Analysis of the Task Assignment based on Guessing Size (TAGS) policy

Hagit Sarfati and Eitan Bachmat

ABSTRACT

We consider the performance of Task Assignment based on Guessing Size (TAGS), a multi-host job assignment policy. This policy, which was introduced by M. Harchol-Balter, was designed to work in a non-preemptive setting with unknown job sizes. The policy which is not work preserving. It has been studied numerically, especially in the important case of Bounded Pareto job size distributions. In this paper we provide the first mathematical analysis of TAGS. We compute exact stability conditions for the policy. The stability condition is given in terms of the accuracy of Markov's inequality as an estimator for the first moment of the job size distribution. A consequence of our analysis is that for any bounded job size distribution, TAGS cannot handle a load which is more than logarithmic in the ratio of longest to shortest job. In the case of Bounded Pareto job size distributions we show that certain approximation equations for average waiting time are conservative and relatively accurate. We then compute near optimal cutoffs in this setting. We show that at low loads the penalty for not knowing job sizes is bounded by a factor of 2. We introduce a variant of TAGS which we call T+F which is much more stable than the original version. When using T+F, the penalty for not knowing job sizes is reasonable for all but the highest loads. Finally, we compare the performance of TAGS and FIFO using actual simulations rather than approximation formulas and explain when each one is preferable.

Keywords: Queueing theory, Multiple host task assignment, Non-preemptive policies, Heavy-tailed distributions.

1. INTRODUCTION

Many installations such as web server farms and computing centers have a multitude of hosts which can serve any incoming request. There has been a growing body of research regarding scheduling policies for such multi-host systems. In this paper we will examine multi-server systems that do not allow preemption and individual job sizes are not known. This situation occurs in super computing centers, [8]. One of the classical assignment methods for multi-server systems is the `FIFO` policy. In this policy all incoming jobs are held in a single queue and are released, in the order of arrival, whenever a server becomes available. It is clear from the description that the policy does not require knowledge of job sizes. Recent empirical data has suggested that many workloads which are typical of networked systems are *heavy-tailed*, [2, 4, 14]. By heavy-tailed we will mean distributions for which the probability of a job of size s or more, decreases polynomially rather than exponentially fast with s, and whose variance is large. When the job size distribution is heavy-tailed there is a substantial number of very long jobs and it may happen that short jobs will have to wait behind the long jobs until they complete, leading to large waiting times.

These considerations led to the design of some variance reducing assignment policies. One of the basic examples of a variance reducing policy is the express line in the supermarket. In the express line we assign a server to handle only small jobs up to a certain size, say up to 10 items. In some supermarkets, there are two express lines, for example, one handling up to 10 items and the other handling 11-20 items. Taken to the extreme we can imagine all the lanes in the supermarket being assigned to handle customers with a certain range of items.

Precisely this idea was introduced in the context of computer servers in [11] as a method of reducing variance in multi-server systems which experience heavy-tailed job size distributions. The resulting family of policies is called Size Interval Task Assignment (`SITA`) policies. The policies in this family are parametrized by the intervals (ranges) of job sizes that each server services. The endpoints of the ranges are called *cutoffs* since they separate the jobs into different servers. The optimal choice of cutoffs depends on the job size distribution, the load and the target function that we wish to minimize.

A `SITA` policy requires knowledge of job sizes. A variant that works in the case of unknown job sizes was introduced and studied numerically in [8]. The variant which was named Task Assignment based on Guessing Size (`TAGS`), operates as follows. All jobs are sent to the first server which processes them up to some given time s_1. If they do not complete by then, they are stopped and are sent to the next server to start from scratch. The second server will work on the job up to some time $s_2 > s_1$, if a job does not complete by then, it is sent to the next server and so on. If there are h servers, the policy is again parametrized by the $h-1$ cutoffs $s_1 < s_2 < \ldots < s_{h-1}$.

While the SITA and TAGS policies seem similar there are some important differences. For example, TAGS unlike SITA is not work preserving, it kills jobs and restarts them from scratch. There was no study of the load handling capabilities of TAGS, when is it stable? and similar questions. Considering low loads, where stability is not an issue, how much performance do we lose simply by not knowing job sizes?

The benchmark job size distribution in many of the studies of TAGS or SITA has either been empirical or the family of Bounded Pareto distributions with parameter $0 < \alpha < 2$. These distributions are heavy-tailed and in their unbounded version have infinite variance. The case of interest is when the ratio of the largest job to smallest job is very large, say six orders of magnitude, leading to very large variance. As noted above, these distributions have been shown in several cases to model web related job sizes.

The SITA studies, [12] and in greater detail [24, 13, 1] have shown that with such job size distributions, the comparison between SITA and FIFO depends on the value of the parameter α. roughly, FIFO having the upper hand for values $\alpha > 1$ and SITA when $\alpha < 1$. We note that these analytical comparisons are asymptotic, when the size of the largest job tends to infinity, no actual comparison has been made for reasobable valuess of the largest job.

The TAGS policy was compared to FIFO numerically in [8], however, these comparisons did not involve simulations of either policy, but rather numerical evaluations of approximate formulas for the waiting time. The approximation formula which was used to assess the performance of FIFO was unfortunately very inaccurate.

In the evaluation of the TAGS policy the approximate waiting time formulas assumed an exponential inter-arrival distribution at all servers. The formulas are not exact, since all but the first server experience non exponential inter-arrival times. While the approximation seems very reasonable it was never verified that it actually produces results which are close to the real values.

The present paper resolves many of the question marks regarding the performance and stability of TAGS. Our analysis of stability conditions holds for all job size distributions and is rather complete. We also provide the first analytical examination of TAGS systems with bounded Pareto job size distributions and an accurate comparison to FIFO in this setting, showing rather precisely when each of the scheduling algorithms prevails. We summarize our results.

In section 2, we review some background material.

In section 3 we study stability issues. The results do not assume Poisson arrivals. We provide a very simple algorithm to decide if there exists a stable TAGS policy on h hosts, given a particular job size distribution X and load ρ. We provide a general formula in terms of X, for the largest (critical) load ρ_{crit}, for which there exists a stable TAGS policy on any number of hosts. In other words, if the system experiences a load above ρ_{crit}, then, regardless of the number of hosts, there will not exist cutoffs that will make the TAGS policy stable. If the system experiences a load below ρ_{crit} then with sufficiently many hosts, we can find cutoffs that will stabilize the TAGS policy. We compute the critical load explicitly for several families of distributions, including the Bounded Pareto distributions. The critical load ρ_{crit} has an interesting probabilistic interpretation. Markov's inequality states that for any s, $E(X) \geq s(1 - X(s))$. We can think of the inequality as providing an estimate for $E(X)$ and we consider the value $\tilde{s}$ which provides the smallest error, i.e., $\tilde{s}(1-X(\tilde{s}))$ is maximal. Measuring the error multiplicatively, we consider $E(X)/\tilde{s}(1 - X(\tilde{s}))$. This is precisely ρ_{crit}. We also show that for any distribution X such that the ratio between largest and smallest job is at most p, $\rho_{crit} \leq \ln(p)+1$, which means that the Markov inequality (for the right value of s) is surprisingly accurate. When X is Bounded Pareto with parameter $\alpha = 1$, this bound is nearly attained. Overall, the results show that TAGS does not handle load well.

From section 4 onward we assume Poisson arrivals and an i.i.d Bounded Pareto job size distribution with a large ratio between the largest to smallest job.

In section 4 we validate via simulations the accuracy of the approximate formulas for average waiting time and show that they are conservative, namely, they overestimate waiting times.

In section 5, we analyze the approximation formulas assuming that they are conservative, as we have verified empirically in section 4. Under this assumption we give explicit formulas for near optimal cutoffs for the TAGS policy and compute the resulting performance. We show that at low loads, the penalty for not knowing the job size, i.e., the ratio in average waiting time between SITA and TAGS is at most 2. This bound holds regardless of the number of hosts or the parameter α. This result shows that the main problem with TAGS is load handling.

Given that TAGS does not handle load well, there is a fast growing performance gap between SITA and TAGS as the load increases. To overcome this difficulty, we introduce in section 5 a new policy, which does not use job size, which we call T+F. The new policy combines TAGS and FIFO to produce a much stabler version of TAGS, whose performance is much closer to that of SITA at all loads.

In section 6, we provide a simulation based comparison between the performance of TAGS and FIFO. The simulation results fit well the analysis of TAGS. In fact, the major stumbling block for a complete analysis of the simulation results is our relatively poor understanding of the performance of FIFO.

In section 7, we conclude the paper and discuss possible future work and open problems.

1.1 Related work

The TAGS policy was first introduced and studied in [8]. The performance was compared to that of FIFO on a workload with Poisson arrivals and Bounded Pareto job size distribution. The policies were not simulated. Instead, approximation equations were used to estimate the performance for both TAGS and FIFO. The systems studied consisted mainly of 2 hosts, sometimes 4. The chosen target function was mostly average slowdown, but in some cases average waiting time was considered as well.

In [22] the authors suggest a generalization of TAGS in which jobs are initially sent to different hosts with given probabilities and then proceed as in TAGS. In [3] the same generalization is considered under the additional favorable assumption that jobs need not restart when passing from one host to another, but rather resume service. This new assumption completely changes the stability properties of the algorithm. This is very different from the original TAGS setting where non-preemptivness is an important assumption. In fact, the algorithm is constructed with this limitation in mind. This setting was also explored in [6] in a heavy traffic

regime.

The problem of finding good cutoffs for TAGS was explored numerically in [23], which suggested the use of formal algebra software.

There have been numerous studies of SITA, both numerical [11], [9], [21, 15, 26, 27, 28], and analytical [7, 12, 13, 24, 1]. However, not all analytical techniques and results carry over to the study of TAGS and the numerics can be very different. In addition, some issues such as stability are unique to TAGS.In fact, mapping the similarities and differences between SITA and TAGS is one of the main goals of the present paper.

The performance characteristics of the FIFO policy which is compared to TAGS in section 6 have been analyzed in a series of papers [17, 16, 18, 19, 20, 25].

2. PRELIMINARIES

2.1 The TAGS assignment policy

In this paper we consider multi-host assignment policies with no preemption, in the case where job sizes are not known. We will assume though, that the job size distribution is known or can be deduced by collecting statistics. The TAGS policy can be described as follows:
There are hosts numbered $1, \ldots, h$. In addition, there are cutoff values $s_1 < s_2 < \ldots < s_{h-1}$. All incoming jobs are sent to host 1. At each host, jobs are serviced in first come, first served (FCFS) order. If a job is being processed by the i'th server and completes before s_i time units it leaves the system. If not, it is stopped and put at the end of the queue of the next host, where it starts from scratch. Jobs at the last host always run to completion.

The TAGS policy can be viewed as an extension to the case of unknown job sizes of a policy called SITA which requires the knowledge of job size. In SITA we also use cutoffs $s_1 < \ldots, s_{h-1}$. We also set $s_0 = 0$ and $s_h = \infty$. Since we know job sizes, an incoming job of size s is assigned upon arrival to the host i for which $s_{i-1} \leq s < s_i$. In the setting of unknown job sizes we do not know which host will satisfy the cutoff relations so we try them one by one.

Later on we will compare TAGS to another non preemptive policy, which does not require knowledge of job sizes, the FIFO policy, where each job is sent to the next available host.

In order to consider the TAGS scheduling policy in a queueing theoretic setting, we consider some further system parameters and conditions. A TAGS system consists of the following data and assumptions

- h - The number of hosts. Unless stated otherwise, we assume that all hosts are identical.
- X_i - A sequence of i.i.d job size random variables. We denote by $X(s) = Prob(X_i < s)$ the common job size probability distribution. The job size is with respect to the computing power of a single host.
- ρ - The incoming load. We assume that the inter-arrival times are given by an i.i.d sequence of random variables with common distribution Y. We let $\lambda = 1/E(Y)$ be the arrival rate and then $\rho = \lambda E(X)$. Since job size is with respect to a single host, a single host can handle a load $\rho < 1$.
- A set of cutoffs, $1 < s_1 < s_2 < \ldots < s_{h-1} < p$ for the TAGS policy.

2.2 Pareto and Bounded Pareto distributions

Let $\alpha > 0$. A distribution X is said to be Pareto if its density f has the form

$$f(s) = cs^{-\alpha-1} \tag{1}$$

in the range $s \geq 1$. A distribution X is said to be Bounded Pareto if its density has the above form, but is restricted to a bounded domain $1 \leq s \leq p$. The Pareto distribution corresponds in this setting to the choice $p = \infty$. The constant, $c > 0$, is a normalizing constant which ensures that $\int_1^p f(s)\, ds = 1$. A simple computation shows that the normalizing constant is

$$c = \frac{\alpha}{1 - (\frac{1}{p})^\alpha} \tag{2}$$

We denote the Bounded Pareto distribution by $B_p(\alpha)$ and the corresponding unbounded Pareto distribution by $B(\alpha)$. Thinking of a bounded distribution as describing job size, we consider the ratio of the largest to smallest jobs. We will call this ratio, the *range* of the job size distribution. Formally, we let a denote the infimum of the set of values s for which $X(s) > 0$ and by b the supremum of the set of values for which $X(s) < 1$. The interval $[a, b]$ is the minimal interval which supports the distribution and the ratio b/a is the range. The range of $B_p(\alpha)$ is obviously p.

More generally we could have considered bounded distributions whose minimal value is different from 1. However, by changing the time unit of measurement, we can always normalize a job size distribution, bounded away from 0, to have a smallest job of size 1. Changing the time unit preserves the range. Of particular interest to us will be the case where $0 < \alpha < 2$, for which the Pareto distribution has infinite variance. We think of the corresponding Bounded Pareto distributions with large range as providing a natural model for high variance job size distributions.

2.3 The objective function

Let X denote the job size random variable and let W denote the difference between the time spent in the system and the job size. We call W the *waiting time*. The objective function that we consider is a normalized version of average waiting time

$$E(N) = E(W)/E(X)$$

where $E(W)$ is the average waiting time and $E(X)$ is the average job size. The quantity $E(N)$ is preferable to $E(W)$ since it is invariant under time unit changes.

3. BOUNDS ON THE LOAD OF A TAGS SYSTEM

We consider a TAGS system with h identical hosts. let $X(s)$ denote the job size distribution function associated with the job size random variable X, namely $X(s)$ is the probability that a job takes less than s time units to complete. For simplicity we will assume as before that the smallest job has size 1 and the largest has size p. We let λ be the rate of job requests to the TAGS system and $\rho = \lambda E(X)$ be the total load on the system. We note that TAGS can be described as a multi-class forward feeding network. The classes correspond

to the jobs which terminate service at host i. The network is forward feeding in the sense that jobs never reenter the same server. For such systems it is known, [5], that stability is equivalent to the condition that each server in the network is sub-critical. The following elementary lemma provides a tight bound on the ability of a TAGS system with any number of hosts to stably support a load ρ.

THEOREM 3.1. *Let $X(s)$ be any bounded, job size distribution function with job sizes in the range $[1, p]$. Let*

$$M(X) = \sup_s s(1 - X(s))$$

where sup *denotes the supremum. Given a load ρ, there exists a number h and cutoffs $s_1, ..., s_{h-1}$ such that the* TAGS *system with h hosts and cutoffs s_i can stably support a workload with job size distribution X if and only if*

$$\rho < E(X)/M(X)$$

Proof: We begin with the second statement. Assume $\rho > E(X)/M(X)$ and choose an s such that

$$\rho > E(X)/(s(1 - X(s)))$$

Let $s_{i-1} \leq s \leq s_i$ be the range of job sizes whose service completes at host i. All jobs of size at least s will pass through the i'th host and the host will spend at least s time units on each such job. The rate of jobs of size at least s is $\lambda(1 - X(s))$, therefore the system must satisfy

$$\lambda s(1 - X(s)) \leq 1$$

in order to be stable. By definition $\rho = \lambda E(X)$ so we get

$$\rho s(1 - X(s))/E(X) < 1$$

a contradiction.

Conversely, assume that $\rho < E(X)/M(X)$ or equivalently $\rho M(X)/E(X) < 1$. All jobs of size at least s_{i-1} pass through host i. We know that s_i is an upper bound on the time spent by host i on any such job. Therefore, the utilization on the host i is bounded from above by

$$\lambda(1 - X(s_{i-1}))s_i \leq \lambda M(X)s_i/s_{i-1}$$

$$= \rho M(X)/E(X)(s_i/s_{i-1}).$$

Fix any $r > 1$ such that $r\rho M(X)/E(X) < 1$ and let $s_i = r^i$, then by the above argument we have that host i will have a utilization below 1. We see that $h = [\log_r(p)] + 1$ hosts will suffice. *q.e.d*

The theorem leads us to define the *critical load* of a distribution $X(s)$,

$$\rho_{crit}(X) = E(X)/M(X)$$

The theorem shows that for a given job size distribution TAGS systems can only support loads which are below the critical load, regardless of the number of hosts available to the system.

Given $\rho < E(X)/M(X)$ we can determine the exact number of hosts needed for constructing a stable TAGS system with load ρ by the following procedure. The load ρ_i on the i'th server of the TAGS system is given by

$$\rho_i = \lambda p_i E(X_i) + \lambda s_i(1 - X(s_i))$$

where $E(X_i)$ is the average size of jobs between s_{i-1} and s_i, and $p_i = X(s_i) - X(s_{i-1})$ is the probability that a job has size in the range $[s_{i-1}, s_i]$. Fixing s_{i-1} it is easy to see that ρ_i is a non decreasing function in s_i. Let $\tilde{s}_1$ be such that $\rho_1 = 1$. More generally, having computed $\tilde{s}_{i-1}$ we iteratively determine $\tilde{s}_i$ to be such that $\rho_i = 1$. If no s exists for which $\rho_i = 1$ then i hosts suffice for establishing a stable TAGS system. If a TAGS system has h hosts and i hosts suffice to for a stable system we define the *number of spare servers* $\tilde{h}$ by the formula $\tilde{h} = h - i - 1$. This definition will be important in later sections.

3.1 Some examples of critical loads

We compute the critical load of Bounded Pareto distributions and of restrictions of Weibull distributions.

3.1.1 Bounded Pareto distributions

THEOREM 3.2. *Consider the Bounded Pareto job size distribution $B = B_p(\alpha)$, where α is any real. Let $a = 1/p$ be the reciprocal to the range of the distribution. Let $\rho_{crit} = \rho_{crit}(\alpha, p)$ be the critical load of $B_p(\alpha)$.*
1) If $a \leq (1 - \alpha)^{1/\alpha}$ then,

$$\rho_{crit} = (1 - a^{1-\alpha})(1 - \alpha)^{-1/\alpha}$$

2) If $\alpha < 1$ and $a^\alpha \geq 1 - \alpha$ then

$$\rho_{crit} = \frac{\alpha}{\alpha - 1}\left(\frac{1 - a^{\alpha-1}}{1 - a^\alpha}\right)$$

3) If $\alpha = 1$ then $\rho_{crit} = \frac{p}{p-1}\ln(p)$

Proof: From the formulas for the $B = B_p(\alpha)$ distribution we have that

$$s(1 - B(s)) = \frac{1}{1 - a^\alpha}(s^{1-\alpha} - p^{-\alpha}s)$$

Differentiating and setting to 0 we see that the maximum of $s(1 - B(s))$ is obtained when

$$s = p(1 - \alpha)^{1/\alpha}$$

assuming that $s \geq 1$.

If $s < 1$, or equivalently when $a \geq (1 - \alpha)^{1/\alpha}$, then $s(1 - B(s))$ is a decreasing function in the range $1 \leq s \leq p$ and hence maximal at $s = 1$. Since $B(1) = 0$ we have $M(B) = 1$ and therefore $E(B)/M(B) = E(B)$ We recall that

$$E(B) = \frac{\alpha(1 - p^{1-\alpha})}{(\alpha - 1)(1 - a^\alpha)}$$

for $\alpha \neq 1$. Therefore,

$$E(B) = \frac{\alpha}{\alpha - 1}\left(\frac{1 - a^{\alpha-1}}{1 - a^\alpha}\right)$$

When $\alpha = 1$ we have

$$E(B) = \frac{1}{1 - \frac{1}{p}}(\ln(p))$$

as required.

Consider now the case $s \geq 1$. Plugging s into $s(1 - B(s))$ we obtain

$$M(B) = \frac{1}{1 - a^\alpha}(p^{1-\alpha}(1 - \alpha)^{1/\alpha - 1} - p^{1-\alpha}(1 - \alpha)^{1/\alpha})$$

$$= \frac{1}{1 - a^\alpha}p^{1-\alpha}(1 - \alpha)^{1/\alpha}\frac{\alpha}{1 - \alpha}$$

Taking the ratio we obtain

$$\frac{E(B)}{M(B)} = (1-a^{1-\alpha})(1-\alpha)^{-1/\alpha}$$

as required. *q.e.d*

We have the following corollary regarding the behavior of the critical load for fixed α as p tends to infinity.

COROLLARY 3.3. *When $p \longrightarrow \infty$ we have the following*
1) If $\alpha < 1$ then $\rho_{crit}(\alpha, p) \rightarrow (1-\alpha)^{-1/\alpha}$ and convergence is from below.
2) If $\alpha = 1$ then $\rho_{crit}(1, p) \geq \ln(p)$ and $\rho_{crit}(1, p)/\ln(p) \rightarrow 1$.
3) If $\alpha > 1$ then $\rho_{crit}(\alpha, p) \rightarrow \frac{\alpha}{\alpha-1}$ and convergence is from below. q.e.d

3.1.2 Restrictions of Weibull distributions

The Weibull distribution with parameter k has a distribution function of the form

$$W_k(s) = 1 - e^{-s^k}$$

for $s \geq 0$. We will compute the critical load for this unbounded distribution. we can think of the result as the limit of the critical load for the distributions which are obtained by restricting the Weibull distribution to an interval $[\varepsilon, p]$ and letting $\varepsilon \rightarrow 0$ and $p \rightarrow \infty$. It is known that $E(W_k) = \Gamma(1+\frac{1}{k})$ where $\Gamma(k) = \int_0^\infty t^{k-1}e^{-t}dt$ is the Gamma function. This is easily seen by using the change of variable $t = s^k$. To compute the maximum of $s(1-W_k(s))$ we differentiate $(s(1-W_k(s))' = e^{-s^k}(1-ks^{k-1})$. Setting to zero we get $s = (\frac{1}{k})^{1/k}$ and for that value of s we get $s(1-W_k(s)) = (\frac{1}{ke})^{1/k}$, where e is the natural base. Letting k tend infinity, we see that $E(W_k)$ tends to $\Gamma(1) = 1$, hence the critical load tends to 1. On the other hand, if k tends to zero, say $k = 1/n$ with n an integer, we get $E(W_{1/n}) = \Gamma(n+1) = n!$. The maximal value of $s(1-W_k(s))$ is $(\frac{n}{e})^n$. According to Stirling's formula, as n tends to infinity, $\rho_{crit}(W_{1/n}) \sim \sqrt{2\pi n}$. For the exponential distribution which is given by setting $k = 1$ we get that the critical load is e.

3.2 A universal bound on the load

The following theorem provides a general bound on the load of any `TAGS` system when the job size distribution has range p.

THEOREM 3.4. *For any distribution X with range p we have $\rho_{crit}(X) \leq \ln(p) + 1$.*

Proof: Let $X(s)$ be a bounded distribution of range p. We claim that for any $\varepsilon > 0$, $X(s)$ may be approximated by a continuous distribution $Y(s)$ of the same range such that $|\rho_{crit}(X) - \rho_{crit}(Y)| < \varepsilon$. To see this, decompose the range interval $[1, p]$ into n equal sub-intervals, with endpoints $1 = x_0, x_1, ..., x_n = p$. Consider the distribution Y_n which linearly extrapolates X between its endpoint values on each sub-interval $[x_i, x_{i+1}]$. Since X and Y_n are both monotone non decreasing functions they are Riemann integrable and it is obvious from the definition that $\lim_n\ E(Y_n) = E(X)$. We claim that $\lim_n\ M(Y_n) = M(X)$ as well. Indeed, For any $s \in [x_i, x_{i+1}]$ we have

$$s(1-X(s)) \leq s(1-X(x_i))$$

$$= (s-s_i)(1-X(x_i)) + x_i(1-X(x_i))$$

$$\leq \frac{p}{n} + x_i(1-Y_n(s_i)) \leq \frac{p}{n} + M(Y_n)$$

hence $M(X) \leq \frac{p}{n} + M(Y_n)$. Applying the same argument to Y_n instead of X and noting that $x_i(1-Y_n(x_i)) = x_i(1-X(x_i)) \leq M(X)$ we see that $M(Y_n) \leq \frac{p}{n} + M(X)$ and both inequalities together yield the claim for n large enough.

Given the above approximation it is enough to prove the bound for continuous distributions. Let X be continuous. We claim that we can find a distribution Y such that:

1) $\rho_{crit}(X) \leq \rho_{crit}(Y)$
2) Y is also supported on $[1, p]$.
3) $M(Y) = 1$.

We always have $M(X) \geq 1(1-X(1)) = 1$. If $M(X) = 1$ there is nothing to prove, hence we assume that $M(X) > 1$. Let $\tilde{s} > 1$ be such that $\tilde{s}(1-X(\tilde{s})) = M(X)$, $\tilde{s}$ exists by continuity of X. Consider the distribution $\tilde{Y}$ which is supported on the interval $[M(X), p]$ which is defined as follows, for $s \geq \tilde{s}$,

$$\tilde{Y}(s) = X(s)$$

and for $M(X) \leq s \leq \tilde{s}$,

$$1 - \tilde{Y}(s) = M(X)/s$$

By construction $M(X) = M(\tilde{Y})$. Also by the definition of $M(X)$ and the construction of $\tilde{Y}$ we have for all s, $1-\tilde{Y}(s) \geq 1-X(s)$. Consequently

$$E(X) = \int_{s=0}^{p} 1-X(s)\ ds \leq \int_{s=0}^{p} 1-\tilde{Y}(s)\ ds = E(\tilde{Y})$$

We conclude that $\rho_{crit}(X) \leq \rho_{crit}(\tilde{Y})$. Finally we define Y to be a rescaling of $\tilde{Y}$ by the formula $Y(s) = \tilde{Y}(M(X)s)$. Rescaling does not change ρ_{crit}, the support of Y is in $[1, p/M(X)]$, which is contained in $[1, p]$, and $M(Y) = 1$.

Following the above argument it is sufficient to prove the bound for continuous distributions which satisfy $M(X) = 1$. In this case we have $\rho_{crit}(X) = E(X)$, hence our goal is to bound $E(X)$. Since $M(X) = 1$ we have for any s, $s(1-X(s)) \leq 1$ or $1-X(s) \leq 1/s$. Consequently,

$$E(X) = \int_0^p 1-X(s)\ ds =$$

$$\int_0^1 1\ ds + \int_1^p 1-X(s)\ ds \leq 1 + \int_1^p 1/s\ ds = 1 + \ln(p)$$

As desired. *q.e.d*

The distribution $B_p(1)$ has critical load $\ln(p) < \rho_{crit} = \frac{p}{p-1}\ln(p) < \ln(p) + 1$, which shows that the bound is very tight. It also shows that $B_p(1)$ has nearly optimal critical load.

4. THE APPROXIMATION EQUATIONS

h	α	Simulated AWT	Calculated AWT
2	0.2	1368.72	1408.87
2	0.4	577.91	583.34
2	0.6	214.06	215.64
2	0.8	79.60	79.86
2	1	33.83	34.02
2	1.2	18.07	17.99
2	1.4	11.38	11.38
2	1.6	7.83	7.88
2	1.8	5.25	5.73
3	0.2	2323.63	2657.09
3	0.4	704.26	769.79
3	0.6	185.89	197.60
3	0.8	52.70	54.32
3	1	19.89	20.35
3	1.2	11.76	12.03
3	1.4	9.74	10.12
3	1.6	9.09	10.20
3	1.8	9.59	11.14
4	0.2	11180.74	13982.31
4	0.4	1468.58	1789.23
4	0.6	255.05	287.54
4	0.8	54.47	58.84
4	1	18.82	19.72
4	1.2	12.17	13.01
4	1.4	13.28	15.15
4	1.6	22.72	28.05
4	1.8	143.89	181.70
5	0.6	484.40	597.45
5	0.8	68.04	77.26
5	1	21.04	22.81
5	1.2	15.24	17.31
5	1.4	28.93	36.34
6	0.8	98.86	117.45
6	1	25.71	29.04
6	1.2	22.46	27.19
7	0.8	173.79	221.62
7	1	33.70	39.98
7	1.2	43.60	55.88
8	1	50.17	60.72

Table 1: Comparison of actual waiting time from simulations of TAGS systems, with the estimate from the approximation formulas, which assume Poisson arrivals at all hosts

The average waiting time of a TAGS system with Poisson arrivals has no exact analytical formula, because the input stream to the second host and beyond is not Poisson anymore. We consider an approximation which assumes Poisson arrivals to all hosts. Once the assumption is made, an approximation to the average waiting time can be computed using the Pollaczek-Khinchine equation for an M/G/1 queue. This approximation was suggested in [8] and was used for calculating the performance of TAGS systems. It was also suggested that the approximation will over-estimate the average waiting time since the input streams, to all but the first host, tend to be more regular than Poisson, having near constant inter-arrivals. Such input streams should lead to better response times than a Poisson stream.

To analyze the true performance of TAGS and to compare it with the approximation equations, we developed an efficient computer simulation of a TAGS system. Instead of simulating the queues at the different hosts, the simulation follows the jobs through the system one by one. For each host we hold a "current time" counter which states when the host will become available to process the next job. As we track a new job through the system, these counters are updated and at the same time, the total amount of overhead time the job spends in the system is recorded. This approach turns out to be much more efficient than keeping track of the queues at each host. Using this approach we were able to easily run, on a standard PC, the TAGS system on 10^8 jobs, the number we fixed for each simulation run. The TAGS systems that were examined had Bounded Pareto job size distributions with varying values of $0 < \alpha < 2$ and varying numbers of hosts. The load per host was fixed to be 1/2, so that the total load was $\rho = h/2$. The smallest job size was of size 1 and the largest job size was $p = 10^4$. This value was chosen because for $\alpha = 2$, the probability of a job of size greater than s is about s^2, hence the probability for a job of size greater than 10^4 is approximately 10^{-8}. This means that for α close to 2, and 10^8 jobs in a simulation run we would not get jobs substantially greater than 10^4, therefore, there was no point in choosing a larger value for p in experiments across all values of α. The values of $s_1, \ldots, s_{h-1}$ for the system were chosen to be close to optimal for minimizing average waiting time in the approximate equations. While such a choice is not necessary for the validation process, we thought it would be the most interesting choice to explore.

The results showed that the average waiting time value which is computed using the approximation equations is always close to the value computed from the simulations. Moreover, as conjectured in [8], the computed value always over-estimates the actual average waiting time. As might be expected, the computed values are closest to the simulation results when the number of hosts is small. The computed values for $h = 2$ are essentially identical to the simulated values, except for the case $\alpha = 1.8$ where there was a 10% difference. For larger values of h the error can be as large as 20%, a value which we still consider to be very reasonable. The largest errors occur for the extreme values of α, away from the central value $\alpha = 1$, where the errors are smallest. A few more experiments where performed with the larger value $p = 10^6$ and $\alpha < 1$. The results are essentially the same, the approximate equations are fairly accurate and conservative.

Following these results we analyze in the next section the approximation equations under the assumption that they are conservative.

5. ASYMPTOTIC ANALYSIS WITH BOUNDED PARETO JOB SIZE DISTRIBUTION

In this section we analyze the performance of TAGS when the job size distribution is Bounded Pareto, with parameter α in the range $0 < \alpha < 2$. We also assume for simplicity that $\alpha \neq 1$, although the main results hold in that case as well.

5.1 The low load case, $\rho < 1$

Let us assume that the total system utilization satisfies $\rho < 1$. We consider a family of TAGS queueing systems, parametrized by $p > 1$. The number of hosts, h and the load ρ are fixed independently of p. The job size distribution is the Bounded Pareto distribution $B = B_p(\alpha)$ with α fixed. For a given p, the cutoffs, $s_i(p)$, $i = 1, \ldots, h-1$, are chosen from the set $\Delta_{h,p}$ of all cutoff parameters $1 < s_1 < \ldots < s_{h-1} < p$, which forms an $h-1$ dimensional open simplex. We denote the average normalized waiting time of the resulting TAGS system, as computed using the approximate

equations by $E(N)^{TAGS}(p,h,\rho,\alpha,s_1(p),\ldots,s_{h-1}(p)))$. We are interested in analyzing the asymptotic behavior of this quantity as $p\to\infty$. Of particular interest is the asymptotic behavior of $E(N)$ for an optimal choice of parameters $s_i(p)$. Let $E(N)^{OPT-T}(p,h,\rho,\alpha)$ denote the optimal (minimal) value of $E(N)^{TAGS}(p,h,\rho,\alpha,s_1(p),\ldots,s_{h-1}(p)))$ over the simplex $\Delta_{h,p}$.

Let $E(N)^{OPT-S}(p,h,\rho,\alpha)$ denote the corresponding optimal value of the normalized waiting time of `SITA` over the same simplex.

The next result is a comparison theorem between `SITA` and `TAGS` systems. Roughly speaking, the theorem states that at low loads the penalty for not knowing job sizes is bounded by a factor of 2 regardless of the value of α, the number of hosts h or the load ρ (as long as it low, $\rho<1$).

We will use the notation $f\sim g$ to denote two quantities f,g whose ratio tends to 1 as p tends to infinity.

THEOREM 5.1. *Fix $\rho<1$ and consider the family of Bounded Pareto distributions with α fixed, $0<\alpha<2$ and varying value of p.*

Assume that the approximation equations are conservative, then, as $p\to\infty$, the average waiting time in a `TAGS` *system with optimal cutoffs is at most twice the average waiting time of a* `SITA` *system with optimal cutoffs.*

More precisely, let

$$q=\frac{\alpha}{2-\alpha}$$

and let

$$\mu=\frac{(q^{h-1}-1)q}{q^h-1}$$

then

$$\frac{E(N^{OPT-T})}{E(N^{OPT-S})}\sim(\frac{2}{\alpha})^{\mu}\le 2 \tag{3}$$

Proof:

Assume that for all $i=1,\ldots,h$, $s_i(p)/s_{i-1}(p)\to\infty$ as $p\to\infty$, and that $\rho<1$. We show that

$$E(N)^{SITA}(\alpha,h,\rho,p,s_1(p),\ldots,s_{h-1}(p)) \tag{4}$$

$$\sim\sum_{i=1}^{h} f_i^{SITA}(\alpha,\rho)s_{i-1}^{-\alpha}s_i^{2-\alpha} \tag{5}$$

and

$$E(N)^{TAGS}(\alpha,h,\rho,p,s_1(p),\ldots,s_{h-1}(p)) \tag{6}$$

$$\sim\sum_{i=1}^{h} f_i^{TAGS}\alpha,\rho s_{i-1}^{-\alpha}s_i^{2-\alpha} \tag{7}$$

for some constants $f_i^{SITA}(\alpha,\rho), f_i^{TAGS}(\alpha,\rho)$ such that for $i<h$

$$f_i^{TAGS}(\alpha,\rho)=\frac{2}{\alpha}f_i^{SITA}(\alpha,\rho) \tag{8}$$

and for $i=h$

$$f_h^{TAGS}(\alpha,\rho)=f_h^{SITA}(\alpha,\rho) \tag{9}$$

Let p_i^{SITA} be the portion of jobs which pass through host i in a `SITA` system. These are precisely the jobs with size $s_{i-1}<s\le s_i$. These are also the jobs that pass through host i but not through host $i+1$ in the `TAGS` system. For bounded Pareto distributions we have

$$p_i^{SITA}=\frac{1}{1-(\frac{1}{p})^{\alpha}}(s_{i-1}^{\alpha}-s_i^{\alpha}) \tag{10}$$

We also compute p_i^{TAGS} the portion of jobs which pass through host i in a `TAGS` system, that is, jobs of size $s_{i-1}<s$.

$$p_i^{TAGS}=\frac{1}{1-(\frac{1}{p})^{\alpha}}(s_{i-1}^{\alpha}-p^{\alpha}) \tag{11}$$

By (10-11) and our assumption that $s_i(p)/s_{i-1}(p)\to\infty$ we get

$$p_i^{SITA}\sim p_i^{TAGS}\sim s_{i-1}^{-\alpha} \tag{12}$$

We also get the more precise formula

$$p_i^{TAGS}-p_i^{SITA}\sim s_i^{-\alpha} \tag{13}$$

except for the last host for which

$$p_h^{SITA}=p_h^{TAGS} \tag{14}$$

Let $E(B_{i,SITA}^j)$ be the j'th moment of service time in the i'th host of a `SITA` system. A simple calculation shows that

$$E(B_{i,SITA}^j)=\frac{\alpha s_{i-1}^{\alpha}}{1-(\frac{s_{i-1}}{s_i})^{\alpha}}\frac{s_{i-1}^{j-\alpha}-s_i^{j-\alpha}}{\alpha-j} \tag{15}$$

Following (15) the average job size for the i'th host with $\alpha<1$ satisfies

$$E(B_{i,SITA})\sim\frac{\alpha}{1-\alpha}s_{i-1}^{\alpha}s_i^{1-\alpha} \tag{16}$$

while for $\alpha>1$

$$E(B_{i,SITA})\sim\frac{\alpha}{\alpha-1}s_{i-1} \tag{17}$$

Similarly, for $\alpha<1$

$$E(B)\sim\frac{\alpha p^{1-\alpha}}{1-\alpha} \tag{18}$$

and for $\alpha>1$

$$E(B)\sim\frac{\alpha}{\alpha-1} \tag{19}$$

For the second moment we have

$$E(B_{i,SITA}^2)\sim\frac{\alpha}{2-\alpha}s_{i-1}^{\alpha}s_i^{2-\alpha} \tag{20}$$

In comparison, in a `TAGS` system, the jobs which pass through host i consist of those which do not pass onto host $i+1$ and those who do. The former have average service time moments $E(B_{i,SITA}^j)$ which is given by formula (15), while for the latter we get s_i^j at host i. We conclude that the j'th service time moment at host i in a `TAGS` system, $E(B_{i,TAGS})$ is the weighted average

$$E(B_{i,TAGS}^j)=\frac{p_i^{SITA}}{p_i^{TAGS}}E(B_{i,SITA}^j)+(1-\frac{p_i^{SITA}}{p_i^{TAGS}})s_i^j \tag{21}$$

From (21) and (12-17) we have for $\alpha<1$ and $i<h$

$$E(B_{i,TAGS})\sim(\frac{\alpha}{1-\alpha}+1)s_{i-1}^{\alpha}s_i^{1-\alpha}\sim\frac{1}{\alpha}E(B_{i,SITA}) \tag{22}$$

while for $\alpha<1$ and $i=h$ or for $\alpha>1$ and all i

$$E(B_{i,TAGS})\sim\frac{\alpha}{1-\alpha}s_{i-1}^{\alpha}s_i^{1-\alpha}\sim E(B_{i,SITA}) \tag{23}$$

Similarly for the second moment of the service time, when $i < h$ we have

$$E((B_{i,TAGS})^2) \sim \frac{2}{\alpha} E(B_{i,SITA}^2) \tag{24}$$

While for $i = h$ we have

$$E((B_{i,TAGS})^2) \sim E(B_{i,SITA}^2) \tag{25}$$

Let $\lambda = \rho/E(B)$ denote the job arrival rate to the whole system. Let λ_i^{SITA} and λ_i^{TAGS} denote the rate of job arrival at host i in a `SITA` and `TAGS` system respectively. The latter includes all jobs that will pass later on to the next host. We have

$$\lambda_i^{TAGS} = \lambda p_i^{TAGS} \sim \lambda p_i^{SITA} = \lambda_i^{SITA} \tag{26}$$

The utilization of host i in a `SITA` system is given by

$$\rho_i^{SITA} = \lambda_i^{SITA} E(B_{i,SITA})$$

with the corresponding equation

$$\rho_i^{TAGS} = \lambda_i^{TAGS} E(B_{i,TAGS})$$

for a `TAGS` system.
Using formulas (10,11,16-19,22,23) a simple computation shows that for all $0 < \alpha < 2$ and $1 \leq i \leq h$

$$1 - \rho_i^{SITA} \sim 1 - \rho_i^{TAGS} \tag{27}$$

In a `SITA` system the inter-arrival rate at host i is exponentially distributed with rate λ_i^{SITA}, hence, we can apply the Pollaczek-Khinchine formula for waiting time in an M/G/1 queue to each server individually, to obtain $E(W)_i^{SITA}$ the average waiting time at host i. The formula is

$$E(W_i^{SITA}) = \frac{\lambda_i^{SITA} E(B_{i,SITA}^2)}{2(1 - \rho_i^{SITA})} \tag{28}$$

Using our approximation assumption that the inter-arrival rate at host i is exponentially distributed we can apply the same formula with the corresponding quantities for a `TAGS` system to obtain $E(W_i^{TAGS})$, the average waiting time at host i. A comparison of the two expressions using formulas (26,24,25,27) shows that for $i < h$

$$E(W_i^{TAGS}) \sim \frac{2}{\alpha} E(W_i^{SITA}) \tag{29}$$

while for $i = h$

$$E(W_i^{TAGS}) \sim E(W_{SITA}) \tag{30}$$

We recall from the description of the `TAGS` algorithm that a job which finishes service at host i spends an additional time of $T_i = \sum_{j=1}^{i-1} s_j \leq (h-1)s_{i-1}$ being serviced at hosts $1, 2, \ldots, i-1$ and that the average excess service time satisfies $E(T) = \sum_{i=1}^{h} p_i T_i \leq (h-1)E(B)$ or

$$E(T)/E(B) \leq h - 1 \tag{31}$$

We have $E(N)^{TAGS} \geq E(N)^{OPT-T} \geq E(N)^{OPT-S}$. But by the asymptotic result in [12] we have $E(N)^{OPT-S} \to \infty$, therefore $E(T)$ is asymptotically negligible.

The average waiting time $E(W^{SITA})$ in a `SITA` system is the weighted sum

$$E(W^{SITA}) = \sum_{i=1}^{h} p_i^{SITA} E(W_i^{SITA}) \tag{32}$$

From the fact that $E(T)$ is negligible we see that for a `TAGS` systems, the corresponding formula is

$$E(W^{TAGS}) \sim \sum_{i=1}^{h} p_i^{TAGS} E(W_i^{TAGS}) \tag{33}$$

Using formulas (12,18,19,29,30,32) we obtain, after some straightforward computations, formulas (4, 6, 8, 9).

It has been shown in [13, 24, 1] that the optimal cutoffs for computing $E(N)^{OPT-S}$ satisfy the assumption that $s_{i+1}(p)/s_i(p) \to \infty$ for all i. The same argument also applies for the cutoffs used for computing $E(N)^{OPT-T}$. Thus we are reduced to computing the asymptotic value as $p \to \infty$ of the ratio

$$\frac{Min_\Delta \sum_{i=1}^{h} f_i^{TAGS}(\alpha,\rho) s_{i-1}^{-\alpha} s_i^{2-\alpha}}{Min_\Delta \sum_{i=1}^{h} f_i^{SITA}(\alpha,\rho) s_{i-1}^{-\alpha} s_i^{2-\alpha}}$$

where $\Delta = \Delta_{h,p}$. A simple scaling argument shows that the ratio is independent of p and depends only on the ratios $r_i = f_i^{TAGS}/f_i^{SITA}$. We have shown, (8, 9) that $r_i = \frac{2}{\alpha}$ for $i < h$ and that $r_h = 1$. Plugging these values into lemma 5.3 from [1] which computes such ratios we obtain the expression in (3).

We still need to show that $(\frac{2}{\alpha})^\mu \leq 2$. Since $\mu = \mu(\alpha) < 1$ for all values of $0 < \alpha < 2$, this is obvious for $\alpha \geq 1$. For $\alpha < 1$ we have $q < 1$. This implies $\mu = \frac{(q^{h-1}-1)q}{q^h-1} < q = \frac{\alpha}{2-\alpha}$. Differentiating $(\frac{2}{\alpha})^{\frac{\alpha}{2-\alpha}}$ it is easy to verify that it is an increasing function in the interval $(0, 1]$ with value 2 at $\alpha = 1$, which proves our assertion. *q.e.d*

5.2 Higher loads and the `T+F` policy

The results above can be extended to higher loads. Recall from section 3 the definition of the number of spare servers $\tilde{h}$. Let i be the minimal number of hosts, required to stabilize the `TAGS` system. We then let $\tilde{h} = \tilde{h}^{TAGS} = h - i - 1$. In generalizing to higher loads we will content ourselves with order of magnitude performance estimates. We claim that for higher loads the order of magnitude performance is given by the same formula as in the low load case with h replaced by $\tilde{h}$.

THEOREM 5.2. *Assume that the approximation equations provide the correct order of magnitude performance. For $\alpha > 1$, we have*

$$E(N)^{OPT-T}(\rho, \alpha, h, p) = \Theta(p^{\frac{2\alpha-2}{q^{\tilde{h}}-1}}) \tag{34}$$

, where $q = \frac{\alpha}{2-\alpha}$. For $\alpha < 1$, we have

$$E(N)^{OPT-T}(\rho, \alpha, h, p) = \Theta(p^{\frac{2-2\alpha}{q^{\tilde{h}}-1}}) \tag{35}$$

with $q = \frac{2-\alpha}{\alpha}$.

Proof: Consider the case $\alpha > 1$. Let i be the minimal number of hosts needed for a stable `TAGS` system. We recall the computation of i. We define recursively values $\tilde{s}_i$. Given $\tilde{s}_{i-1}$ we define $\tilde{s}_i$ to be such that in a `TAGS` system with $\tilde{s}_{i-1}$ and $\tilde{s}_i$ as cutoffs, the load on the i'th host is precisely 1. If $\tilde{s}_i$ does not exist because the load on the i'th host is always less than 1, then the minimal number of hosts which are required for a stable system is i. Consider the value of $\tilde{s}_{i-1}(p)$. We observe that $\tilde{s}_{i-1}(p)$ is an increasing function of p, since the coefficient c in the definition of the density is decreasing.

In addition it is bounded by $\tilde{s}_{i-1}(\infty)$, the value corresponding to the Pareto distribution $B(\alpha)$. The remaining load, $\rho_{rem}(\tilde{s}_{i-1}(p))$, coming from jobs of size at least $\tilde{s}_{i-1}(p)$ satisfies $\rho_{rem}(\tilde{s}_{i-1}(p)) < 1$. We choose $s_{i-1}(p) < \tilde{s}_{i-1}(p)$, such that the remaining load still satisfies $\rho_{rem}(s_{i-1}(p)) < 1$. We choose $s_1(p), ..., s_{i-2}(p)$ so that the load on the first, $i-1$ hosts of the TAGS system will be balanced. By the construction of $\tilde{s}_{i-1}(p)$ and the fact that $s_{i-1}(p) < \tilde{s}_{i-1}(p)$, the subsystem consisting of the first $i-1$ hosts will be stable. The remaining workload, which is handled by $\tilde{h}$ hosts is up to scaling again a Bounded Pareto distribution. We use our low load results to choose near optimal cutoffs. By the comparison between SITA and TAGS in theorem 5.1, the contribution to the normalized waiting time of the last $\tilde{h}$ hosts has order of magnitude in both SITA and TAGS. The asymptotic formulas (34,35) which are stated in the theorem are the known results for SITA which can be found in [13, 24, 1]. Since $s_{i-1}(p)$ is bounded, the contribution of the first $i-1$ is bounded and hence, asymptotically negligible. The optimal performance cannot have smaller order of magnitude, since the $i-1$ cutoff of any stable system cannot be larger than $\tilde{s}_{i-1}$ by definition. We conclude that the first $i-1$ hosts cannot asymptotically contribute to lowering the order of magnitude of average waiting time.

For $\alpha > 1$, we use a similar strategy, but we must proceed, from the other end of the job size range, from p. We define $\tilde{s}_{h-1}$ to be such that the load on the last host is precisely 1. Inductively, given $\tilde{s}_{h-j}$ we define $\tilde{s}_{h-j-1}$ to be such that the load on the $h-j$ host will be precisely 1. We can proceed this way to define $\tilde{s}_{\tilde{h}}, \tilde{s}_{\tilde{h}+1}, ..., \tilde{s}_{h-1}$. Since this provides an alternative procedure for calculating the minimal number of required hosts, the remaining load, consisting in this case of the load as seen by a single host which is responsible for jobs in the range $[1, \tilde{s}_{\tilde{h}}]$ will be less than 1. It is easy to check in analogy with the case $\alpha > 1$, that $p/\tilde{s}_{\tilde{h}}$ is bounded. The construction proceeds in analogy with the case of $\alpha > 1$. We choose $s_{\tilde{h}} > \tilde{s}_{\tilde{h}}$, such that the remaining load is still smaller than 1. We choose s_k, $h > k > \tilde{h}$, such that the last $h - \tilde{h}$ hosts are load balanced. It is easy to verify using our previous calculations that, for any $a > b > 0$, if a host is responsible for jobs in the range $[p/a, p/b]$, then its contribution to the normalized average waiting time is bounded. Therefore, asymptotically, the last $h-\tilde{h}$ do not contribute to the normalized waiting time. The first $\tilde{h}$ cutoffs are chosen near optimally, to insure the requested order of magnitude performance. *q.e.d*

We note that both theorem 5.2 and theorem 5.1 are valid more generally for systems with heterogeneous hosts, a setting which was first considered (in the SITA case) in [7]. In this setting not all hosts have identical capabilities. The job size distribution is given with respect to some reference host. Each host in the system, say host i, has an associated power coefficient c_i. A job which takes t time on the reference host, takes t/c_i time on host i. In this setting, apart from choosing cutoffs, we have to choose for each job size range $[s_{i-1}, s_i]$, which host will service jobs up to size s_i. In this setting, the low load condition is replaced by the condition that the strongest host (largest c_i) can handle the entire load.

The arguments that lead to theorem 5.1 boil down to a comparison of waiting times at individual hosts in the system without relying on different hosts having equal strength.

alpha	h=2	h=3	h=4	h=5	h=6	h=7	h=8
0.1	2	3	4	9	NA	NA	NA
0.2	2	2	4	7	34	NA	NA
0.3	2	2	4	6	14	NA	NA
0.4	2	2	4	5	9	33	NA
0.5	2	2	3	5	7	13	NA
0.6	2	2	3	4	6	9	14
0.7	2	2	3	4	5	7	9
0.8	2	2	3	4	5	6	7
0.9	2	2	3	3	4	5	6
1.0	2	2	3	3	4	5	6
1.1	2	2	3	3	4	5	6
1.2	2	2	3	4	4	6	7
1.3	2	2	3	4	5	7	11
1.4	2	2	3	4	6	NA	NA
1.5	2	2	3	5	NA	NA	NA
1.6	2	2	3	6	NA	NA	NA
1.7	2	2	4	NA	NA	NA	NA
1.8	2	2	4	NA	NA	NA	NA
1.9	2	2	5	NA	NA	NA	NA

Table 2: The values of $\tilde{h}^{TAGS}$, when $\rho = h/2$

They also do not depend on the loads on the hosts because of equation (27). As a result, the theorem holds in the heterogeneous setting as well. The proof of theorem 5.2 also goes through, however, we have to redefine the spare server number for the heterogeneous setting. For $\alpha < 1$ this is done by ordering the hosts from the weakest to the strongest and checking how many hosts are in a low load system. When $\alpha > 1$ we order the hosts from the strongest to the weakest and check for the number of hosts in the low load case. The reason is that the load burden in the case $\alpha < 1$ falls on the large job hosts, while for $\alpha > 1$ the opposite is true.

Let $\tilde{h}^{SITA} = \tilde{h}^{FIFO} = h - [\rho]$. We note that this definition is consistent with the definition for TAGS, being $h-i-1$, where i is the minimal number of hosts required for a stable SITA or FIFO system. Following the arguments above, the asymptotic, order of magnitude, performance of SITA, depends on $\tilde{h}^{SITA}$ and is given by replacing $\tilde{h}$ by $\tilde{h}^{SITA}$ in the statement of theorem 5.2. Similarly, it is known (see the next section) that the order of magnitude performance of FIFO depends on $\tilde{h}^{FIFO}$.

The table charts the minimal number of hosts which are needed to stabilize a TAGS system when $\rho = h/2$. The corresponding values for SITA or FIFO are $[h/2] + 1$. The symbol NA, denotes the case that $h/2 \geq \rho_{crit}$ and hence, TAGS cannot be stable regardless of the number of hosts. As can be easily seen, in many cases the minimal number of hosts which are required to stabilize TAGS is substantially bigger than in the case of SITA, and consequently the performance has worse order of magnitude, since the number of effective hosts is smaller.

We see that a major problem of TAGS in comparison with SITA is that $\tilde{h} = \tilde{h}^{TAGS}$ either does not exist, due to inherent instability, or is much smaller than $\tilde{h}^{SITA}$.

The arguments in theorem 5.2 suggest that we may substantially improve $\tilde{h}$ and consequently, the order of magnitude performance of TAGS at higher loads if we combine TAGS with work preserving policies like FIFO or even random assignment, which are better at consuming load. Our improved version of TAGS which combines elements of FIFO will be denoted by T+F. For $\alpha > 1$, we let $\tilde{s}$ be such that the remaining load ρ_{rem} of jobs which are greater than $\tilde{s}$ is 1.

Our plan is to service all incoming jobs with $[\rho]$ or $[\rho]+1$ hosts, working in FIFO or random assignment mode. However, if a job exceeds $\tilde{s}+\varepsilon$ (for some small ε) time units it is killed and sent to the remaining servers which process jobs according to the TAGS policy. It is obvious that such a FIFO system with $[\rho]+1$ hosts will be stable since the system experiences less work than a corresponding FIFO system which processes all jobs to completion. However, in some cases $[\rho]$ hosts will suffice for a stable system, in which case we will use only $[\rho]$ hosts. The number of remaining hosts will be $\tilde{h}^{T+F}$. To determine the precise condition for $[\rho]$ hosts to suffice in the FIFO subsystem, we provide an asymptotic formula for $\tilde{s}$. A comparison of the load coming from jobs of size, between 1 and $\tilde{s}$, and the total load with $p \to \infty$ shows that $\tilde{s}$ satisfies $1-\tilde{s}^{1-\alpha} \sim \frac{\rho-1}{\rho}$, or $\tilde{s}^{1-\alpha} \sim 1-\rho$. Since the system will also spend $\tilde{s}$ time units on each job of size at least s, a simple calculation shows that the actual load experienced by the hosts in the FIFO system will be $\rho(1-\frac{1}{\alpha}\tilde{s}^{1-\alpha})$. Plugging the asymptotic value of $\tilde{s}$ into the last expression and requiring that it is at most $[\rho]$, yields $\rho-\frac{1}{\alpha}<[\rho]$ or $\frac{1}{\alpha}>\rho-[\rho]$.

Similarly, when $\alpha<1$, we define $\tilde{s}$ to be such that the load of a single server in a TAGS system servicing jobs in the range $[1,\tilde{s}]$ is precisely 1. The remaining load, coming from jobs whose size is in the range $[\tilde{s},p]$ will satisfy $\rho-1 \le \rho_{rem} \le \rho$ and hence, will require either $[\rho]+1$ or $[\rho]$ hosts for a stable FIFO system. The remaining $\tilde{h}^{T+F}$ hosts will manage all incoming jobs using TAGS with the last cutoff being $\tilde{s}-\varepsilon$. Jobs which are of size greater than $\tilde{s}-\varepsilon$ are sent to begin service from scratch in the FIFO system. Using formula (22) it is easy to see that $[\rho]$ hosts suffice in the FIFO subsystem if $\alpha>\rho-[\rho]$. The asymptotic order of magnitude performance of T+F is given by theorem 5.2 with $\tilde{h}$ being replaced by $\tilde{h}^{T+F}$, which satisfies

$$\tilde{h}^{SITA}-1 \le \tilde{h}^{T+F} \le \tilde{h}^{SITA}$$

a vast improvement over TAGS.

The basic strategy of the T+F policy can be applied to more general job size distributions. However, as can be seen already in the Bounded Pareto case, the details of the implementation vary substantially with the workload. Basically, the FIFO managed hosts should be responsible for values of s such that $s(1-X(s))$ is large. This is difficult to implement when the workload is such that $s(1-X(s))$ is not monotonic, or when the workload is dynamic.

5.2.1 The analysis of some variants

We can also consider the performance of TAGS under the relaxed assumption that jobs can be resumed on the next host from the point in which they were stopped in the previous host. This setting was explored in [6] and in [3].

While this assumption improves the stability of TAGS and hence the number of effective servers, we show that it has no effect on the asymptotic performance of TAGS in the low load case when $\rho<1$ and the workload is Bounded Pareto. To see this, we again consider the case where $s_i/s_{i-1}\to\infty$. In that case, the work done on a job in the first $i-1$ servers is $\sum_{j=1}^{i-1} s_j < (1+\varepsilon)s_{i-1}$, for any $\varepsilon>0$ and p large enough. Let $\tilde{s}_{i-1}=(1+\varepsilon)s_{i-1}$ and consider a host in a TAGS system which handles jobs in the range $[\tilde{s}_{i-1},s_i]$. Assume also that from each job whose original size was $s \ge s_{i-i}$ we subtracted $\tilde{s}_{i-1}$ work. Such a host experiences less traffic and has less work for each job than the i'th host in the TAGS system where we assume that work is resumed. However, a short look at formulas (12,15,28,29,30) shows that the i'th host in the new system has asymptotically the same contribution to waiting time as the i'th host of the original system and our conclusion follows.

6. A SIMULATION COMPARISON BETWEEN THE TAGS AND FIFO POLICIES

In this section we will compare empirically the TAGS policy with the FIFO policy for Bounded Pareto job size distributions. In the previous section we analyzed the performance of TAGS on such workloads. The analysis of the performance of FIFO in [17, 16, 18, 19, 20, 25] leads to the following estimate

$$Pr(W^{FIFO}>s)=\Theta((1-X_e(s))^{\tilde{h}}) \tag{36}$$

6.1 The simulation comparison

We provide a detailed simulation based comparison between FIFO and TAGS policies for various values of h and α. We keep the load per host fixed at 0.5, which means that the load on the system is $\rho=h/2$. We will use the analysis of the previous sections to provide a partial explanation of the results.

We note that a comparison for two hosts was provided in [8], however, it was based on an approximate calculation for the performance of FIFO which proved to be far from accurate.

In addition, the normalization which was used gave very different ranges (values of p) to different values of α. For example, when $\alpha=0.2$ it assumed a Bounded Pareto job size distribution with $p=10^{66}$, while for $\alpha=2$ it assumed $p=10^6$. Thus it was difficult to compare results for different values of α. We fix the value $p=10^6$ across all values of α. Fixed p normalization was first used in [7]. It is a more natural normalization since it is time unit invariant.

As a consequence of these changes our results and insights are somewhat different than those which can be found in [8].

We started by using 10^8 jobs in the simulation. We observe that it is hard to simulate heavy-tailed distributions with large range. In particular, for $\alpha>1$ we had difficulty in covering the entire range of jobs and therefore for $\alpha \ge 1.3$ we used 10^9 jobs instead of 10^8. This still did not solve the problem, but we feel that it is unlikely that real systems will have many more jobs than that.

The results are summarized in table 3. The rows of the table list the value of the parameter α, while the columns list the number of hosts h in the system. We recall that we fix $p=10^6$ throughout and $\rho=h/2$. each table entry consists of two numbers. The left number is the normalized average waiting time under the TAGS policy, while the number on the right is the same for FIFO. The numbers are rounded downward to the nearest integer. Sometimes the letter T appears in the entry. That indicates that the performance of TAGS was superior to that of FIFO. In other cases the letters NA appear as the table entry. That indicates that there is no stable TAGS system with these parameters, hence FIFO wins by default.

The results for 2 servers convey the situation which was examined in section 5. The load is $\rho=1$ which is at the edge of the low load assumption $\rho<1$ for the analysis. We clearly see in the results for TAGS the approximate symmetry between α and $2-\alpha$, which comes from the symmetry

for SITA and the comparison theorem. We also see that the performance is worst near $\alpha = 1$ and gets better as we move towards the extreme values $\alpha = 0$ and $\alpha = 2$. For FIFO, we see that performance for $\alpha \geq \frac{h+1}{h} = 3/2$ is indeed good and is superior to that of TAGS as expected from the asymptotic analysis. The normalized waiting time then climbs steeply until reaching a maximal value at $\alpha = 0.9$ and then falls off less steeply as we approach $\alpha = 0$. The order of magnitude of the normalized average waiting time asymptotics, suggest that the maximal value should be at $\alpha = 1$. However, if we follow the heuristic of considering formula (36) as an equality, the resulting calculations suggest that for $\alpha < 1$, with p fixed, as $\tilde{h}^{FIFO}$ increases, $E(N)$, decreases like a function of the form $\tilde{h}^{-\frac{1}{1-\alpha}}$. This heuristic argument suggests that for fixed p, the effect of the number of hosts is stronger for larger values of α and consequently, we expect the maximal value to drift towards smaller values of α as $\tilde{h}$ increases. This phenomenon is indeed observed in our table.

Also as suggested by the asymptotic theory, TAGS starts outperforming FIFO once $\alpha < 3/2$. However, for α small the asymptotic difference is less important since p^{α} is small and the higher loads in a TAGS due to job loss take effect, leading FIFO to outperform TAGS when $\alpha < 0.3$.

The results of section 5 show the asymptotic low load performance of TAGS is worst at $\alpha = 1$ and improves towards $\alpha = 0$ and $\alpha = 2$. On the other hand the results of section 3 show that the load handling capabilities of TAGS have the exact opposite behavior, they are best at $\alpha = 1$ and deteriorate rapidly towards $\alpha = 0$ and $\alpha = 2$. When $h = 3$ (and $\rho = 1.5$) these contrasting tendencies essentially cancel each other and the normalized waiting time shows little variation across the different values of α.

The results for $h \geq 4$ show the strong effects of the increasing load. Whereas for FIFO and SITA the values near $\alpha = 1$ pose the biggest challenge, for TAGS, the performance at $\alpha = 1$ is best since that is where TAGS can handle load better. The range of α in which TAGS outperforms FIFO decreases steadily, for $\alpha > 1$. The reason is that the performance of FIFO is better asymptotically than that of SITA, let alone TAGS. On the other hand, for $\alpha < 1$ where the performance of FIFO is not as strong it is more an issue of the different load handling capabilities and hence TAGS is able to hold on to a lead a bit longer. Meanwhile, the increased load allows stable TAGS in only a very small range of α values centered at $\alpha = 1$. This phenomenon which follows easily from our load handling analysis was not noted at all in previous studies. With 6 hosts there is still substantial improvement of TAGS over FIFO for $\alpha = 0.9$ and $\alpha = 1$. beyond 8 hosts the performance of FIFO is better than that of TAGS for all values of α. Apart from the stability conditions, the other factor is that we are not in the asymptotic regime anymore. For $\alpha = 1$, where TAGS suffers least from stability issues, given 8 hosts, each of the hosts in a TAGS system is responsible for a small range of approximate size $10^{6/8} < 6$ and lowering the range still further would have little effect. On the other hand, every additional host will improve the constants involved in the performance of FIFO more substantially.

alpha	h=2	h=3	h=4	h=6	h=8
0.1	4, 2	7, 1	99, 1	NA	NA
0.2	5, 4	7, 2	34, 1	NA	NA
0.3	6, 10 T	7, 4	22, 2	NA	NA
0.4	7, 22 T	7, 10 T	15, 5	NA	NA
0.5	9, 51 T	6, 23 T	9, 12 T	NA	NA
0.6	11, 112 T	6, 48 T	7, 26 T	NA	NA
0.7	12, 219 T	5, 94 T	5, 49 T	11, 16 T	NA
0.8	15, 407 T	5, 150 T	4, 77 T	4, 24 T	12, 8
0.9	16, 555 T	5, 185 T	3, 79 T	3, 20 T	4, 5 T
1.0	15, 370 T	4, 151 T	3, 63 T	2, 12 T	3, 3 T
1.1	14, 175 T	4, 63 T	3, 30 T	3,3 T	4, 1
1.2	16, 124 T	5, 11 T	4, 6 T	4,1	12, 0
1.3	13, 51 T	5, 11 T	4, 1	11, 0	NA
1.4	11, 14 T	5, 3	5, 0	NA	NA
1.5	8, 4	5, 1	7, 0	NA	NA
1.6	7, 2	6, 0	17, 0	NA	NA
1.7	5, 1	6, 0	22, 0	NA	NA
1.8	3, 1	5, 0	NA	NA	NA
1.9	3, 0	6, 0	NA	NA	NA
2.0	2, 0	5, 0	NA	NA	NA

Table 3: Comparison of TAGS and FIFO

7. CONCLUSION

In this paper we studied the TAGS policy, which assigns jobs to servers in the non-preemptive regime, with unknown job sizes. We examined, the stability conditions of the assignment policy and obtained a formula for the maximal load that a TAGS system can support in terms of the job size distribution function. When arrivals are Poisson and the job size distribution is Bounded Pareto, we verified that some approximation equations for average waiting time are conservative and rather accurate. It would be interesting to prove, rather than verify, such claims. We examined the approximation equations and showed that at low loads, the penalty for not knowing the job size is at most a factor of 2, regardless of α, the load (as long as its low) or the number of hosts. It would be interesting to establish a low load comparison theorem for general distributions. We developed a variant policy which combines TAGS and FIFO, which can handle higher loads much better than TAGS. We compared TAGS to FIFO and found that, in general TAGS is more appropriate for values with $\alpha < 1.1$, where FIFO tends to exhibit more difficulties. The improvement over FIFO is sometimes significant. However, as the number of hosts increases mildly, FIFO becomes the better policy across all values of α.

8. REFERENCES

[1] E. Bachmat and H. Sarfati, Analysis of SITA policies, *Performance Evaluation*, 67(2), 102-120, 2010.

[2] Basher N., A. Mahanti, A. Mahanti, C. Williamson, and M. Arlitt, A comparative analysis of web and Peer-to-Peer traffic, in *Proceedings of WWW2008*, 287-296, 2008.

[3] Broberg J., Z. Tari, P. Zeephongsekul, Task assignment with work-conserving migration, *Parallel Computing*, 32, 808830, 2006.

[4] Crovella M.E., Taqqu M.S. and Bestavros A., Heavy-tailed probability distributions in the world wide web. In *A practical guide to heavy tails*, Chapman and Hall, New York, Chapter 1, 1-23, 1998.

[5] Down D., S.P. Meyn, Stability of acyclic multiclass queueing networks. *IEEE Trans. Automat. Control*, 40, 916919, 1995.

[6] El-Taha M., B. Maddah, Allocation of service time in a multiserver system. *Management Science*, 52(4), 623637, 2006.

[7] Feng H., V. Misra, D. Rubenstein, Optimal state-free, size-aware dispatching for heterogeneous M/G/-type systems, *Performance evaluation*, 62, 475-492, 2005.

[8] Harchol-Balter M., Task assignment with unknown duration, *Journal of the ACM*, 49(2), 260-288, 2002.

[9] Harchol-Balter M., M. Crovella, Method and apparatus for assigning tasks in a distributed server system, U.S. patent number 6,223,205, 2001.

[10] Harchol-Balter M., M. Crovella, C. Murta, Task assignment in a distributed system: Improving performance by load unbalancing, *Proceedings of SIGMETRICS 98*, poster session, 1998. Full version published as a Boston University technical report number BUCS-TR-1997-019.

[11] Harchol-Balter M., M. Crovella. C. Murta, On choosing a task assignment policy for a distributed server system, *IEEE Journal of parallel and distributed computing*, Vol. 59, 204-228, 1999.

[12] Harchol-Balter M., A. Scheller-Wolf, A. Young, Surprising Results on Task Assignment in Server Farms with High-Variability Workloads, *Proceedings of ACM SIGMETRICS 2009 Conference on Measurement and Modeling of Computer Systems*, 2009.

[13] M. Harchol-Balter and R. Vesilo, To Balance or Unbalance Load in Size-Interval Task Allocation, *Probability in the Engineering and Informational Sciences* , vol. 24, 219-244, 2010.

[14] Mitzenmacher M, Dynamic Models for File Sizes and Double Pareto Distributions, *Internet Mathematics*, 1(3), 305-334, 2004.

[15] Riska A., W. Sun, E. Smirni, G. Ciardo, AdaptLoad: effective balancing in clustered web servers under transient load conditions, *Proceedings of the 22nd International Conference on Distributed Computing Systems, (ICDCS)*, 104-112, 2002.

[16] Scheller-Wolf A., Necessary and Sufficient Conditions for Delay Moments in FIFO Multiserver Queues: Why s Slow Servers are Better than One Fast Server for Heavy-Tailed Systems *Operations Research*, 51, 748-758, 2003.

[17] Scheller-Wolf A., Further delay moment results for FIFO multiserver queues. *Queueing Systems*, 34, 387400, 2000.

[18] Scheller-Wolf A., K. Sigman, New bounds for expected delay in FIFO GI/GI/c queues. *Queueing Systems*, 26, 169186, 1997.

[19] Scheller-Wolf A., R. Vesilo, Structural interpretation and derivation of necessary and sufficient conditions for delay moments in FIFO multiserver queues. *Queueing Systems*, 54, 221232, 2006.

[20] Scheller-Wolf A., R. Vesilo, Structural Interpretation and Derivation of Necessary and Sufficient Conditions for Delay Moments in FIFO Multiserver Queues *Queueing Systems* 54, 221-232, 2007.

[21] Schroeder B., M. Harchol-Balter, Evaluation of Task Assignment Policies for Supercomputing Servers: The Case for Load Unbalancing and Fairness, *Proc. of the 9th IEEE Symposium on High Performance Distributed Computing (HPDC)*, 2000.

[22] Tari Z., J. Broberg, A. Zomaya, R. Baldoni, A least flow-time first load sharing approach for distributed server farm, *Journal of Parallel and Distributed Computing*, 65, 832842, 2005.

[23] Thomas N.,Comparing job allocation schemes where service demand is unknown, *Journal of Computer and System Sciences*, 74, 10671081, 2008.

[24] R. Vesilo, Asymptotic analysis of load distribution for size-interval task allocation with bounded Pareto job sizes, *In Proceedings of the 14th IEEE International Conference on Parallel and Distributed Systems (ICPADS08), 810 December, Melbourne, Australia*, pp. 129137, 2008.

[25] Whitt W., The impact of a heavy-tailed service-time distribution upon the M/GI/s waiting-time distribution. *Queueing Systems*, 36, 7187, 2000.

[26] Zhang Q., A. Riska, W. Sun, E. Smirni, G. Ciardo, Workload-aware load balancing for clustered Web servers" , *IEEE Transactions on Parallel and Distributed Systems*, 16(3), 219-233, 2005.

[27] Zhang Q., N. Mi, A. Riska, E. Smirni, Load Balancing for Performance Differentiation in Dual-Priority Clustered Servers, *in the Proceedings of the 3rd International Conference on the Quantitative Evaluation of Systems (QEST)*, 385-394, 2006.

[28] Zhang Q., N. Mi, A. Riska, E. Smirni, Performance-Guided Load (Un)balancing under Autocorrelated Flows, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 19(5), 652-665, 2008.