Heavy Traffic Optimal Resource Allocation Algorithms for Cloud Computing Clusters

Siva Theja Maguluri and R. Srikant
Department of ECE and CSL
University of Illinois at Urbana-Champaign
siva.theja@gmail.com; rsrikant@illinois.edu

Lei Ying
School of ECEE
Arizona State University
lying6@asu.edu

***Abstract*—Cloud computing is emerging as an important platform for business, personal and mobile computing applications. In this paper, we study a stochastic model of cloud computing, where jobs arrive according to a stochastic process and request resources like CPU, memory and storage space. We consider a model where the resource allocation problem can be separated into a routing or load balancing problem and a scheduling problem. We study the join-the-shortest-queue routing and power-of-two-choices routing algorithms with MaxWeight scheduling algorithm. It was known that these algorithms are throughput optimal. In this paper, we show that these algorithms are queue length optimal in the heavy traffic limit.**

***Index Terms*—Scheduling, load balancing, cloud computing, resource allocation.**

I. Introduction

Cloud computing services are emerging as an important resource for personal as well as commercial computing applications. Several cloud computing systems are now commercially available, including Amazon EC2 system [7], Google's AppEngine [1], and Microsoft's Azure [3]. A comprehensive survey on cloud computing can be found in [9], [2], [18].

In this paper, we focus on cloud computing platforms that provide infrastructure as service. Users submit requests for resources in the form of virtual machines (VMs). Each request specifies the amount of resources it needs in terms of processor power, memory, storage space, etc.. We call these requests jobs. The cloud service provider first queues these requests and then schedules them on physical machines called servers.

Each server has a limited amount of resources of each kind. This limits the number and types of jobs that can be scheduled on a server. The set of jobs of each type that can be scheduled simultaneously at a server is called a configuration. The convex hull of the possible configurations at a server is the capacity region of the server. The total capacity region of the cloud is then the Minkowski sum of the capacity regions of all servers.

The simplest architecture for serving the jobs is to queue them at a central location. In each time slot, a central scheduler chooses the configuration at each server and allocates jobs to the servers, in a preemptive manner. As pointed out in [16], this problem is then identical to scheduling in an ad hoc wireless network with interference constraints. In practice, however, jobs are routed to servers upon arrival. Thus, queues are maintained at each individual server. It was shown in [16] that join-the-shortest queue-type algorithms for routing, along with the MaxWeight scheduling algorithm [23] at each server is throughput optimal. The focus of this paper is to study the delay, or equivalently, the queue length performance of the algorithms presented in [16].

Characterizing the exact delay or queue length in general is difficult. So, we study the system in the heavy-traffic regime, i.e., when the exogenous arrival rate is close to the boundary of the capacity region. In this regime, for some systems, the multi-dimensional state of the system reduces to a single dimension, called state-space collapse. In [17], [24], a method was outlined to use the state-space collapse for studying the diffusion limits of several queuing systems. This procedure has been successfully applied to a variety of multiqueue models served by multiple servers [21], [11], [12], [4]. But these models assume that the system is work conserving, i.e., queued jobs are processed at maximum rate by each server. Stolyar [22], generalized this notion of state-space collapse and resource pooling to a generalized switch model, where it is hard to define work-conserving policies. This was used to establish the heavy traffic optimality of the MaxWeight algorithm.

Most of these results are based on considering a scaled version of queue lengths and time, which converges to a regulated Brownian motion, and then show sample-path optimality in the scaled time over a finite time interval. This then allows a natural conjecture about steady state distribution. In [8], the authors present an alternate method to prove heavy traffic optimality that is not only simpler, but shows heavy traffic optimality in unscaled time. In addition, this method directly obtains heavy-traffic optimality in steady state. The method consists of the following three steps.

(1) *Lower bound:* First a lower bound is obtained on the weighted sum of expected queue lengths by comparing with a single-server queue. A lower bound for the single-server queue, similar to the Kingman bound [14], then gives a lower bound to the original system.
(2) *State-space collapse*: The second step is to show that the state of the system collapses to a single dimension. Here, it is not a complete state-space collapse, as in the Brownian limit approach, but an approximate one. In particular, this step is to show that the queue length along a certain direction increases as the exogenous arrival rate gets closer to the boundary of the capacity region but the

queue length in any perpendicular direction is bounded.

(3) *Upper bound*: The state-space collapse is then used to obtain an upper bound on the weighted queue length. This is obtained by using a natural Lyapunov function suggested by the resource pooling. Heavy-traffic optimality can be obtained if the lower bounds and the upper bounds coincide.

In this paper, we apply the above three-step procedure to study the resource allocation algorithms presented in [16]. We briefly review the results in [16] now. Jobs are first routed to the servers, and are then queued at the servers, and a scheduler schedules jobs at each server. So, we need an algorithm that has two components, viz.,

1) a *routing algorithm* that routes new jobs to servers in each time slot (we assume that the jobs are assigned to a server upon arrival and they cannot be moved to a different server) and
2) a *scheduling algorithm* that chooses the configuration of each server, i.e., in each time slot, it decides which jobs to serve. Here we assume that jobs can be preempted, i.e., a job can be served in a time slot, and then be preempted if it is not scheduled in the next time slot. Its service can be resumed in the next time it is scheduled. Such a model is applicable in situations where job sizes are typically large.

It was shown in [16] that using the join-the-shortest-queue (JSQ) routing and MaxWeight scheduling algorithm is throughput optimal. In Section III, we show that this policy is queue length optimal in the heavy traffic limit. We use the three step procedure described above to prove the heavy traffic optimality. The lower bound in this case is identical to the case of the MaxWeight scheduling problem. However, state-space collapse does not directly follow from the corresponding results for the MaxWeight algorithm in [8] due to the additional routing step here. We use this to obtain an upper bound that coincides with the lower bound in the heavy traffic limit.

JSQ needs queue length information of all servers at the router. In practice, this communication overhead can be quite significant when the number of servers is large. An alternative algorithm is the power-of-two-choices routing algorithm. In each time slot, two servers are chosen uniformly at random and new arrivals are routed to the server with the shorter queue. It was shown in [16] that the power-of-two-choices routing algorithm with the MaxWeight scheduling is throughput optimal if all the servers are identical. Here, we show that the heavy-traffic optimality in this case is a minor modification of the corresponding result for JSQ routing and MaxWeight scheduling.

A special case of the resource allocation problem is when all the jobs are of same type. In this case, scheduling is not required at each server. The problem reduces to a routing-only problem which is well studied [19], [5], [6], [13], [20]. For reasons to be explained later, the results, from Section III cannot be applied in this case since the capacity region is along a single dimension (of the form $\lambda < \mu$). In Section IV, we show heavy traffic optimality of the power-of-two-choices routing algorithm. The lower and upper bounds in this case are identical to the case of JSQ routing in [8]. The main contribution here is to show state-space collapse, which is somewhat different compared to [8]. The results here complement the heavy-traffic optimality results in [6], [13] which were obtained using Brownian motion limits.

Note on Notation: The set of real numbers, the set of non-negative real numbers,

and the set of positive real numbers are denoted by $\mathbb{R}$, $\mathbb{R}_+$ and $\mathbb{R}_{++}$ respectively. We denote vectors in $\mathbb{R}^J$ or $\mathbb{R}^M$ by x, in normal font. We use bold font $\mathbf{x}$ to denote vectors in $\mathbb{R}^{JM}$. Dot product in the vector spaces $\mathbb{R}^J$ or $\mathbb{R}^M$ is denoted by $\langle x, y\rangle$ and the dot product in $\mathbb{R}^{JM}$ is denoted by $\langle \mathbf{x}, \mathbf{y}\rangle$.

II. System Model and Algorithm

Consider a discrete time cloud computing system as follows. There are M servers indexed by m. Each server has I different kinds of resources such as processing power, disk space, memory, etc.. Server m has $R_{i,m}$ units of resource i for $i \in \{1, 2, 3, ..., I\}$. There are J different types of jobs indexed by j. Jobs of type j need $r_{i,j}$ units of resource i for their service. A job is said to be of size D if it takes D units of time to finish its service. Let D_{max} be the maximum allowed service time.

Let $\mathcal{A}_j(t)$ denote the set of type-j jobs that arrive at the beginning of time slot t. Indexing the jobs in $\mathcal{A}_j(t)$ from 1 through $|\mathcal{A}_j(t)|$, we define $a_j(t) = \sum_{k\in\mathcal{A}_j(t)} D_k$, to be the overall size of the jobs in $\mathcal{A}_j(t)$ or the total time slots requested by the jobs in $\mathcal{A}_j(t)$. Thus, $a_j(t)$ denotes the total work load of type j that arrives in time slot t. We assume that $a_j(t)$ is a stochastic process which is i.i.d. across time slots, $\mathbb{E}[a_j(t)] = \lambda_j$ and $\Pr(a_j(t) = 0) > \epsilon_A$ for some $\epsilon_A > 0$ for all j and t. Many of these assumptions can be relaxed, but we make these assumptions for the ease of exposition. Second moments of the arrival processes are assumed to be bounded. Let $var[a_j(t)] = \sigma_j^2$, $\lambda = (\lambda_1, ...\lambda_J)$ and $\sigma = (\sigma_1, ...\sigma_J)$. We denote $\sigma^2 = (\sigma_1^2, ...\sigma_J^2)$.

In each time slot, the central router routes the new arrivals to one of the servers. Each server maintains J queues corresponding to the work loads of the J different types of jobs. Let $q_{j,m}(t)$ denote the total backlogged job size of the type j jobs at server m at time slot t.

Consider server m. We say that server m is in configuration $s = (s_1, s_2, ..., s_J) \in (\mathbb{Z}_+)^J$ if the server is serving s_1 jobs of type 1, s_2 jobs of type 2 etc. This is possible only if the server has enough resources to accommodate all these jobs. In other words, $\sum_{j=1}^{J} s_j r_{i,j} \leq R_{i,m} \forall i \in \{1, 2, ..., I\}$. Let s_{max} be the maximum number of jobs of any type that can be scheduled on any server. Let $\mathcal{S}_m$ be the set of feasible configurations on server m. We say that s is a maximal configuration if no other job can be accommodated i.e., for every j' $s + e_{j'}$ (where $e_{j'}$ is the unit vector along j') violates at least one of the resource constraints. Let $\mathcal{C}_m^*$ be the convex hull of the

maximal configurations of server m. Let $\mathcal{C}_m = \{s \in (\mathbb{R}_+)^J : s \leq s^* \text{ for some } s^* \in \mathcal{C}_m^*\}$. Here $s \leq s^*$ means $s_j \leq s_j^* \forall j \in \{1, 2, ..., J\}$. $\mathcal{C}_m$ can be thought of as the capacity region for server m. Note that if $\lambda \in interior(\mathcal{C}_m)$, there exists an $\epsilon > 0$ such that $\lambda(1+\epsilon) \in \mathcal{C}_m$. $\mathcal{C}_m$ is a convex polytope in the nonnegative quadrant of $\mathbb{R}^J$.

Define $\mathcal{C} = \sum_{m=1}^{M} \mathcal{C}_m = \{s \in (\mathbb{R}_+)^J : \exists s^m \in \mathcal{C}_m \ \forall \ m \text{ s.t. } s \leq \sum_{m=1}^{M} s^m\}$. Here s^m just denotes an element in $\mathcal{C}_m$ and not m^{th} power of s. Then, $\mathcal{C} = \sum_{m=1}^{M} \mathcal{C}_m$, where $\sum$ denotes the Minkowski sum of sets. So, $\mathcal{C}$ is again a convex polytope in the nonnegative quadrant of $\mathbb{R}^J$. So, $\mathcal{C}$ can be described by a set of hyperplanes as follows:

$$\mathcal{C} = \{s \geq 0 : \left\langle c^{(k)}, s \right\rangle \leq b^{(k)}, k = 1, ...K\}$$

where K is the number of hyperplanes that completely defines $\mathcal{C}$, and $(c^{(k)}, b^{(k)})$ completely defines the k^{th} hyperplane $\mathcal{H}^{(k)}$, $\left\langle c^{(k)}, s \right\rangle = b^{(k)}$. Since $\mathcal{C}$ is in the first quadrant, we have

$$||c^{(k)}|| = 1 \quad , c^{(k)} \geq 0, \quad b^{(k)} \geq 0 \quad for\ k = 1, 2, ...K.$$

It was shown in [16] that $\mathcal{C}$ is the capacity region of this system. Similar to $\mathcal{C}$, define $\mathcal{S} = \sum_{m=1}^{M} \mathcal{S}_m$.

Lemma 1: Given the k^{th} hyperplane $\mathcal{H}^{(k)}$ of the capacity region $\mathcal{C}$ (i.e., $\left\langle c^{(k)}, \lambda \right\rangle = b^{(k)}$), for each server m, there is a $b_m^{(k)}$ such that $\left\langle c^{(k)}, \lambda \right\rangle = b_m^{(k)}$ is the boundary of the capacity region $\mathcal{C}_m$, and $b^{(k)} = \sum_{m=1}^{M} b_m^{(k)}$. Moreover, for every set $\left\{\lambda_m^{(k)} \in \mathcal{C}_m\right\}_m$ such that $\lambda^{(k)} = \sum_{m=1}^{M} \lambda_m^{(k)}$ and $\lambda^{(k)} \in \mathcal{C}$ lies on the k^{th} hyperplane $\mathcal{H}^{(k)}$, we have that $\left\langle c^{(k)}, \lambda_m^{(k)} \right\rangle = b_m^{(k)}$. See [15] for proof of the lemma.

III. JSQ Routing and MaxWeight Scheduling

In this section, we will study the performance of JSQ routing with MaxWeight scheduling, as described in Algorithm 1.

Let $Y_{j,m}(t)$ denote the state of the queue for type-j jobs at server m, where $Y_{j,m}^i(t)$ is the (backlogged) size of the i^{th} type-j job at server m. It is easy to see that $\mathbf{Y}(t) = \{Y_{j,m}(t)\}_{j,m}$ is a Markov chain under the JSQ routing and MaxWeight scheduling. Then, $q_{j,m}(t) = \sum_i Y_{j,m}^i(t)$ is a function of the state $Y_{j,m}(t)$.

The queue lengths of work load evolve according to the following equation:

$$\begin{aligned} q_{j,m}(t+1) &= q_{j,m}(t) + a_{j,m}(t) - s_j^m(t) \\ &= q_{j,m}(t) + a_{j,m}(t) - \overline{s_j^m}(t) + \overline{u}_{j,m}(t) \end{aligned} \quad (1)$$

where $\overline{u}_{j,m}(t)$ is the unused service, given by $\overline{u}_{j,m}(t) = \overline{s_j^m}(t) - s_j^m(t)$, $\overline{s_j^m}(t)$ is the MaxWeight schedule and $s_j^m(t)$ is the actual schedule chosen by the scheduling algorithm and the arrivals are

Algorithm 1 JSQ Routing and MaxWeight Scheduling

1) *Routing Algorithm:* All the type j arrivals in a time slot are routed to the server with the smallest queue length for type j jobs, i.e., the server $m_j^* = \underset{m \in \{1,2,...M\}}{\arg\min}\ q_{j,m}$. Ties are broken uniformly at random.
2) *Scheduling Algorithm:* In each time slot, server m chooses a configuration $\overline{s^m} \in \mathcal{C}_m^*$ so that $\overline{s^m} = \underset{\overline{s^m} \in \mathcal{C}_m^*}{\arg\max} \sum_{j=1}^{J} \overline{s_j^m} q_{j,m}$. It then schedules up to a maximum of $\overline{s_j^m}$ jobs of type j (in a preemptive manner). Note that even if the queue length is greater than the allocated service, all of it may not be utilized, e.g., when the backlogged size is from a single job, since different chunks of the same job cannot be scheduled simultaneously. Denote the actual number of jobs chosen by s_j^m. Note that if $q_{j,m} \geq D_{max} s_{max}$, then $\overline{s_j^m} = s_j^m$.

$$a_{j,m}(t) = \begin{cases} a_j(t) & \text{if } m = m_j^*(t) \\ 0 & \text{otherwise} \end{cases}. \quad (2)$$

Here, m_j^* is the server chosen by the routing algorithm for type j jobs. Note that

$$\overline{u}_{j,m}(t) = 0 \text{ when } q_{j,m}(t) + a_{j,m}(t) \geq D_{max} s_{max}. \quad (3)$$

Also, denote $s = (s_j)_j$ where $s_j = \sum_{m=1}^{M} s_j^m$. Denote $\mathbf{a} = (a_{j,m})_{j,m}$, $\mathbf{s} = (s_j^m)_{j,m}$ and $\overline{\mathbf{u}} = (\overline{u}_{j,m})_{j,m}$. Also denote $\mathbf{1}$ to be the vector with 1 in all components.

It was shown in [16] that this algorithm is throughput optimal. Here, we will show that this algorithm is heavy traffic optimal.

Recall that the capacity region is bounded by K hyperplanes, each hyperplane $\mathcal{H}^{(k)}$ described by its normal vector $c^{(k)}$ and the value $b^{(k)}$. Then, for any $\lambda \in interior(\mathcal{C})$, we can define the distance of λ to $\mathcal{H}^{(k)}$ and the closest point, respectively, as

$$\begin{aligned} \epsilon^{(k)} &= \min_{s \in \mathcal{H}^{(k)}} ||\lambda - s|| \\ \lambda^{(k)} &= \lambda + \epsilon^{(k)} c^{(k)} \end{aligned} \quad (4)$$

where $\epsilon^{(k)} > 0$ for each k since $\lambda \in interior(\mathcal{C})$. We let $\epsilon \triangleq \left(\epsilon^{(k)}\right)_{k=1}^{K}$ denote the vector of distances to all hyperplanes. Note that $\lambda^{(k)}$ may be outside the capacity region $\mathcal{C}$ for some hyperplanes. So define

$$\mathcal{K}_\lambda \triangleq \left\{k \in \{1, 2, ...K\} : \lambda^{(k)} \in \mathcal{C}\right\}$$

$\mathcal{K}_\lambda$ identifies the set of *dominant hyperplanes* whose closest point to λ is on the boundary of the capacity region $\mathcal{C}$ hence is a feasible average rate for service. Note that for any $\lambda \in interior(\mathcal{C})$, the set $\mathcal{K}_\lambda$ is non-empty, and hence is well-defined. We further define

$$\mathcal{K}_\lambda^o \triangleq \left\{k \in \mathcal{K}_\lambda : \lambda^{(k)} \in Relint(\mathcal{F}^{(k)})\right\}$$

where $\mathcal{F}^{(k)}$ denotes the face on which $\lambda^{(k)}$ lies and $Relint$ means relative interior. Thus, $\mathcal{K}_\lambda^o$ is the subset of faces in $\mathcal{K}_\lambda$

for which the projection of λ is not shared by more than one hyperplane.

For $\epsilon \triangleq \left(\epsilon^{(k)}\right)_{k=1}^{K} > 0$, let $\lambda^{(\epsilon)}$ be the arrival rate in the interior of the capacity region so that its distance from the hyperplane $\mathcal{H}^{(k)}$ is $\epsilon^{(k)}$. Let $\lambda^{(k)}$ be the closest point to $\lambda^{(\epsilon)}$ on $\mathcal{H}^{(k)}$. Thus, we have $\lambda^{(k)} = \lambda^{(\epsilon)} + \epsilon^{(k)} c^{(k)}$. Let $\mathbf{q}^{(\epsilon)}(t)$ be the queue length process when the arrival rate is $\lambda^{(\epsilon)}$.

Define $\mathbf{c}^{(k)} \in \mathbb{R}_+^{JM}$, indexed by j, m as $\mathbf{c}_{j,m} = \frac{c_j}{\sqrt{M}}$. We expect that the state space collapse occurs along the direction $\mathbf{c}^{(k)}$. This is intuitive. For a fixed j, JSQ routing tries to equalize the queue lengths across servers. For a fixed server m, we expect that the state space collapse occurs along $c^{(k)}$ when approaching the hyperplane $\mathcal{H}^{(k)}$, as shown in [8]. Thus, for JSQ routing and MaxWeight, we expect that the state space collapse occurs along $\mathbf{c}^{(k)}$ in $\mathbb{R}^{JM}$.

For each $k \in \mathcal{K}^o_{\lambda^{(\epsilon)}}$, define the projection and perpendicular component of $\mathbf{q}^{(\epsilon)}$ to the vector $\mathbf{c}^{(k)}$ as follows:

$$\mathbf{q}_{||}^{(\epsilon,k)} \triangleq \left\langle \mathbf{c}^{(k)}, \mathbf{q}^{(\epsilon)} \right\rangle \mathbf{c}^{(k)}; \quad \mathbf{q}_{\perp}^{(\epsilon,k)} \triangleq \mathbf{q}^{(\epsilon)} - \mathbf{q}_{||}^{(\epsilon,k)}$$

In this section, we will prove the following proposition.

Proposition 1: Consider the cloud computing system described in Section II. Assume all the servers are identical, i.e., $R_{i,m} = R_i$ for all servers m and resources i and that JSQ routing and MaxWeight scheduling as described in Algorithm 1 is used. Let the exogenous arrival rate be $\lambda^{(\epsilon)} \in Interior(\mathcal{C})$ and the standard deviation of the arrival vector be $\sigma^{(\epsilon)} \in \mathbb{R}^J_{++}$ where the parameter $\epsilon = \left(\epsilon^{(k)}\right)_{k=1}^{K}$ is so that $\epsilon^{(k)}$ is the distance of $\lambda^{(\epsilon)}$ from the k^{th} hyperplane $\mathcal{H}^{(k)}$ as defined in (4). Then for each $k \in \mathcal{K}^o_{\lambda^{(\epsilon)}}$, the steady state queue length satisfies

$$\epsilon^{(k)} \mathbb{E}\left[\left\langle \mathbf{c}^{(k)}, \mathbf{q}(t) \right\rangle\right] \leq \frac{\zeta^{(\epsilon,k)}}{2} + B_2^{(\epsilon,k)}$$

where $\zeta^{(\epsilon,k)} = \frac{1}{\sqrt{M}} \left\langle \left(c^{(k)}\right)^2, \left(\sigma^{(\epsilon)}\right)^2 \right\rangle + \frac{\left(\epsilon^{(k)}\right)^2}{\sqrt{M}}$ $B_2^{(\epsilon,k)}$ is $o(\frac{1}{\epsilon^{(k)}})$.

In the heavy traffic limit as $\epsilon^{(k)} \downarrow 0$, this bound is tight, i.e.,

$$\lim_{\epsilon^{(k)} \downarrow 0} \epsilon^{(k)} \mathbb{E}\left[\left\langle \mathbf{c}^{(k)}, \mathbf{q}^{(\epsilon)} \right\rangle\right] = \frac{\zeta^{(k)}}{2} \tag{5}$$

where $\zeta^{(k)} = \frac{1}{\sqrt{M}} \left\langle \left(c^{(k)}\right)^2, (\sigma)^2 \right\rangle$.

We will prove this proposition by following the three step procedure described in Section I, by first obtaining a lower bound, then showing state space collapse and finally using the state space collapse result to obtain an upper bound.

A. Lower Bound

Since $\lambda^{(\epsilon)}$ is in the interior of $\mathcal{C}$, the process $\left\{\mathbf{q}^{(\epsilon)}(t)\right\}_t$ has a steady state distribution. We will obtain a lower bound on $\mathbb{E}\left[\left\langle \mathbf{c}^{(k)}, \mathbf{q}^{(\epsilon)} \right\rangle\right] = \mathbb{E}\left[\sum_{j=1}^{J} \frac{c_j^{(k)}}{\sqrt{M}} \left(\sum_{m=1}^{M} q_{jm}\right)\right]$ in steady state as follows.

Consider the single server queuing system, $\phi^{(\epsilon)}(t)$ with arrival process $\frac{1}{\sqrt{M}} \left\langle c^{(k)}, a^{(\epsilon)}(t) \right\rangle$ and service process given by $\frac{b^{(k)}}{\sqrt{M}}$ at each time slot. Then $\phi(t)$ is stochastically smaller than $\left\langle \mathbf{c}^{(k)}, \mathbf{q}(t)^{(\epsilon)} \right\rangle$. Thus, we have

$$\mathbb{E}\left[\left\langle \mathbf{c}^{(k)}, \mathbf{q}^{(\epsilon)} \right\rangle\right] \geq \mathbb{E}\left[\phi^{(\epsilon)}\right].$$

Using ϕ^2 as Lyapunov function for the single server queue and noting that the drift of it should be zero in steady state, one can bound $\mathbb{E}\left[\overline{\phi}^{(\epsilon)}\right]$ as follows [8]

$$\epsilon^{(k)} \mathbb{E}\left[\overline{\phi}^{(\epsilon)}\right] \geq \frac{\zeta^{(\epsilon,k)}}{2} - B_1^{(\epsilon,k)}.$$

where $\left(c^{(k)}\right)^2 = \left(\left(c_j^{(k)}\right)^2\right)_{j=1}^{J}$, $B_1^{(\epsilon,k)} = \frac{b^{(k)} \epsilon^{(k)}}{2}$ and $\zeta^{(\epsilon,k)} = \frac{1}{\sqrt{M}} \left\langle \left(c^{(k)}\right)^2, \left(\sigma^{(\epsilon)}\right)^2 \right\rangle + \frac{\left(\epsilon^{(k)}\right)^2}{\sqrt{M}}$.

Thus, in the heavy traffic limit as $\epsilon^{(k)} \downarrow 0$, we have that

$$\lim_{\epsilon^{(k)} \downarrow 0} \epsilon^{(k)} \mathbb{E}\left[\left\langle \mathbf{c}^{(k)}, \overline{\mathbf{q}}^{(\epsilon)} \right\rangle\right] \geq \frac{\zeta^{(k)}}{2} \tag{6}$$

where $\zeta^{(k)} = \frac{1}{\sqrt{M}} \left\langle \left(c^{(k)}\right)^2, (\sigma)^2 \right\rangle$.

B. State Space Collapse

In this subsection, we will show that there is a state space collapse along the direction $\mathbf{c}^{(k)}$. We know that as the arrival rate approaches the boundary of the capacity region, i.e., $\epsilon^{(k)} \to 0$, the steady state mean queue length $\mathbb{E}[||\mathbf{q}||] \to \infty$. We will show that as $\epsilon^{(k)} \to 0$, queue length projected along any direction perpendicular to $\mathbf{c}^{(k)}$ is bounded. So the constant does not contribute to the first order term in $\frac{1}{\epsilon^{(k)}}$, in which we are interested. Therefore, it is sufficient to study a bound on the queue length along $\mathbf{c}^{(k)}$. This is called state-space collapse.

Define the following Lyapunov functions.

$$V(\mathbf{q}) \triangleq \sum_{m=1}^{M} \sum_{j=1}^{J} q_{j,m}^2, \; W_{\perp}^{(k)}(\mathbf{q}) \triangleq \left\| \mathbf{q}_{\perp}^{(k)} \right\|, \; W_{||}^{(k)}(\mathbf{q}) \triangleq \left\| \mathbf{q}_{||}^{(k)} \right\|$$

$$V_{||}^{(k)}(\mathbf{q}) \triangleq \left\langle \mathbf{c}^{(k)}, \mathbf{q}^{(\epsilon)} \right\rangle^2 = \left\| \mathbf{q}_{||}^{(k)} \right\|^2 = \frac{1}{M} \left(\sum_{m=1}^{M} \sum_{j=1}^{J} q_{j,m} c_j \right)^2.$$

Define the drift of $V(\mathbf{q})$ as

$$\Delta V(\mathbf{q}) \triangleq \left[V(\mathbf{q}(t+1)) - V(\mathbf{q}(t))\right] \mathcal{I}(\mathbf{q}(t) = \mathbf{q}). \tag{7}$$

Define the drift of other lyapunov functions similarly.

To show the state space collapse happens along the direction of $\mathbf{c}^{(k)}$, we will need a result by Hajek [10], which gives a bound on $\left\| \mathbf{q}_{\perp}^{(k)} \right\|$ if the drift of $W_{\perp}^{(k)}(\mathbf{q})$ is negative. Here we use the following special case of the result by Hajek, as presented in [8].

Lemma 2: For an irreducible and aperiodic Markov Chain $\{X[t]\}_{t \geq 0}$ over a countable state space $\mathcal{X}$, suppose $Z : \mathcal{X} \to \mathbb{R}_+$ is a nonnegative-valued Lyapunov function. This drift of Z at X is assumed to satisfy the following conditions:

1) There exists an $\eta > 0$, and a $\kappa < \infty$ such that for all $X \in \mathcal{X}$ with $Z(X) \geq \kappa$,
$$\mathbb{E}[\Delta Z(X) | X[t] = X] \leq -\eta.$$
2) There exists a $D < \infty$ such that for all $X \in \mathcal{X}$,
$$\mathbb{P}\left(|\Delta Z(X)| \leq D\right) = 1.$$

Then, there exists a $\theta^\star > 0$ and a $C^\star < \infty$ such that

$$\limsup_{t\to\infty} \mathbb{E}\left[e^{\theta^\star Z(X[t])}\right] \leq C^\star.$$

If we further assume that the Markov Chain $\{X[t]\}_t$ is positive recurrent, then $Z(X[t])$ converges in distribution to a random variable $\bar{Z}$ for which

$$\mathbb{E}\left[e^{\theta^\star \bar{Z}}\right] \leq C^\star,$$

which directly implies that all moments of $\bar{Z}$ exist and are finite.

We also need Lemma 7 from [8], which gives the drift of $W_\perp^{(k)}(\mathbf{q})$ in terms of drifts of $V(\mathbf{q})$ and $V_{||}^{(k)}(\mathbf{q})$.

Lemma 3: Drift of $W_\perp^{(k)}$ can be bounded as follows:

$$\Delta W_\perp^{(k)}(\mathbf{q}) \leq \frac{1}{2\left\|\mathbf{q}_\perp^{(k)}\right\|}(\Delta V(\mathbf{q}) - \Delta V_{||}^{(k)}(\mathbf{q})) \quad \forall\ \mathbf{q} \in \mathbb{R}_+^J \tag{8}$$

Let us first consider the last term in this inequality. It can be shown that (see [15] for details),

$$\begin{aligned}
&\mathbb{E}\left[\triangle\ V_{||}^{(k)}(\mathbf{q}^{(\epsilon)})\middle|\ \mathbf{q}^{(\epsilon)}(t) = \mathbf{q}^{(\epsilon)}\right] + K_2 + \frac{2\epsilon^{(k)}}{\sqrt{M}}||\mathbf{q}_{||}^{(\epsilon,k)}|| \\
=&\frac{2||\mathbf{q}_{||}^{(\epsilon,k)}||}{\sqrt{M}}\sum_{m=1}^{M}\sum_{j=1}^{J} c_j\left(\lambda_j^{m(k)} - \mathbb{E}\left[s_j^{m(\epsilon)}(t)|\mathbf{q}(t) = \mathbf{q}^{(\epsilon)}\right]\right) \qquad (9)\\
\geq& 0 \qquad (10)
\end{aligned}$$

where $K_2 = 2JMs_{max}^2$. The last inequality is true because $\left\langle c^{(k)}, s^{m(\epsilon)}\right\rangle \leq b_m^{(k)}$ for every $s^{m(\epsilon)}(t) \in \mathcal{C}_m$ and from Lemma 1, for each m, there exists $b_m^{(k)}$ such that $\sum_{j=1}^{J} c_j\lambda_j^{m(k)} = b_m^{(k)}$.

Now, let us consider the first term in (8). By definition of $a_{j,m}(t)$, (2) we have

$$\begin{aligned}
\mathbb{E}\left[\sum_{m=1}^{M}\sum_{j=1}^{J} 2q_{j,m}^{(\epsilon)}a_{j,m}(t)\right] &= \sum_{j=1}^{J} 2q_{j,m_j^*}^{(\epsilon)}\lambda_j^{(\epsilon)} \\
&\leq \sum_{j=1}^{J} 2\lambda_j^{(\epsilon)}\sum_{m=1}^{M}\frac{q_{j,m}^{(\epsilon)}}{M}. \qquad (11)
\end{aligned}$$

By expanding the drift of $V(\mathbf{q}^{(\epsilon)})$ and using (3) and (11), it can be easily seen that

$$\begin{aligned}
&\mathbb{E}\left[\triangle\ V(\mathbf{q}^{(\epsilon)})|\mathbf{q}^{(\epsilon)}(t) = \mathbf{q}^{(\epsilon)}\right] \\
\leq& K' + \sum_{j=1}^{J} 2\lambda_j^{(\epsilon)}\sum_{m=1}^{M}\frac{q_{j,m}^{(\epsilon)}}{M} - \mathbb{E}\left[\sum_{m=1}^{M}\sum_{j=1}^{J} 2q_{j,m}^{(\epsilon)}s_j^m(t)\right] \qquad (12)\\
=& K_1 - \frac{2\epsilon^{(k)}}{\sqrt{M}}||\mathbf{q}_{||}^{(\epsilon,k)}|| + 2\sum_{m=1}^{M}\mathbb{E}\left[\min_{r^m\in\mathcal{C}_m}\sum_{j=1}^{J} q_{j,m}^{(\epsilon)}\left(\frac{\lambda_j^{(k)}}{M} - r_j^m\right)\right] \qquad (13)
\end{aligned}$$

where $K_1 = K' + 2JMD_{max}s_{max}^2$ and $K' = M\left(\sum_j\left(\lambda_j^2 + \sigma_j^2\right) + 2Js_{max}^2(1 + D_{max})\right)$. The last equality follows from (11) and using the fact that scheduling is done based on MaxWeight policy (see [15] for details),

Assuming *all the servers are identical*, we have that for each m, $\mathcal{C}_m = \{\lambda/M : \lambda \in \mathcal{C}\}$. So, $\mathcal{C}_m$ is a scaled version of $\mathcal{C}$. Thus, $\lambda^m = \lambda/M$. Since $k \in \mathcal{K}^o_{\lambda^{(\epsilon)}}$, we also have that $k \in \mathcal{K}^o_{\lambda^{m(\epsilon)}}$ for the capacity region $\mathcal{C}_m$. Thus, there exists $\delta^{(k)} > 0$ so that

$$\mathcal{B}_{\delta^{(k)}}^{(k)} \triangleq \mathcal{H}^{(k)} \cap \{r \in \mathbb{R}_+^J : ||r - \lambda^{(k)}/M|| \leq \delta^{(k)}\}$$

lies strictly within the face of $\mathcal{C}_m$ that corresponds to $\mathcal{F}^{(k)}$. (Note that this is the only instance in the proof of Proposition 1 that we use the assumption that all the servers are identical.) Call this face $\mathcal{F}_m^{(k)}$. Thus we have,

$$\begin{aligned}
&\mathbb{E}\left[\triangle\ V(\mathbf{q}^{(\epsilon)})|\mathbf{q}^{(\epsilon)}(t) = \mathbf{q}^{(\epsilon)}\right] - \left(K_1 - \frac{2\epsilon^{(k)}}{\sqrt{M}}||\mathbf{q}_{||}^{(\epsilon,k)}||\right) \\
\leq& 2\sum_{m=1}^{M}\mathbb{E}\left[\min_{r^m\in\mathcal{B}_{\delta^{(k)}}^{(k)}}\sum_{j=1}^{J} q_{j,m}^{(\epsilon)}\left(\frac{\lambda_j^{(k)}}{M} - r_j^m\right)\right] \qquad (14)\\
=& 2\sum_{m=1}^{M}\mathbb{E}\left[\min_{r^m\in\mathcal{B}_{\delta^{(k)}}^{(k)}}\sum_{j=1}^{J}\left(q_{j,m}^{(\epsilon)} - \left\|\mathbf{q}_{||}^{(k)}\right\|\frac{c_j}{\sqrt{M}}\right)\left(\frac{\lambda_j^{(k)}}{M} - r_j^m\right)\right] \qquad (15)\\
=& 2\sum_{m=1}^{M}\mathbb{E}\left[\min_{r^m\in\mathcal{B}_{\delta^{(k)}}^{(k)}}\sum_{j=1}^{J} q_{\perp j,m}^{(\epsilon,k)}\left(\frac{\lambda_j^{(k)}}{M} - r_j^m\right)\right] \\
=& -2\delta^{(k)}\sum_{m=1}^{M}\sqrt{\sum_{j=1}^{J}\left(q_{\perp j,m}^{(\epsilon,k)}\right)^2} \qquad (16)\\
\leq& -2\delta^{(k)}\sqrt{\sum_{m=1}^{M}\sum_{j=1}^{J}\left(q_{\perp j,m}^{(\epsilon,k)}\right)^2} \qquad (17)\\
=& -2\delta^{(k)}\left\|\mathbf{q}_\perp^{(k)}\right\|. \qquad (18)
\end{aligned}$$

Equation (15) is true because c is a vector perpendicular to the face $\mathcal{F}_m^{(k)}$ of $\mathcal{C}_m$ whereas both $\lambda^{(k)}/M$ and r^m lie on the face $\mathcal{F}_m^{(k)}$. So, $\frac{1}{\sqrt{M}}\left\|\mathbf{q}_{||}^{(k)}\right\|\sum_{j=1}^{J} c_j\left(\frac{\lambda_j^{(k)}}{M} - r_j^m\right) = 0$. Equation (16) is true because $\sum_{j=1}^{J} q_{\perp j,m}^{(\epsilon,k)}\left(\frac{\lambda_j^{(k)}}{M} - r_j^m\right)$ is inner product in $\mathbb{R}_+^J$ which is minimized when r^m is chosen to be on the boundary of $\mathcal{B}_{\delta^{(k)}}^{(k)}$ so that $\left(\frac{\lambda_j^{(k)}}{M} - r_j^m\right)_j$ points in the opposite direction to $\left(q_{\perp j,m}^{(\epsilon,k)}\right)_j$.

Now substituting (10) and (18) in (8), we get

$$\begin{aligned}
\mathbb{E}\left[\Delta W_\perp^{(k)}(\mathbf{q}^{(\epsilon)})|\mathbf{q}^{(\epsilon)}(t) = \mathbf{q}^{(\epsilon)}\right] &\leq \frac{K_1 + K_2}{2\left\|\mathbf{q}_\perp^{(\epsilon,k)}\right\|} - \delta^{(k)} \\
&\leq \frac{-\delta^{(k)}}{2}\text{if}\left(\left\|\mathbf{q}_\perp^{(\epsilon,k)}\right\| \geq \frac{K_1 + K_2}{\delta^{(k)}}\right).
\end{aligned}$$

Moreover, since the departures in each time slot are bounded and the arrivals are finite there is a $D < \infty$ such that

$\mathbb{P}(|\Delta Z(X)| \leq D)$ almost surely. Now, applying Lemma 2, we have the following proposition.

Proposition 2: Assuming all the servers are identical, for $\lambda^{(\epsilon)} \in \mathcal{C}$, under JSQ routing and MaxWeight scheduling, for every $k \in \mathcal{K}^o_{\lambda^{(\epsilon)}}$, there exists a set of finite constants $\{N_r^{(k)}\}_{r=1,2,...}$ such that $\mathbb{E}\left[\left\|\mathbf{q}_{\perp}^{(\epsilon,k)}\right\|^r\right] \leq N_r^{(k)}$ for all $\epsilon > 0$ and for each $r = 1, 2, ...$

As in [22], [8], note that $k \in \mathcal{K}^o_{\lambda^{(\epsilon)}}$ is an important assumption here. If $k \in \mathcal{K} \setminus \mathcal{K}^o_{\lambda^{(\epsilon)}}$, i.e., if the arrival rate approaches a corner point of the capacity region as $\epsilon^{(k)} \to 0$, then there is no constant $\delta^{(k)}$ so that $\mathcal{B}^{(k)}_{\delta^{(k)}}$ lies in the face $\mathcal{F}^{(k)}$. In other words, the $\delta^{(k)}$ depends on $\epsilon^{(k)}$ and so the bound obtained by Lemma 2 also depends on $\epsilon^{(k)}$.

Remark: As stated in Proposition 1, our results hold only for the case of identical servers, which is the most practical scenario. However, we have written the proofs more generally whenever we can so that it is clear where we need the identical server assumption. In particular, in this subsection, up to Equation (13), we do not need this assumption, but we have used the assumption after that, in analyzing the drift of $V(\mathbf{q})$. The upper bound in the next section is valid more generally if one can establish state-space collapse for the non-identical server case. However, at this time, this is an open problem.

C. Upper Bound

In this section, we will obtain an upper bound on the weighted queue length, $\mathbb{E}\left[\left\langle \mathbf{c}^{(k)}, \mathbf{q}^{(\epsilon)}\right\rangle\right]$ in steady state, and show that in the asymptotic limit as $\epsilon^{(k)} \downarrow 0$, this coincides with the lower bound.

Noting that the drift of $\Delta W_{||}^{(k)}$ is zero in steady state, it can be shown, as in Lemma 8 from [8] that in steady state, for any $\mathbf{c} \in \mathbb{R}_+^{JM}$, we have

$$\mathbb{E}\left[\langle \mathbf{c}, \mathbf{q}(t)\rangle \langle \mathbf{c}, \bar{\mathbf{s}}(t) - \mathbf{a}(t)\rangle\right] \tag{19}$$

$$= \frac{\mathbb{E}\left[\langle \mathbf{c}, \bar{\mathbf{s}}(t) - \mathbf{a}(t)\rangle^2\right]}{2} + \frac{\mathbb{E}\left[\langle \mathbf{c}, \bar{\mathbf{u}}(t)\rangle^2\right]}{2} \tag{20}$$

$$+ \mathbb{E}\left[\langle \mathbf{c}, \mathbf{q}(t) + \mathbf{a}(t) - \bar{\mathbf{s}}(t)\rangle \langle \mathbf{c}, \bar{\mathbf{u}}(t)\rangle\right] \tag{21}$$

An upper bound on $\mathbb{E}\left[\left\langle \mathbf{c}^{(k)}, \mathbf{q}^{(\epsilon)}\right\rangle\right]$ can be obtained by bounding each of the above terms. Due to space constraints, we just state the final result here. Complete proof is in [15]. We will first explain some terms that appear in the upper bound.

Define $\widetilde{\mathcal{C}}_m \subseteq \mathbb{R}_+^{JM}$ as $\widetilde{\mathcal{C}}_m = \mathcal{C}_1 \times ... \times \mathcal{C}_M$. Then, $\widetilde{\mathcal{C}}_m$ is a convex polygon.

Claim 1: Let $q^m \in \mathbb{R}_+^J$ for each $m \in \{1, 2, ...M\}$. Denote $\mathbf{q} = (q^m)_{m=1}^M \in \mathbb{R}_+^{JM}$. If, for each m, $(s^m)^*$ is a solution of $\max_{s \in \mathcal{C}_m} \langle q^m, s\rangle$ then $\mathbf{s}^* = ((s^m)^*)_m$ is a solution of $\max_{\mathbf{s} \in \widetilde{\mathcal{C}}_m} \langle \mathbf{q}, \mathbf{s}\rangle$. Therefore, choosing a MaxWeight schedule at each server is same as choosing a MaxWeight schedule from the convex polygon, $\widetilde{\mathcal{C}}_m$. See [15] for the proof of claim. Since there are a finite number of feasible schedules, given $\mathbf{c}^{(k)} \in \mathbb{R}_+^{JM}$ such that $||\mathbf{c}^{(k)}|| = 1$, there exists an angle $\theta^{(k)} \in (0, \frac{\pi}{2}]$ such that, for all $\mathbf{q} \in \left\{\mathbf{q} \in \mathbb{R}_+^{JM} : ||\mathbf{q}_{||}^{(k)}|| \geq ||\mathbf{q}|| \cos\left(\theta^{(k)}\right)\right\}$, (i.e., for all $\mathbf{q} \in \mathbb{R}_+^{JM}$ such that $\theta_{\mathbf{q}\mathbf{q}_{||}^{(k)}} \leq \theta^{(k)}$ where $\theta_{\mathbf{ab}}$ represents the angle between vectors $\mathbf{a}$ and $\mathbf{b}$), we have

$$\left\langle \mathbf{c}^{(k)}, \bar{\mathbf{s}}(t)\right\rangle \mathcal{I}\left(\mathbf{q}(t) = \mathbf{q}\right) = b^{(k)}/\sqrt{M}.$$

Define

$$\gamma^{(k)} = \min\left\{b^{(k)} - \langle c, r\rangle : r \in \mathcal{S} \setminus \mathcal{F}^{(k)}\right\}.$$

Then the upper bound can be shown to be [15]

$$\epsilon^{(k)} \mathbb{E}\left[\left\langle \mathbf{c}^{(k)}, \mathbf{q}(t)\right\rangle\right] \leq \frac{\zeta^{(\epsilon,k)}}{2} + B_2^{(\epsilon,k)}$$

where $B_2^{(\epsilon,k)} \to 0$ as $\epsilon^{(k)} \downarrow 0$. In the heavy traffic limit as $\epsilon^{(k)} \downarrow 0$, we have that

$$\lim_{\epsilon^{(k)} \downarrow 0} \epsilon^{(k)} \mathbb{E}\left[\left\langle \mathbf{c}^{(k)}, \overline{\mathbf{q}}^{(\epsilon)}\right\rangle\right] \leq \frac{\zeta^{(k)}}{2} \tag{22}$$

where $\zeta^{(k)}$ was defined as $\zeta^{(k)} = \frac{1}{\sqrt{M}}\left\langle \left(c^{(k)}\right)^2, (\sigma)^2\right\rangle$. Thus, (6) and (22) establish the first moment heavy-traffic optimality of JSQ routing and MaxWeight scheduling policy. The proof of Proposition 1 is now complete.

D. Power-of-Two-Choices Routing and MaxWeight Scheduling

JSQ routing needs complete queue length information at the router. In practice, this communication overhead can be considerable when the number of servers is large. An alternate algorithm is the power-of-two-choices routing algorithm.

In this algorithm, in each time slot t, for each type of job m, two servers $m_1^j(t)$ and $m_2^j(t)$ are chosen uniformly at random. All the type m job arrivals in this time slot are then routed to the server with the shorter queue length among these two, i.e., $m_j^*(t) = \underset{m \in \{m_1^j(t), m_2^j(t)\}}{\arg\min} q_{j,m}(t)$.

It was shown in [16] that power-of-two-choices routing algorithm with MaxWeight scheduling is throughput optimal if all the servers are identical. From the proof of throughput optimality, one obtains

$$\mathbb{E}\left[\triangle V(\mathbf{q}^{(\epsilon)}) | \mathbf{q}^{(\epsilon)}(t) = \mathbf{q}^{(\epsilon)}\right]$$

$$\leq K' + \sum_{j=1}^{J} 2\lambda_j \sum_{m=1}^{M} \frac{q_{j,m}^{(\epsilon)}}{M} - \mathbb{E}\left[\sum_{m=1}^{M}\sum_{j=1}^{J} 2q_{j,m}^{(\epsilon)} s_j^m(t)\right] \tag{23}$$

Note that this inequality is identical to (12), in the proof of state-space collapse of JSQ routing and MaxWeight scheduling policy. Also note that the remainder of the proof of state-space collapse and upper bound in Sections III-B and III-C is independent of the routing policy. Moreover, the proof of lower bound in Section III-A is also valid here. Thus, once we have the above relation, the proof of heavy traffic optimality of this policy is identical to that of JSQ routing and MaxWeight scheduling policy.

IV. Power-of-Two-Choices routing

In this section, we consider the power-of-two-choices routing algorithm, without any scheduling. This is a special case of the model considered in the previous section when all the jobs are of the same type. In this case, there is a single queue at each server and no scheduling is needed.

Note on Notation: In this section, since $J = 1$ here, we just denote all vectors (in $\mathbb{R}^M$) in bold font $\mathbf{x}$.

The result from previous section is not applicable here because of the following reason. In Proposition 1, a sequence of systems with arrival rate approaching a face of the capacity region, along its normal vector were considered. The normal vector of the face plays an important role in the state space collapse, and so the upper bound obtained is in terms of this normal. So, this result cannot be applied if the arrival rates were approaching a corner point where there is no common normal vector. In particular, the proof of state space collapse in Section III-B is not applicable here because one cannot define a ball $\mathcal{B}^{(k)}_{\delta^{(k)}}$ as in (14) at a corner point.

Let $\mathcal{A}(t)$ denote the set of jobs that arrive at the beginning of time slot t. Let D_k be the size of k^{th} job. We define $a(t) = \sum_{k\in\mathcal{A}(t)} D_k$, to be the overall size of the jobs in $\mathcal{A}(t)$ or the total time slots requested by the jobs. We assume that $a(t)$ is a stochastic process which is i.i.d. across time slots, $\mathbb{E}[a(t)] = \lambda$ and $\Pr(a(t) = 0) > \epsilon_a$ for some $\epsilon_a > 0$ for all t. Let $\sigma^2 = var[a(t)]$. Let $X(t)$ denote the servers chosen at time slot t. So, $X(t)$ can take one of ${}^MC_2$ values of the form (m, m') where $m, m' \in \mathbb{Z}_+$ and $1 \le m < m' \le M$. Here ${}^MC_2$ denotes the number of 2-combinations in a set of size M. Note that $X(t)$ is an i.i.d. random process with a uniform distribution over all possible values. Define ${}^MC_2$ different arrival processes denoted by $a_{m,m'}(t)$ with $1 \le m < m' \le M$ as follows. If $x(t) = (\hat{m}, \hat{m'})$, then

$$a_{m,m'}(t) = \begin{cases} a(t) & \text{for } m = \hat{m} \text{ and } m' = \hat{m'} \\ 0 & \text{otherwise} \end{cases}.$$

Thus, $\{a_{m,m'}(t)\}$ can be thought of as a set of correlated arrival processes. They are correlated so that only one of them can have a non-zero value at each time. Let $\lambda_{m,m'} = \mathbb{E}[a_{m,m'}(t)]$. Then $\lambda_{m,m'} = \frac{\lambda}{{}^MC_2}$. The arrivals in $a_{m,m'}(t)$ can be routed only to either server m or server m'. According to the power-of-two-choices algorithm, all the jobs are then routed to the server with smallest queue among m and m'. Ties are broken at random. Let $a_m(t)$ denote the arrivals to server m at time t after routing.

Let μ be the amount of service available in each time slot at each server. Not all of this service may be used either because the queue is empty or because different chunks of same job cannot be served simultaneously. Let $s_m(t)$ be the actual amount of service scheduled available in time slot t at server m. Let $u_m(t)$ denote the unused service which is defined as $u_m(t) = \mu - s_m(t)$. Let $q_m(t)$ denote the queue length at server m at time t, and let $\mathbf{q}(t)$ denote the vector $(q_1(t), q_2(t), ...q_M(t))$ Then, we have

$$q_m(t+1) = q_m(t) + a_m(t) - \mu + u_m(t).$$

Note that

$$u_m(t) = 0 \text{ whenever } q_m(t) + a_m(t) \ge D_{max}\mu. \quad (24)$$

We again follow the procedure used in the previous section to show heavy traffic optimality. Since power-of-two-choices algorithm tries to equalize any two randomly chosen queues, we expect that there is a state-space collapse along the direction where all queues are equal, similar to JSQ algorithm.

Let $\mathbf{c}_1 = \frac{1}{\sqrt{M}}(1, 1, ...1)$ be the unit vector in $\mathbb{R}^M$ along which we expect state-space collapse. Let $\mathbf{1}$ denote the vector (1,1,.....1). For any $\mathbf{Q} \in \mathbb{R}^M$, define $\mathbf{Q}_{||}$ to be the component of $\mathbf{Q}$ along $\mathbf{c}_1$, i.e., $\mathbf{Q}_{||} = \langle \mathbf{Q}, \mathbf{c}_1 \rangle \mathbf{c}_1$ where $\langle ., . \rangle$ denotes the canonical dot product. Thus, $\mathbf{Q}_{||} = \frac{\sum_m Q_m}{M}\mathbf{1}$. Define $\mathbf{Q}_\perp$ to be the component of $\mathbf{Q}$ perpendicular to $\mathbf{Q}_{||}$, i.e., $\mathbf{Q}_\perp = \mathbf{Q} - \mathbf{Q}_{||}$.

Define the Lyapunov functions $V_{||}(\mathbf{Q}) = ||\mathbf{Q}_{||}||^2 = \frac{\left(\sum_m Q_m\right)^2}{M}$ and $W_\perp(\mathbf{Q}) = ||\mathbf{Q}_\perp|| = \left[\sum_m Q_m^2 - \frac{\left(\sum_m Q_m\right)^2}{M}\right]^{\frac{1}{2}}$.

A. Lower Bound

Consider an arrival process with arrival rate $\lambda^{(\epsilon)}$ such that $\epsilon = M\mu - \lambda^{(\epsilon)}$. Let $\mathbf{q}^{(\epsilon)}(t)$ denote the corresponding queue length vector. Since the system is stabilizable, there exists a steady-state distribution of $\mathbf{q}^{(\epsilon)}(t)$. Again, lower bounding $(\sum_m \overline{\mathbf{q}}^{(\epsilon)})$ by a single queue length as in Section III-A, we have

$$\mathbb{E}\left[\sum_m \overline{\mathbf{q}}^{(\epsilon)}\right] \ge \frac{\left(\sigma^{(\epsilon)}\right)^2 + \epsilon^2}{2\epsilon} - B_1$$

where $B_1 = \frac{Ms_{max}}{2}$. Thus, in the heavy-traffic limit we have

$$\liminf_{\epsilon\to 0} \epsilon\mathbb{E}\left[\sum_m \overline{\mathbf{q}}^{(\epsilon)}\right] \ge \frac{\sigma^2}{2}. \quad (25)$$

B. State Space Collapse

For simplicity of notation, in this sub-section, we write $\mathbf{q}$ for $\mathbf{q}^{(\epsilon)}$. We will bound the drift of the Lyapunov function $W_\perp(\mathbf{Q})$, and again use Lemma 2 to obtain state space collapse. We again use (8) with $\mathbf{c}_1$ instead of $\mathbf{c}^{(k)}$ to get the drift of $W^{(k)}_\perp(\mathbf{q})$ in terms of drifts of $V(\mathbf{q})$ and $V^{(k)}_{||}(\mathbf{q})$.

It can be shown that the first term can be bounded [15],

$$\mathbb{E}\left[\triangle\, V_{||}(\mathbf{q})|\mathbf{q}(t) = \mathbf{q}\right] \ge -K_3 - 2\frac{\epsilon}{M}\left(\sum_m q_m\right) \quad (26)$$

where $K_3 = 2M\mu^2$.

Now, consider the first term in (8). Let p be a permutation of $(1, 2, ...M)$ so that $q_{p(1)} \le q_{p(2)} \le ..... \le q_{p(M)}$. Let p' be the inverse permutation. In other words, $p'(m)$ is the position of m in the permutation p. Let $q_{min} = q_{p(1)}$ and $q_{min} = q_{p(M)}$. Expanding $[\triangle\, V(\mathbf{q})|\mathbf{q}(t)]$ and using (24), it is easy to see that

$$\begin{aligned}
&\mathbb{E}\left[\triangle\, V(\mathbf{q})|\mathbf{q}(t) = \mathbf{q}\right] - \left(K_4 - 2\mu\sum_m q_m(t)\right) \\
\le&\mathbb{E}_X\mathbb{E}\left[\sum_m 2q_m(t)a_m(t)|\mathbf{q}(t) = \mathbf{q}, X(t) = i, j\right] \\
\le&\frac{2\lambda q_{min}}{{}^MC_2} + \frac{1}{{}^MC_2}\sum_{(i,j)\neq(p(1),p(M))} \mathbb{E}\left[q_i(t)a(t) + q_i(t)a(t)|X(t) = i, j\right]
\end{aligned}$$

$$= -\frac{\lambda}{{}^{M}C_2}(q_{max}-q_{min}) + \frac{2\lambda}{M}\sum_m q_m(t)$$

$$=2\mu\sum_m q_m(t) - 2\frac{\epsilon}{M}\sum_m q_m(t) - \frac{\lambda}{{}^{M}C_2}(q_{max}-q_{min})$$

where $K_4 = M(2\mu^2(D_{max}+1)+\sigma^2+\lambda^2)$. Note that

$$||\mathbf{q}_\perp|| = \sqrt{\sum_m \left(q_m - \frac{\sum_m q_m}{M}\right)^2} \leq \sqrt{M\left(q_{max}-q_{min}\right)^2}.$$

Thus, we have,

$$\mathbb{E}\left[\triangle\, V(\mathbf{q})|\mathbf{q}(t)=\mathbf{q}\right] \leq K_4 - 2\frac{\epsilon}{M}\sum_m q_m(t) - \frac{\lambda}{{}^{M}C_2}\frac{||\mathbf{q}_\perp||}{\sqrt{M}}$$

Substituting this and (26) in (8), we have

$$\mathbb{E}\left[\triangle\, W_\perp(\mathbf{q})\right] \leq \frac{K_3+K_4}{2||\mathbf{q}_\perp||} - \frac{\lambda}{{}^{M}C_2}\frac{1}{2\sqrt{M}}.$$

This means that we have negative drift for sufficiently large $W_\perp(\mathbf{q})$. Since the drift of $W_\perp(\mathbf{q})$ is finite with probability 1, using Lemma 2, there exist finite constants $\{N'_r\}_{r=1,2,...}$ such that $\mathbb{E}\left[||\overline{\mathbf{q}}^{(\epsilon)}||^r\right] \leq N'_r$ for each $r=1,2,....$

C. Upper Bound

The upper bound is again obtained by bounding each of the terms in (21). This is identical to the case of JSQ routing (Proposition 3 in [8]). So, we will not repeat the proof here, but just state the upper bound.

$$\mathbb{E}\left[\sum_m \overline{\mathbf{q}}^{(\epsilon)}\right] \geq \frac{\left(\sigma^{(\epsilon)}\right)^2+\epsilon^2}{2\epsilon} - B_2^{(\epsilon)}$$

where $B_2^{(\epsilon)} = M\sqrt{\frac{N_2 s_{max}}{\epsilon}} + \frac{s_{max}}{2}$. Thus, in heavy traffic limit, we have

$$\liminf_{\epsilon\to 0}\epsilon\mathbb{E}\left[\sum_m \overline{\mathbf{q}}^{(\epsilon)}\right] \geq \frac{\sigma^2}{2}.$$

This coincides with the heavy-traffic lower bound in (25). This establishes the first-moment heavy-traffic optimality of power-of-two choices routing algorithm.

V. Conclusions

We considered a stochastic model for load balancing and scheduling in cloud computing clusters. We studied the performance of JSQ routing and MaxWeight scheduling policy under this model. It was known that this policy is throughput optimal. We have shown that it is heavy traffic optimal when all the servers are identical. We also found that using the power-of-two-choices routing instead of JSQ routing is also heavy traffic optimal.

We then considered a simpler setting where the jobs are of the same type, so only load balancing is needed. It has been established by others using diffusion limit arguments that the power-of-two-choices algorithm is heavy traffic optimal. We presented a steady-state version of this result here using Lyapunov drift arguments.

VI. Acknowledgments

Research was funded in part by ARO MURI W911NF-08-1-0233 and NSF grant CNS-0963807.

References

[1] AppEngine. http://code.google.com/appengine/.

[2] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, et al. Above the clouds: A Berkeley view of cloud computing. 2009. Tech. Rep. UCB/eeCs-2009-28, EECS department, U.C. Berkeley.

[3] Azure. http://www.microsoft.com/windowsazure/.

[4] S. L. Bell and R. J. Williams. Dynamic scheduling of a parallel server system in heavy traffic with complete resource pooling: asymptotic optimality of a threshold policy. *Electronic J. of Probability*, pages 1044–1115, 2005.

[5] M. Bramson, Y. Lu, and B. Prabhakar. Randomized load balancing with general service time distributions. In *Proceedings of the ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, SIGMETRICS '10, pages 275–286, New York, NY, USA, 2010. ACM.

[6] H. Chen and H. Q. Ye. Asymptotic optimality of balanced routing, 2010. http://myweb.polyu.edu.hk/~lgtyehq/papers/ChenYe11OR.pdf.

[7] EC2. http://aws.amazon.com/ec2/.

[8] A. Eryilmaz and R. Srikant. Asymptotically tight steady-state queue length bounds implied by drift conditions. *Queueing Systems*, pages 1–49, 2012.

[9] I. Foster, Y. Zhao, I. Raicu, and S. Lu. Cloud computing and grid computing 360-degree compared. In *Grid Computing Environments Workshop, 2008. GCE'08*, pages 1–10, 2008.

[10] B. Hajek. Hitting-time and occupation-time bounds implied by drift analysis with applications. *Advances in Applied Probability*, pages 502–525, 1982.

[11] J. M. Harrison. Heavy traffic analysis of a system with parallel servers: Asymptotic optimality of discrete review policies. *Ann. App. Probab.*, pages 822–848, 1998.

[12] J. M. Harrison and M. J. Lopez. Heavy traffic resource pooling in parallel-server systems. *Queueing Systems*, pages 339–368, 1999.

[13] Y. T. He and D. G. Down. Limited choice and locality considerations for load balancing. *Performance Evaluation*, 65(9):670 – 687, 2008.

[14] J. F. C. Kingman. Some inequalities for the queue GI/G/1. *Biometrika*, pages 315–324, 1962.

[15] S. Maguluri, R. Srikant, and L. Ying. Heavy traffic optimal resource allocation algorithms for cloud computing clusters. Technical Report, http://arxiv.org/abs/1206.1264.

[16] S. T. Maguluri, R. Srikant, and L. Ying. Stochastic models of load balancing and scheduling in cloud computing clusters. In *Proc. IEEE Infocom.*, pages 702–710, 2012.

[17] M.Bramson. State space collapse with application to heavy-traffic limits for multiclass queueing networks. *Queueing Systems Theory and Applications*, pages 89 – 148, 1998.

[18] D. A. Menasce and P. Ngo. Understanding cloud computing: Experimentation and capacity planning. In *Proc. 2009 Computer Measurement Group Conf.*, 2009.

[19] M. Mitzenmacher. *The Power of Two Choices in Randomized Load Balancing*. PhD thesis, University of California at Berkeley, 1996.

[20] R. L. D. N. D. Vvedenskaya and F. I. Karpelevich. Queueing system with selection of the shortest of two queues: An asymptotic approach. *Problems of Information Transmission*, 32(1):15–27, 1996.

[21] M. I. Reiman. Some diffusion approximations with state space collapse. In *Proceedings of International Seminar on Modelling and Performance Evaluation Methodology, Lecture Notes in Control and Information Sciences*, pages 209–240, Berlin, 1983. Springer.

[22] A. Stolyar. MaxWeight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic. *Adv. Appl. Prob.*, 14(1), 2004.

[23] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Trans. Automat. Contr.*, 4:1936–1948, December 1992.

[24] R. J. Williams. Diffusion approximations for open multiclass queueing networks: Sufficient conditions involving state space collapse. *Queueing Systems Theory and Applications*, pages 27 – 88, 1998.