Analysis of Rate-Based Pull and Push Strategies with Limited Migration Rates in Large Distributed Networks

W. Minnebo and B. Van Houdt
Department of Mathematics and Computer Science
University of Antwerp - iMinds
Middelheimlaan 1, B-2020 Antwerp, Belgium
{wouter.minnebo,benny.vanhoudt}@uantwerpen.be

ABSTRACT

In this paper we analyze the performance of pull and push strategies in large homogeneous distributed systems where the number of job transfers per time unit is limited. Job transfer strategies which rely on lightly-loaded servers to attract jobs from heavily-loaded servers are known as pull strategies, whereas for push strategies the heavily loaded servers initiate the job transfers to lightly loaded servers. To this end, servers transmit probe messages to discover other servers that are able to take part in a job transfer.

Previous work on rate-based pull and push strategies focused on the impact of the probe rate on the mean job response time. In this paper we also limit the overall migration rate and show that any predefined migration rate can be matched by both the rate-based pull and push strategies. We present closed form formulas for the mean response time (as a function of the allowed probe and migration rate) and validate their accuracy by simulation.

We also introduce and analyze a new pull strategy and show that under high loads it is superior to the push strategies considered, while the push strategies offer only a very limited gain for medium to low load scenarios.

Categories and Subject Descriptors

C.4 [**Computer Systems Organization**]: Performance of Systems; D.4.8 [**Operating Systems**]: Performance

Keywords

Distributed computing, performance analysis, processor scheduling

1. INTRODUCTION

In order to optimally use the available resources in a distributed network it is desirable to be able to dynamically relocate jobs among a large number of processing nodes. Jobs may enter the network via one or multiple central dispatchers (e.g., [4,10,12,14,15]) or via the processing nodes themselves (e.g., [2, 3, 8, 13]). A central dispatcher will distribute the jobs among the nodes using some load balancing algorithm. In a more distributed approach, the nodes themselves will arrange for jobs to relocate after they are scheduled. Two approaches are common: push and pull. In a push variant (or work sharing) highly loaded nodes attempt to find lightly loaded nodes to migrate jobs to. Pull variants (or work stealing) reverse the roles, so that lightly loaded nodes try to attract work from the highly loaded nodes.

Several authors studied the performance of push and pull strategies. A comparison for a homogeneous distributed system with Poisson arrivals and exponential job lengths was presented in [1, 2] and extended to heterogeneous systems in [9, 11]. These studies showed that the pull strategy is superior under high load conditions, while the push strategy achieves a lower mean delay under low to moderate loads.

Nodes typically communicate by means of probe messages, exchanging information such as queue length. For simplicity we assume that sending/receiving probe messages is instantaneous and does not incur an extra computational or bandwidth cost. When a node wants to push or pull a job, it probes a random other node to see if a transfer between the nodes would be allowed.

Under a traditional pull or push strategy a server sends a maximum of L_p probes the instant its last job completes or the instant a job arrives when the server is already busy [1,2]. The fraction of queues sending probe messages is different, and as a result pull and push strategies achieve a different overall probe rate for the same load of the system. This makes a performance comparison biased, as sometimes the strategy with the higher probe rate is best [5].

In [5] rate-based pull and push variants are introduced that can match any predetermined probe rate R, allowing the comparison of pull and push strategies when they use the same number of probes. In these variants, probes are no longer sent at job arrival or completion times but at a fixed rate r as long as the server is idle (for pull) or has jobs waiting (for push). The main result in [5] showed that the rate-based push strategy results in a lower mean delay if and only if

$$\lambda < \frac{\sqrt{(R+1)^2 + 4(R+1)} - (R+1)}{2},$$

under the so-called infinite system model and that a hybrid pull/push strategy is always inferior to the pure pull or push strategy.

VALUETOOLS 2015, December 14-16, Berlin, Germany

DOI 10.4108/eai.14-12-2015.2262564

In [6] the model was extended with an extra parameter T, where a node is considered highly loaded if it has more than T jobs. This allowed the construction of the max-push strategy that extended the range of λ values where the push variants outperformed the pull strategy.

All prior work, including [5, 6], assumed zero cost for job transfers, which is not always realistic. When jobs are difficult to migrate, it would be desirable to be able to limit migrations to a predefined overall migration rate M, while not exceeding the predefined overall probe rate R.

This paper makes the following contributions:

1. We indicate how to set the parameter r (and T) of the push, pull and max-push strategy to match any predefined migration rate M.
2. We argue that setting $T = 1$ for the pull strategy is no longer optimal when an overall migration limit M is considered, as was the case in [6] and introduce a new pull strategy, called the conditional-pull strategy.
3. We show that the conditional-pull strategy is equivalent in stationary queue length distribution to the max-push variant when the overall probe rate R tends to infinity, i.e., when only an overall migration limit M is considered.
4. We consider a system where both an overall probe limit (R) and an overall migration limit (M) are imposed. For this system we compare the performance of push and pull strategies. We find that even for moderate R the conditional-pull performs almost as well as the max-push for low to moderate loads, and performs significantly better for higher loads.

The paper is structured as follows. Section 2 summarizes the rate-based strategies considered in this paper. In Section 3 we briefly summarize earlier work concerning rate-based pull and push strategies, and introduce an overall migration limit M for these strategies. Also, we derive an expression for the corresponding maximum probe rate $r_{both|M}$ for the rate-based push and pull, and rewrite the mean delay in an equivalent form. In Section 4 we adapt the max-push strategy to match M by finding the corresponding probe rate $r_{mp|M}$, after summarizing earlier work. Section 5 considers pull strategies with $T > 1$, and introduces the new conditional-pull strategy. It is shown that the conditional-pull is equivalent to the max-push strategy in case there is no probe limit R and only a migration limit M. In addition, the infinite system model describing the evolution of the conditional pull strategy is numerically validated, and argued to be the proper limiting process as the system size tends to infinity. Finally, we compare the mean delay of max-push and conditional-pull in Section 6.

2. RATE BASED STRATEGIES

We consider a continuous-time system of N queues, where each queue has a single server and infinite buffer. Each queue operates under Poisson job arrivals with rate $\lambda < 1$, and exponential service time with mean 1. Jobs are processed in a first-come-first-served order.

Traditional strategies send a maximum of L_p probes the instant a server's last job completes or the instant a job arrives when the server is already busy. In contrast, under rate-based strategies probes are no longer sent at job arrival or completion times but at a fixed rate r as long as the server is idle (pull) or has at least T jobs waiting (push). More formally, probe messages are transmitted by a server according to an interrupted Poisson process with rate r.

The strategies considered in this paper can be summarized as follows:

1. *Rate-based Push:* As soon as the queue length exceeds T, a server starts to generate probe messages according to a Poisson process with rate r. Whenever the queue length drops below T, this process is interrupted until the queue length exceeds T again. The node that is probed is selected at random and is only allowed to accept a job if it is idle.
2. *Rate-based Pull:* Whenever a server is idle it generates probe messages according to a Poisson process with rate r. This process is interrupted whenever the server is busy. The node that is probed is selected at random and is only allowed to transfer one of its jobs if its queue length exceeds T.
3. *Max-Push:* The instant a new job arrives at a queue with length T, probes are sent at an infinite rate. When $\lambda < 1$ this corresponds to stating that the job is instantaneously transferred to an empty server. A server with T jobs in its queue, generates probe messages according to a Poisson process with rate r. Whenever the queue length drops to $T - 1$, this process is interrupted as long as the queue length remains below T. The node that is probed is selected at random and is only allowed to accept a job if it is idle.
4. *Conditional-Pull:* Whenever a node is idle, the node will generate probe messages according to a Poisson process with rate r. This process is interrupted whenever the server becomes busy. The probed node is selected at random and the probe is always successful if there are at least T jobs waiting to be served, and successful with some probability p (matching M, see (26)) if there are exactly $T - 1$ jobs waiting to be served.

We do not consider hybrid strategies, which combine both push and pull behavior. These were proven to be inferior to a pure push or pull strategy when $T = 1$ [5, Theorem 4].

3. PULL AND PUSH STRATEGIES

Infinite system models and closed form solutions for both pull and push strategies were introduced in [5] and [6]. Before introducing new constraints and strategies, we briefly summarize the main findings of [5] and [6].

The evolution of both the rate-based pull and push strategy under the infinite system model is described by a set of ODEs denoted as $\frac{d}{dt}x(t) = F(x(t))$, where $x(t) = (x_1(t), x_2(t), \ldots)$ and $x_i(t)$ represents the fraction of the number of nodes with at least i jobs at time t.

From [6, Theorem 2 and 3], it is known that $\frac{d}{dt}x(t) = F(x(t))$ has a unique fixed point $\bar{\pi} = (\bar{\pi}_1, \bar{\pi}_2, \ldots)$ with $\sum_{i\geq 1} \bar{\pi}_i < \infty$ that is a global attractor, given by

$$\bar{\pi}_i = \frac{\lambda\left((1+r)\lambda^{i-1} - r\lambda^T\right)}{1 + r(1-\lambda^T)} \qquad 1 \leq i \leq T+1, \tag{1}$$

$$\bar{\pi}_i = \bar{\pi}_{T+1}\left(\frac{\lambda}{1+(1-\lambda)r}\right)^{i-T-1} \qquad i > T+1. \tag{2}$$

This fixed point is used in conjunction with Little's Law in [6, Corollary 1] to formulate the mean delay D_{both} of a job under the push or pull strategy:

$$D_{both} = \frac{1}{1-\lambda} - \frac{r\lambda^T\left(\frac{\lambda}{(1-\lambda)(1+r)} + T\right)}{1 + r(1-\lambda^T)}. \tag{3}$$

From the relationships $R = (1-\bar{\pi}_1)r_{pull|R}$ and $R = r_{push|R}\bar{\pi}_{T+1}$, we find

$$R = (1-\lambda)r_{pull|R}, \tag{4}$$

and

$$R = \frac{\lambda^{T+1}}{(1-\lambda^T) + 1/r_{push|R}}. \tag{5}$$

It follows that whenever $R > \lambda^{T+1}/(1-\lambda^T)$, the rate $r_{push|R}$ can be chosen arbitrarily large (i.e., $r_{push|R} = \infty$).

3.1 Limiting the Overall Migration Rate

When the overall migration rate is limited, the choice of r must satisfy this constraint. We first indicate how to set r to match M, and rewrite the formula for the mean delay. Then we show whether R or M is the strictest constraint for a given load λ.

Theorem 1. *Both rate-based pull and push strategies match a predefined migration rate M by letting the probe rate $r = r_{both|M}$, with $r_{both|M}$:*

$$r_{both|M} = \frac{M}{(\lambda(1-\lambda)+M)\lambda^T - M}. \tag{6}$$

For this setting, both strategies achieve the same mean delay.

Proof. The relationship (6) readily follows from the formulation of the overall migration rate for both rate-based strategies:

$$M_{both} = \frac{r(1-\lambda)\lambda^{T+1}}{r(1-\lambda^T)+1}. \tag{7}$$

From a push perspective this equation describes the fraction of nodes with queue length larger than or equal to $T+1$ ($\bar{\pi}_{T+1} = \lambda^{T+1}/(1+r(1-\lambda^T))$ from (1)) sending probes at rate r, succeeding with probability $(1-\lambda)$. For a pull strategy the overall migration rate is expressed as the fraction of empty queues $(1-\lambda)$ sending probes at rate r and succeeding when probing a queue of length $T+1$ or longer ($\bar{\pi}_{T+1} = \lambda^{T+1}/(1+r(1-\lambda^T))$ from (1)). □

Theorem 2. *The mean delay of rate-based pull and push strategies can be expressed as*

$$D_{both} = \frac{1}{1-\lambda}\left(1 - \frac{M_{both}}{\lambda}\left(T + \frac{\lambda}{(1-\lambda)(1+r)}\right)\right). \tag{8}$$

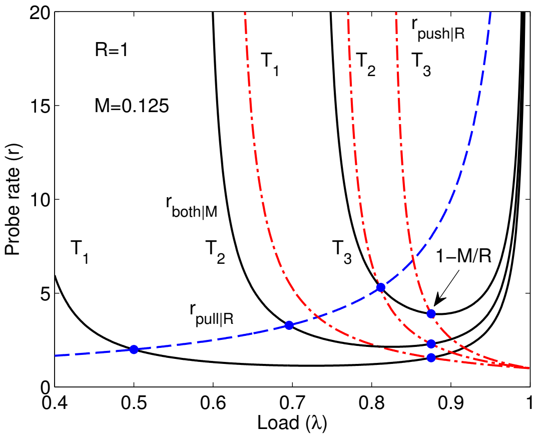

Figure 1: The probe rates r imposed by either the probe limit $R = 1$ for push (dot-dashed) and pull (dashed), or migration limit $M = 1/8$ (full), for $T = 1, 2, 3$. Note that $r_{pull|R}$ is independent of T.

Proof. Equation (8) follows by rewriting (7) to

$$\frac{r\lambda^T}{1+r(1-\lambda^T)} = \frac{M_{both}}{\lambda(1-\lambda)},$$

and substituting this expression in (3). □

The previous theorem shows that the improvement in mean delay compared to a standard M/M/1 queue, can be expressed as a migration frequency (M/λ) times a migration gain $(T + \lambda/((1-\lambda)(1+r)))$. The migration frequency denotes how many migrations per job take place on average. The migration gain quantifies the number of places in the queue the migrating job skips. All migrating jobs skip at least T places by construction of the strategy, and skip more places depending on the queue length of the job sender. The average number of places skipped above T equals the average number of customers in an M/M/1 queue with service rate $1 + r(1-\lambda)$, which equals $\lambda/((1-\lambda)(1+r))$.

Both the overall migration limit M and the overall probe limit R impose a maximum on r. In case $R \leq M$, all probes are allowed to generate a migration, so r will only be constrained by R. In any practical setting there will be more probes allowed than migrations, and the probe rate will be constrained by R or M depending on λ. An overview of the probe rates matching R or M, for both push and pull strategies with $T = 1, 2, 3$, is given in Figure 1.

To determine the range of λ in which each constraint is most strict, we determine the intersection of $r_{both|M}$ with $r_{push|R}$ and $r_{pull|R}$.

Lemma 1. *The probe rates $r_{push|R}$ and $r_{both|M}$ intersect at $\lambda = 0$ and $1 - M/R$ only, and both rates are positive for $\lambda = 1 - M/R$ if and only if $\left(1 - \frac{M}{R}\right)^T > \frac{R^2}{R-M+R^2}$.*

The proof is given in Appendix A of [7].

THEOREM 3. *For the rate-based push strategy r should be set as follows in order to respect both the probe rate R and migration rate M:*

1. *$R \geq \lambda^{T+1}/(1-\lambda^T)$ and $M \geq \lambda^{T+1}(1-\lambda)/(1-\lambda^T)$: r can be arbitrarily large.*
2. *$R \geq \lambda^{T+1}/(1-\lambda^T)$ and $M < \lambda^{T+1}(1-\lambda)/(1-\lambda^T)$: r can be at most $r_{both|M}$.*
3. *$R < \lambda^{T+1}/(1-\lambda^T)$ and $M \geq \lambda^{T+1}(1-\lambda)/(1-\lambda^T)$: r can be at most $r_{push|R}$.*
4. *$R < \lambda^{T+1}/(1-\lambda^T)$ and $M < \lambda^{T+1}(1-\lambda)/(1-\lambda^T)$: Let $\tau = \left(1-\frac{M}{R}\right)^T$ and $\upsilon = \frac{R^2}{R-M+R^2}$.*
    - *If $\tau > \upsilon$, r can be at most $r_{both|M}$ if $\lambda < 1-M/R$ and at most $r_{push|R}$ otherwise.*
    - *If $\tau \leq \upsilon$, r can be at most $r_{push|R}$.*

The proof is given in Appendix B of [7].

LEMMA 2. *If $M < \frac{R}{1+TR}$, $r_{both|M} - r_{pull|R}$ has a unique root λ_T in $(0,1)$, otherwise it has no roots in $(0,1)$.*

The proof is given in Appendix C of [7]. For $T = 1$, the unique root λ_T of Lemma 2 reduces to $\lambda_1 = \sqrt{M + M/R}$. However, there seems to be no closed form expression for general T.

THEOREM 4. *For the rate-based pull strategy r should be set as follows in order to respect both the probe rate R and migration rate M:*

1. *$M \geq \lambda^{T+1}(1-\lambda)/(1-\lambda^T)$: r can be at most $r_{pull|R}$.*
2. *$M < \lambda^{T+1}(1-\lambda)/(1-\lambda^T)$:*
    - *If the migration limit is sufficiently high ($M \geq \frac{R}{1+TR}$), then r can be at most $r_{pull|R}$.*
    - *If the migration limit is sufficiently low ($M < \frac{R}{1+TR}$), r can be at most $r_{pull|R}$ if $\lambda < \lambda_T$, and at most $r_{both|M}$ otherwise.*

The proof is given in Appendix D of [7].

4. MAX-PUSH

The rate-based push is unable to reach an overall request rate higher than $\lambda^{T+1}/(1-\lambda^T)$ for any T. When the overall probe limit R exceeds this value, it is possible to use the remaining request rate by using the max-push variant as introduced in [6]. This strategy lets nodes with a queue length of T send probes at a finite rate $r_{mp|R}$, and migrates all new arrivals to queues with length T by sending probes at an infinite rate until an empty server is found. As $\lambda^{T+1}/(1-\lambda^T)$ is an increasing function in λ and decreasing in T, the unique solution for λ to $\lambda^{T+1}/(1-\lambda^T) = R$ is increasing in T. Therefore, there is a unique $T > 1$ satisfying

$$\lambda^{T+1}/(1-\lambda^T) \leq R < \lambda^T/(1-\lambda^{T-1}). \tag{9}$$

For this T, the evolution of the max-push strategy can be described by a set of ODEs $\frac{d}{dt}x(t) = G(x(t))$, where $x(t) = (x_1(t), x_2(t), \ldots)$ and $x_i(t)$ represents the fraction of the number of nodes with at least i jobs at time t.

Theorems 7, 8 from [6] show that the set of ODEs has a unique fixed point $\dot{\pi} = (\dot{\pi}_1, \ldots, \dot{\pi}_T)$ that is a global attractor, and can be expressed as

$$\dot{\pi}_i = \lambda^i \frac{1 + (\frac{\lambda}{1-\lambda} + r)(1-\lambda^{T-i})}{1 + (\frac{\lambda}{1-\lambda} + r)(1-\lambda^{T-1})}, \tag{10}$$

for $1 \leq i \leq T$. The mean delay D_{mp} of a job under the max-push strategy is given by [6, Corollary 3]:

$$D_{mp} = \frac{1-\lambda^T + (\frac{\lambda}{1-\lambda} + r)(1 - T\lambda^{T-1} + (T-1)\lambda^T)}{1 + r(1-\lambda)(1-\lambda^{T-1}) - \lambda^T}. \tag{11}$$

For the max-push strategy the overall probe rate R equals

$$R = \dot{\pi}_T \left(\frac{\lambda}{1-\lambda} + r\right), \tag{12}$$

as the instantaneous transfer of an arrival to a queue with T jobs requires $1/(1-\lambda)$ probe messages on average. Therefore, a predefined overall probe rate R can be matched by setting

$$r_{mp|R} = \frac{R}{\lambda^{T-1}(R+\lambda) - R} - \frac{\lambda}{1-\lambda}, \tag{13}$$

where $0 \leq r_{mp|R} < \infty$ for $\lambda^{T+1}/(1-\lambda^T) \leq R < \lambda^T/(1-\lambda^{T-1})$.

4.1 Limiting the Overall Migration Rate

As the rate-based push strategy cannot exceed the probe rate $\lambda^{T+1}/(1-\lambda^T)$, it is unable to exceed an overall migration rate of $(1-\lambda)\lambda^{T+1}/(1-\lambda^T)$. It follows that when $M > (1-\lambda)\lambda^{T+1}/(1-\lambda^T)$, queues with a length of at least $T+1$ can probe at an arbitrarily high rate without exceeding the migration limit M, effectively reducing all queues to a length of at most T. Queues with length T can then send probes with a finite r to match M. In other words, in order to match M (instead of R as in the previous section), set T such that

$$\frac{(1-\lambda)\lambda^{T+1}}{1-\lambda^T} \leq M < \frac{(1-\lambda)\lambda^T}{1-\lambda^{T-1}}. \tag{14}$$

and determine the probe rate $r_{mp|M}$ when the queue length equals T by the following theorem:

THEOREM 5. *The max-push strategy matches a predefined migration rate M by letting the probe rate $r = r_{mp|M}$ with*

$$r_{mp|M} = \lambda \left(\frac{M}{((1-\lambda)\lambda + M)\lambda^T - \lambda M} - \frac{1}{1-\lambda}\right). \tag{15}$$

When matching M this way, $\dot{\pi}_i$ for $i \leq T$ reduces to

$$\dot{\pi}_i = \lambda^i - \frac{M(1-\lambda^{i-1})}{1-\lambda}, \tag{16}$$

Proof. The relationship (15) follows from the formulation of the overall migration rate. The migrations of new arrivals in queues with length T are given by $\dot{\pi}_T\lambda$. The migrations resulting from a successful probe sent by queues with length T are given by $\dot{\pi}_T r(1-\lambda)$. Probes are successful if they locate an empty server, which they do with probability $1-\lambda$. Therefore, the overall migration rate can be expressed as

$$M_{mp} = \dot{\pi}_T \left(\frac{\lambda}{1-\lambda} + r\right)(1-\lambda) \tag{17}$$
$$= \frac{(1-\lambda)((1-\lambda)r + 2\lambda)\lambda^{T+1}}{\lambda(r(1-\lambda)+1) - ((1-\lambda)r+\lambda)\lambda^T}.$$

The reduction of $\dot{\pi}_i$ to (16) follows from substitution of (15) in (10), and shows the improvement over an M/M/1 queue directly. □

Theorem 6. *For the max-push strategy r and T should be set as follows in order to respect both the probe rate R and migration rate M:*

- *If $\lambda < 1 - M/R$, T must be chosen according to* (14) *and r can be at most $r_{mp|M}$.*
- *If $\lambda > 1 - M/R$, T must be chosen according to* (9) *and r can be at most $r_{mp|R}$.*
- *If $\lambda = 1 - M/R$, both constraints are equivalent.*

The proof is given in Appendix E of [7].

Theorem 7. *The mean delay of the max-push strategy can be expressed as*

$$D_{mp} = \frac{1}{1-\lambda}\left(1 - \frac{M_{mp}}{\lambda}\left(\frac{\alpha+\beta}{M_{mp}}\right)\right), \tag{18}$$

with $\alpha = \dot{\pi}_T\lambda T$ and $\beta = \dot{\pi}_T r(1-\lambda)(T-1)$. When $r = r_{mp|M}$, the mean delay D_{mp} reduces to

$$D_{mp|M} = \frac{1}{1-\lambda} + \frac{M\left(1-\lambda^T\right)}{(1-\lambda)^2\lambda} - \frac{MT + \lambda^{T+1}}{(1-\lambda)\lambda}. \tag{19}$$

Proof. The improvement in mean delay compared to a standard M/M/1 queue, can be expressed as a migration frequency (M_{mp}/λ) times a migration gain. The migration frequency denotes how many migrations per job take place on average. The migration gain quantifies the number of places in the queue the migrating job skips. The fraction of migrating jobs *arriving* at a queue with length T $(\dot{\pi}_T\lambda/M_{mp})$ skip T places in the queue. The fraction of migrating jobs from queues with length T, being $\pi_T r(1-\lambda)/M_{mp}$, skip $T-1$ places in the queue. Hence, the migration gain is $(\alpha+\beta)/M_{mp}$.

The reduction to $D_{mp|M}$ is found by applying Little's Law to the expression for $\dot{\pi}$ in (16), and shows the improvement over an M/M/1 queue explicitly. □

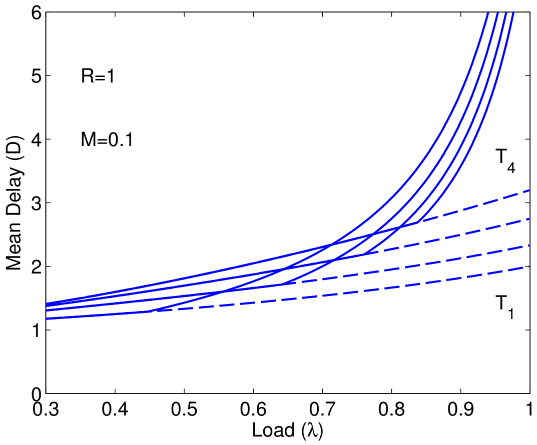

Figure 2: The mean delay of the pull strategy for $T = 1, ..., 4$. The probe rate r is constrained by both R and M (full lines). The delay shown in dashed lines is achieved when there is no migration limit, and only the probe limit is in effect. Choosing $T = 1$ is no longer optimal when a maximum migration rate is imposed.

5. CONDITIONAL PULL

When only considering a maximum allowed probe rate R, the optimal choice for a pull strategy is to let $T = 1$ [6, Theorem 5]. This is no longer the case when taking a maximum allowed migration rate M into account, as shown in Figure 2. Intuitively, when the migration limit is small, it is best to pull jobs from longer queues only, resulting in a lower mean delay.

To reduce the mean delay of the rate-based pull strategy, we introduce the conditional pull strategy that can match both R and M. Empty servers send probes according to an interrupted Poisson process with rate r. Under the conditional pull strategy, empty nodes always accept jobs from queues with length of at least $T+1$ and also accept jobs from a queue with length T with some probability p. This strategy relies on the choice of p to match the migration rate M, and lets the probe rate r be determined by R, i.e. $r = r_{pull|R} = R/(1-\lambda)$. Thus, *both* R and M are matched, this is in contrast with the previous strategies, where r was always chosen as large as possible without exceeding R and M.

First we note that one can easily see that for λ_T, being the unique root defined in Lemma 2, $\lambda_T < \lambda_{T+1}$ (as $M/((\lambda(1-\lambda)+M)\lambda^T - M)$ increases in T and $\frac{R}{1-\lambda}$ increases in λ independent of T). Given λ, the conditional pull strategy sets T such that

$$\lambda_{T-1} \le \lambda < \lambda_T, \tag{20}$$

with $\lambda_0 = 0$. To analyze the response time of a job under the conditional pull strategy we introduce a set of ODEs $\frac{d}{dt}x_i(t) = H(x(t))$, where $x(t) = (x_1(t), x_2(t), \ldots)$ and $x_i(t)$ represents the fraction of the number of nodes with at least i jobs at time t. As explained below, the set of ODEs $H(x(t))$ describing the time evolution of the queue lengths under the conditional pull strategy is defined as

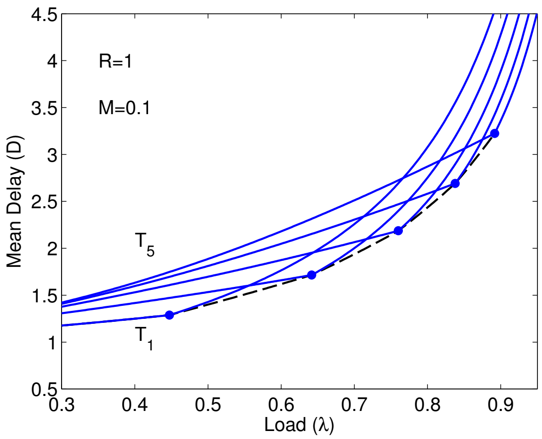

Figure 3: Showing the mean delay of the pull strategy with $T > 1$, respecting a migration limit $\dot{M}$. The conditional pull variant is shown in dashed lines.

$$\frac{dx_1(t)}{dt} = -(x_1(t) - x_2(t)) + (\lambda + rx_{T+1}(t) + rp(x_T(t) - x_{T+1}(t)))(1 - x_1(t)) \quad (21)$$

$$\frac{dx_i(t)}{dt} = \lambda(x_{i-1}(t) - x_i(t)) - (x_i(t) - x_{i+1}(t)), \quad (22)$$

for $1 < i < T$, and

$$\frac{dx_i(t)}{dt} = \lambda(x_{i-1}(t) - x_i(t)) - (1 + rp^{1[i=T]}(1 - x_1(t)))(x_i(t) - x_{i+1}(t)), \quad (23)$$

for $i \geq T$, where $1[A] = 1$ if A is true and $1[A] = 0$ otherwise. The terms $\lambda(x_{i-1}(t) - x_i(t))$ and $x_i(t) - x_{i+1}(t)$, for $i \geq 1$, correspond to arrival and service completions, respectively. Queues of length 1 are created by job transfers at rate $(rx_{T+1}(t) + rp(x_T(t) - x_{T+1}(t)))(1 - x_1(t))$ as the fraction of empty nodes $(1 - x_1(t))$ probe at rate r, and a probe is successful with probability $x_{T+1}(t) + p(x_T(t) - x_{T+1}(t))$. Similarly, migrating jobs reduce the number of queues with exactly i jobs, for $i > T$, at rate $r(1 - x_1(t))(x_i(t) - x_{i+1}(t))$ and at rate $rp(1 - x_1(t))(x_T(t) - x_{T+1}(t))$ for $i = T$.

The next theorem shows that this set of ODEs has a unique fixed point with $\sum_{i\geq 1} \hat{\pi}_i < \infty$. In Appendix F of [7] we briefly argue why this fixed point can be used to approximate the queue length distribution of a node as the number of nodes becomes large. The argument is similar to the one used in [5]. We also validate the accuracy of this approximation by simulation in Section 5.1.

Theorem 8. *The set of ODEs $\frac{d}{dt}x(t) = H(x(t))$ has a unique fixed point $\hat{\pi} = (\hat{\pi}_1, \hat{\pi}_2, \ldots)$ with $\sum_{i\geq 1} \hat{\pi}_i < \infty$. The fixed point can be expressed as:*

$$\hat{\pi}_i = \frac{\lambda^i \left((1-\lambda)r\left(\sum_{j=0}^{T-i} \lambda^j + p(r+1)\left(1-\lambda^{T-i}\right)\right)+1\right)}{(1-\lambda)r\left(\sum_{j=0}^{T-1} \lambda^j + p(r+1)\left(1-\lambda^{T-1}\right)\right)+1} \quad (24)$$

for $1 \leq i \leq T$, and for $i > T$ as

$$\hat{\pi}_i = \pi_T \left(\frac{\lambda}{1 + r(1-\lambda)}\right)^{i-T} \quad (25)$$

Proof. Assume $\hat{\pi}$ is a fixed point with $\sum_{i\geq 1} \hat{\pi}_i < \infty$, meaning $H_i(\hat{\pi}) = 0$ for $i \geq 1$, where $H(x) = (H_1(x), H_2(x), \ldots)$. When $\sum_{i\geq 1} \hat{\pi}_i < \infty$, we can simplify $\sum_{i\geq 1} H_i(\pi) = 0$ to $\lambda - \hat{\pi}_1 = 0$. Hence, $\hat{\pi}_1$ must equal λ. The expressions for $\hat{\pi}_i$ then readily follow from the conditions $H_i(\hat{\pi}) = 0$, for $i \geq 1$. □

Theorem 9. *A predefined overall migration rate M can be matched by setting $p = p_{cp|M}$, with*

$$p_{cp|M} = \frac{M - \bar{\pi}_{T+1} r(1-\lambda)}{(\bar{\pi}_T - \bar{\pi}_{T+1})r(1-\lambda)} \quad (26)$$
$$= \frac{\lambda\left(r\left(-\lambda^2 + \lambda + M\right)\lambda^T - M(r+1)\right)}{(1-\lambda)r(r+1)\left(\lambda M - \left(-\lambda^2 + \lambda + M\right)\lambda^T\right)}.$$

When matching M by setting $p = p_{cp|M}$, $\hat{\pi}_i$ reduces to

$$\hat{\pi}_i = \lambda^i - \frac{M(1-\lambda^{i-1})}{1-\lambda}, \quad (27)$$

for $i \leq T$.

Proof. The fraction of empty queues $(1-\lambda)$ send probes at rate r. Probes are successful with probability 1 if they locate a queue with length at least $T+1$ $(\bar{\pi}_{T+1})$. Probes are successful with probability p if they locate a queue with length equal to T $(\bar{\pi}_T - \bar{\pi}_{T+1})$. In other words:

$$M_{cp} = r(1-\lambda)(\bar{\pi}_{T+1} + p(\bar{\pi}_T - \bar{\pi}_{T+1}), \quad (28)$$

from which (26) follows by algebraic manipulation. The reduction of $\hat{\pi}_i$ to (27) is found by substituting (26) in (24). □

Theorem 10. *The mean delay D_{cp} of a job under the conditional pull strategy equals*

$$D_{cp} = \frac{1}{1-\lambda}\left(1 - \frac{M_{cp}}{\lambda}(\alpha - \beta)\right), \quad (29)$$

with

$$\alpha = T + \frac{\lambda}{(1-\lambda)(1+r)} \quad \text{and} \quad \beta = \frac{r(1-\lambda)p\bar{\pi}_T}{M_{cp}}.$$

Proof. The improvement in mean delay compared to a standard M/M/1 queue, can be expressed as a migration frequency (M_{cp}/λ) times a migration gain $(\alpha - \beta)$. The fraction of migrating jobs where the job is pulled from a queue with length at least $T+1$, $(r(1-\lambda)\bar{\pi}_{T+1}/M_{cp})$ skip α places in the queue: The same remarks as in Theorem 2 apply. The other jobs $((r(1-\lambda)p(\bar{\pi}_T - \bar{\pi}_{T+1}))/M_{cp})$ are pulled from a queue with length equal to T, thus skipping exactly $T-1$ places. In other words, the migration gain can be expressed as:

$$\frac{\alpha r(1-\lambda)\bar{\pi}_{T+1} + (T-1)(r(1-\lambda)p(\bar{\pi}_T - \bar{\pi}_{T+1}))}{M_{cp}},$$

which can be rewritten as $\alpha - \beta$. □

Figure 3 shows the mean delay of the conditional pull strategy. The dots represent λ_T, i.e., the intersection points of $r_{pull|R}$ and $r_{both|M}$. The conditional pull strategy achieves a lower mean delay compared to the rate-based pull strategies with $T > 1$, as it transfers more jobs.

THEOREM 11. *When $R = +\infty$ and M is finite, the max-push and conditional pull strategies have the same stationary queue length distribution.*

PROOF. When there is no probe limit R, the parameter r is allowed to be arbitrarily large for the conditional pull strategy. In this case the maximum queue length will be T as $\lim_{r\to\infty} \hat{\pi}_i = 0$ for $i > T$, see (25). We therefore know from Theorems 5 and 9 that both the max-push and conditional pull strategy have the same queue length distribution when only matching M, if they use the same T. What remains to be shown is that both strategies make use of the same T.

Recall that λ_T was defined as the solution in $(0, 1)$ of $r_{pull|R} - r_{both|M}$, that is,

$$\frac{R}{1-\lambda} = \frac{M}{(\lambda(1-\lambda)+M)\lambda^T - M}.$$

The left hand side tends to infinity as R tends to infinity. Hence, $r_{both|M}$ must tend to infinity, meaning λ_T is the solution to $M = (1-\lambda)\lambda^{T+1}/(1-\lambda^T)$. The λ_T's are thus exactly the M values where the max-push strategy changes its T value, see (14). Hence, both the conditional pull and max-push strategy choose the same T. □

5.1 Model Validation

We validate the infinite system model for the conditional pull strategy by comparing the closed form results of Theorem 10 with time consuming simulation results for systems with a finite number of nodes N. The infinite and finite system model only differ in the system size. Hence, the rate r and probability p in the simulation experiments is independent of N and was determined by using the expression for p from Equation (26) and $r = R/(1-\lambda)$. Each simulated point in the figures represents the average value of 25 simulation runs. Each run has a length of 10^6 time units (where the service time is exponentially distributed with a mean of 1 time unit) and a warm-up period of length $10^6/3$ time units.

		Load (λ)				
		0.5	0.65	0.7	0.75	0.8
System Size (N)	25	1.1e-2	1.7e-2	1.9e-2	2.4e-2	2.9e-2
	50	5.6e-3	8.2e-3	9.5e-3	5.7e-3	7.1e-3
	100	2.8e-3	3.9e-3	4.6e-3	5.7e-3	7.1e-3
	200	1.3e-3	2.0e-3	2.3e-3	2.7e-3	3.4e-3
	400	7.0e-4	1.0e-3	1.2e-3	1.3e-3	1.7e-3
	800	3.5e-4	5.0e-4	5.6e-4	6.6e-4	8.1e-4
	1600	1.6e-4	2.5e-4	2.6e-4	3.8e-4	4.3e-4

Table 1: Relative error of mean delay, given by (29), for the conditional pull strategy with $R = 1$ and $M = 0.1$ when compared to simulation results.

Table 1 compares the mean delay in a finite system with N nodes with the mean delay in the infinite system model under the conditional pull strategy with $R = 1$ and $M = 0.1$ for $N = 25, 50, \ldots, 1600$ and $\lambda = 0.5, 0.65, 0.7, 0.75$ and 0.8. For each combination of N and λ we also show the relative error. The error clearly decreases as N grows, and is worse for larger λ values.

The observed overall migration rate in the simulation is strictly lower than the predefined M, meaning less jobs will be transferred than anticipated. Hence, the mean delay in the simulation experiments is pessimistic. This error is in part due to the choice of p, which was determined using (26). This choice relies on the infinite system model whereas we are now studying a finite system. The relative error in the observed overall migration rate is nearly load-insensitive and decreases linearly as the system doubles in size, as shown in Table 2.

N	25	50	100	200	400	800	1600
Rel. Err.	4%	2%	1%	.5%	0.25%	.13%	.064%

Table 2: Relative error of the observed overall migration rate for finite system size when compared to the targeted migration rate M.

6. PUSH VERSUS PULL STRATEGIES

We compare the performance of the max-push and the conditional pull strategies with a predefined overall probe limit R and migration limit M (using Theorems 7 and 10). The parameter T is determined by the load λ, as each strategy is only defined for a specific T given any λ (see (9), (14) and (20)). For the max-push, the value for T and r is chosen to match the strictest constraint of either R or M depending on the load (see Theorem 6). For the conditional pull all idle servers probe with rate $r = R/(1-\lambda)$, and p is chosen to match M (see Theorem 9).

The mean delay of the max-push and conditional pull strategy with $M = 0.1$ and $R = 0.4$ is shown in Figure 4. The max-push strategy is limited by the probe limit when $\lambda > 1 - M/R$ and by the migration limit when $\sqrt{M} < \lambda < 1 - M/R$. The mean delay of the push strategy is one in case $\lambda < \sqrt{M}$, as all newly arriving jobs at a busy server can be migrated instantaneously to an empty server without violating the R and M constraints. For the conditional pull strategy the limiting factor is R when $\lambda < \sqrt{M + M/R}$, and M for $\lambda > \sqrt{M + M/R}$.

The intervals where both strategies are constrained by M do not always overlap, i.e. $\sqrt{M + M/R}$ can be larger than $1 - M/R$, as is the case for $R = 1$ and $M = 0.3$. When $\sqrt{M + M/R} < 1 - M/R$ both strategies transfer the same number of jobs when $\sqrt{M + M/R} < \lambda < 1 - M/R$. However, the max-push will outperform the conditional pull as the average migration gain is larger. This is not unexpected as the max-push strategy avoids that queues become larger than T, whereas queues with a length exceeding T exist for the conditional pull as it only sends random probes at a finite rate.

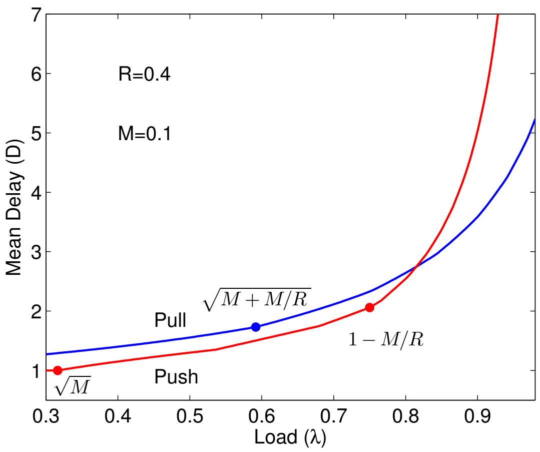

Figure 4: Mean delay of the max-push and conditional pull strategies, with $R = 0.4$ and $M = 0.1$.

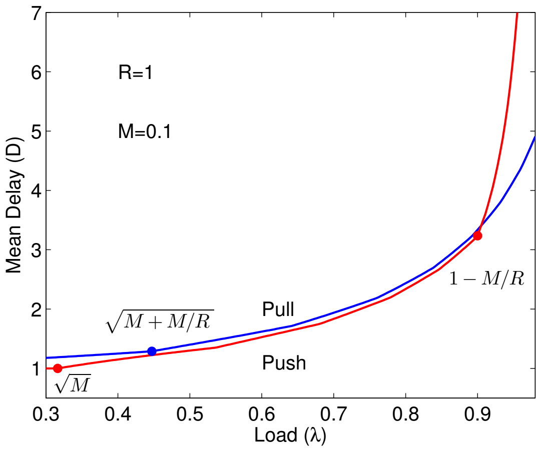

Figure 5: Mean delay of the max-push and conditional pull strategies, with $R = 1$ and $M = 0.1$.

As expected from Theorem 11, the difference in performance between max-push and conditional-pull becomes smaller when increasing R, as shown in Figure 5. As the empty queues send probes with rate $r = R/(1 - \lambda)$, they are allowed to send more probes as R increases. This increases the odds that a long queue is probed, thus lowering the mean delay. This can also be observed by looking at the values for T. By increasing R, a larger value for T can be used for the same load. This requires that jobs are pulled from longer queues, increasing the migration gain per transfer.

In conclusion, whenever the maximum allowed probe rate R clearly exceeds the maximum allowed migration rate M (which is the case that is mainly of practical interest), the pull strategy is either clearly superior (for large λ) or has a similar performance to the max-push strategy (for medium to low λ).

7. REFERENCES

[1] D. Eager, E. Lazowska, and J. Zahorjan. Adaptive load sharing in homogeneous distributed systems. *Software Engineering, IEEE Transactions on*, SE-12(5):662 –675, may 1986.

[2] D. Eager, E. Lazowska, and J. Zahorjan. A comparison of receiver-initiated and sender-initiated adaptive load sharing. *Perform. Eval.*, 6(1):53–68, 1986.

[3] N. Gast and B. Gaujal. A mean field model of work stealing in large-scale systems. *SIGMETRICS Perform. Eval. Rev.*, 38(1):13–24, 2010.

[4] 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. *Perform. Eval.*, 68:1056–1071, 2011.

[5] W. Minnebo and B. Van Houdt. A fair comparison of pull and push strategies in large distributed networks. *IEEE/ACM Transactions on Networking*, 2013.

[6] W. Minnebo and B. Van Houdt. Improved rate-based pull and push strategies in large distributed networks. In *Proc. of the IEEE 21-th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, San Francisco (USA), 2013.

[7] W. Minnebo and B. Van Houdt. Tech. report: Analysis of rate-based pull and push strategies with limited migration rates in large distributed networks. `http://win.ua.ac.be/~vanhoudt/papers/reports/MigrationPaper.pdf`, 2015.

[8] R. Mirchandaney, D. Towsley, and J. Stankovic. Analysis of the effects of delays on load sharing. *IEEE Trans. Comput.*, 38(11):1513–1525, 1989.

[9] R. Mirchandaney, D. Towsley, and J. A. Stankovic. Adaptive load sharing in heterogeneous distributed systems. *J. Parallel Distrib. Comput.*, 9(4):331–346, 1990.

[10] M. Mitzenmacher. The power of two choices in randomized load balancing. *IEEE Trans. Parallel Distrib. Syst.*, 12:1094–1104, October 2001.

[11] I. V. Spilbeeck and B. V. Houdt. Performance of rate-based pull and push strategies in heterogeneous networks. *Performance Evaluation*, 91:2 – 15, 2015. Special Issue: Performance 2015.

[12] A. Stolyar. Pull-based load distribution in large-scale heterogeneous service systems. *Queueing Systems*, 80(4):341–361, 2015.

[13] B. Van Houdt. Performance comparison of aggressive push and traditional pull strategies in large distributed systems. In *Proceedings of QEST 2011, Aachen (Germany), IEEE Computer Society*, pages 265–274, SEP 2011.

[14] N. Vvedenskaya, R. Dobrushin, and F. Karpelevich. Queueing system with selection of the shortest of two queues: an asymptotic approach. *Problemy Peredachi Informatsii*, 32:15–27, 1996.

[15] L. Ying, R. Srikant, and X. Kang. The power of slightly more than one sample in randomized load balancing. In *Proc. of IEEE INFOCOM*, 2015.