hurwitz, and H. Uzawa, *Studies in Linear and Non-Linear Programming*. Stanford University Press, Stanford, California, 1958.

[2] Y. Chen and X. Ye, "Projection onto a simplex," *arXiv preprint arXiv:1101.6081*, 2011.

[3] A. Cherukuri, E. Mallada, and J. Cortés, "Asymptotic convergence of constrained primal–dual dynamics," *Systems & Control Letters*, vol. 87, pp. 10–15, 2016.

[4] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition: A mathematical theory of network architectures," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 255–312, 2007.

[5] N. K. Dhingra, S. Z. Khong, and M. R. Jovanovic, "The proximal augmented lagrangian method for nonsmooth composite optimization," *IEEE Transactions on Automatic Control*, 2018.

[6] D. Feijer and F. Paganini, "Stability of primal–dual gradient dynamics and applications to network optimization," *Automatica*, vol. 46, no. 12, pp. 1974–1981, 2010.

[7] D. Goldsztajn, A. Ferragut, and F. Paganini, "Feedback control of server instances for right sizing in the cloud," in *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2018, pp. 749–756.

[8] M. Lin, A. Wierman, L. L. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," *IEEE/ACM Transactions on Networking (TON)*, vol. 21, no. 5, pp. 1378–1391, 2013.

[9] X. Lin and N. B. Shroff, "Utility maximization for communication networks with multipath routing," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 766–781, 2006.

[10] Y. Lu, Q. Xie, G. Kliot, A. Geller, J. R. Larus, and A. Greenberg, "Join-idle-queue: A novel load balancing algorithm for dynamically scalable web services," *Performance Evaluation*, vol. 68, no. 11, pp. 1056–1071, 2011.

[11] D. Mukherjee, S. Dhara, S. C. Borst, and J. S. van Leeuwaarden, "Optimal service elasticity in large-scale distributed systems," *ACM SIGMETRICS Performance Evaluation Review*, vol. 1, no. 1, p. 25, 2017.

[12] F. Paganini, D. Goldsztajn, and A. Ferragut, "An optimization approach to load balancing, scheduling and right sizing of cloud computing systems with data locality," in *58th IEEE Conference on Decision and Control*, 2019, to appear.

[13] N. Parikh, S. Boyd *et al.*, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[14] R. T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM journal on control and optimization*, vol. 14, no. 5, pp. 877–898, 1976.

[15] R. Srikant and L. Ying, *Communication networks: an optimization, control, and stochastic networks perspective*. Cambridge University Press, 2013.

[16] W. Wang, K. Zhu, L. Ying, J. Tan, and L. Zhang, "Maptask scheduling in mapreduce with data locality: Throughput and heavy-traffic optimality," *IEEE/ACM Transactions on Networking (TON)*, vol. 24, no. 1, pp. 190–203, 2016.