Asymptotic Optimality of Best-Fit for Stochastic Bin Packing

Javad Ghaderi
Columbia University
jghaderi@ee.columbia.edu

Yuan Zhong
Columbia University
yz2561@columbia.edu

R Srikant
University of Illinois at Urbana-Champaign
rsrikant@illinois.edu

ABSTRACT

In the static bin packing problem, items of different sizes must be packed into bins or servers with unit capacity in a way that minimizes the number of bins used, and it is well-known to be a hard combinatorial problem. Best-Fit is among the simplest online heuristics for this problem. Motivated by the problem of packing virtual machines in servers in the cloud, we consider the dynamic version of this problem, when jobs arrive randomly over time and leave the system after completion of their service. We analyze the fluid limits of the system under an asymptotic Best-Fit algorithm and show that it asymptotically minimizes the number of servers used in steady state (on the fluid scale). The significance of the result is due to the fact that Best-Fit seems to achieve the best performance in practice.

1. INTRODUCTION

Cloud computing has gained enormous momentum recently. Cloud customers outsource their storage and computation needs to a cloud data center consisting of a large number of servers. The computation/storage requirement of a customer usually comprises a handful of virtual machines (VMs), with certain amounts of resource requirement for CPU, physical memory, etc, placed at different servers. The VMs can share the same server if they do not violate the capacity constraints of the server. For the purpose of scalability and cost efficiency of the data center, it is necessary to design optimal algorithms for placement of VMs in the servers to minimize consumption of network resources.

In this paper, we consider a data center consisting of an infinite number of servers. VM's arrive randomly over time, they are placed in the servers subject to capacity constraints, and leave the system after some random service time. The goal is to minimize the average number of used servers. The VM placement algorithm has to be online, i.e., each VM upon its arrival has to be placed in some server that can accommodate it. The static one-dimensional version of this problem is closely related to the classical bin packing problem: given a sequence of r objects $L_r = a_1, \cdots, a_r \in (0, 1]$, pack them into bins of unit capacity so as to minimize the number of used bins. The bin packing problem is known to be NP-hard and many approximation algorithms have been developed that can provide the optimal number of bins up to an approximation factor. One of the simplest algorithms among these is the Best-Fit algorithm, in which objects are packed in an online manner, with each object being placed in the "tightest" bin (i.e., with the minimum residual capacity that can still accommodate the object), and, if no such bin is found, the object is placed in a new bin. Despite its simplicity, Best-Fit is known to perform well in practice. It was first proved in [1] that under Best-Fit, in the worst case, the number of bins used is asymptotically within a factor 1.7 of the optimal number, as the list size $r \to \infty$. In the case that the objects are drawn from some general distribution, the expected asymptotic performance ratio of Best-Fit is strictly greater than one [2]. In this paper, we consider the stochastic version of this problem when *objects arrive randomly over time and leave the system after completion of their service*, as considered previously by Stolyar and Zhong [3], [4], [5], which in turn is an infinite-server version of a model originally proposed by Maguluri et al. [6].

Research supported in part by NSF Grant ECCS-1202065.

2. SYSTEM MODEL

We assume that there are n different types of jobs. Jobs of type i arrive according to a Poisson process with rate $\lambda_i r$, and remain in the system for an exponentially distributed amount of time with mean $1/\mu_i$, with $r > 0$ being some scaling parameter. First, consider the one-dimensional problem where each job is represented by its size (scalar). There is an infinite number of servers, each with (normalized) unit capacity. Jobs of type i require a fraction s_i of the capacity. Given the job profiles, there is a finite set of possible server configurations, where each configuration is a vector $k = (k_1, \cdots, k_n)$ with k_i representing the number of type i jobs in the server. The packing constraint imposes that $\sum_{i=1}^n s_i k_i \leq 1$. We use $\mathcal{K}$ to denote the set of all possible configurations. Note that the number of jobs of type i in the system in steady state is simply a Poisson random variable with mean $\rho_i r$, where $\rho_i = \lambda_i/\mu_i$, regardless of placement algorithm. Without loss of generality, we can normalize such that $\sum_i \rho_i = 1$. For each $k \in \mathcal{K}$, let $X_k(t)$ be the number of servers in configuration k at time t. Thus, the total number of servers in use at time t is $\sum_{k\in\mathcal{K}, k\neq 0} X_k(t)$. Note that r controls the population size in the system and the asymptotic (expected) performance ratio is defined as $r \to \infty$.

3. ASYMPTOTIC BEST-FIT ALGORITHM

We define the *level of configuration* k as

$$u_k = \sum_i k_i s_i. \tag{1}$$

Thus $1 - u_k$ is the residual capacity of configuration k. Upon

a job arrival of type i at time t, it is placed in a server with configuration k, $k+e_i \in \mathcal{K}$, with probability

$$P_{k,i}^{(a)} = \frac{X_k(t)\exp(\frac{2s_i}{\sqrt{a}}u_k)}{\sum_{\tilde{k}\neq 0:\tilde{k}+e_i\in\mathcal{K}} X_{\tilde{k}}(t)\exp(\frac{2s_i}{\sqrt{a}}u_{\tilde{k}}) + X_0(t)} \quad (2)$$

where $X_0(t) = \lceil \exp(-1-\frac{1}{a})Y(t)\rceil$ is a designated set of empty servers (zero-servers), $Y(t)$ is the total number of jobs in the system, and $a > 0$ is some number specified by the algorithm.

Let $P_{k,i}(t) = \lim_{a\to 0} P_{k,i}^{(a)}(t)$. Also Define

$$\mathcal{K}_i^\star(t) = \underset{k:k+e_i\in\mathcal{K}, X_k(t)>0}{\arg\max} u_k, \quad (3)$$

to be the set of available configurations with the maximum level that can accommodate job i. Then, if $\mathcal{K}_i^\star(t) \neq \varnothing$,

$$P_{k,i}(t) = \frac{X_k(t)}{\sum_{\tilde{k}\in\mathcal{K}_i^*(t)} X_{\tilde{k}}}; \text{ if } k \in \mathcal{K}_i^\star(t) \quad (4)$$

$$P_{k,i}(t) = 0; \text{ otherwise.} \quad (5)$$

In other words, as $a \to 0$, with high probability the algorithm chooses one of the servers whose configuration is in $\mathcal{K}_i^\star$ uniformly at random, i.e., it places the job in one of the "tightest" servers uniformly at random. If there is no partially filled server that can accommodate the job, i.e., $\mathcal{K}_i^\star(t) = \varnothing$, the job is placed in a new server. Hence the algorithm approaches the true *Best-Fit* as $a \to 0$.

THEOREM 1. *Let $BF^{(a)}(r)$ and $OPT(r)$ be respectively the number of servers used by the asymptotic Best-Fit algorithm (with parameter a) and the optimal algorithm in steady state. Then*

$$\lim_{r\to\infty} \frac{\mathbb{E}\left[BF^{(a)}(r)\right]}{\mathbb{E}\left[OPT(r)\right]} \leq 1 + \frac{(1+e^{-1})}{\sum_i \rho_i s_i}\left(\sqrt{a} + a\log|\mathcal{K}|\right) + a\log\left(\lceil\frac{1}{s_{min}}\rceil!\right),$$

where $s_{min} = \min_i s_i$ and $|\mathcal{K}|$ is the size of the configuration space. Note that the right-hand-side goes to 1 as $a \to 0$.

Theorem 1 suggests that Best-Fit is asymptotically optimal, although a rigorous argument for the exchange of limits in a and r is not trivial. This contrasts with the lack of throughput optimality of Best-Fit in [6].

3.1 Extension to multidimensional packing

So far, we assumed that the job sizes are scalars. In general, each job s_i could be a d-dimensional vector of resource requirements, i.e., $s_i = (s_{i1}, \cdots, s_{id})$, where $s_{i\ell}$ is the required fraction of resources of type ℓ, $1 \leq \ell \leq d$ from a server. Although there is no standard definition of Best-Fit in this case, it is possible to define a Generalized Best-Fit where the level of each configuration k is defined as the weighted sum of the levels of dimensions, i.e.,

$$u_k = \sum_{\ell=1}^{d} w_\ell \sum_i k_i s_{i\ell}. \quad (6)$$

Here w_ℓ's are some nonnegative fixed weights. For example, memory is usually the main constraint in cloud servers [7], thus the cloud operator can put a larger weight for the memory. The job placement probabilities for the corresponding asymptotic Generalized-Best-Fit is defined as

$$P_{k,i}^{(a)}(t) = \frac{X_k(t)\exp(\frac{2\sum_\ell w_\ell s_{i\ell}}{\sqrt{a}}u_k)}{\sum_{\tilde{k}\neq 0:\tilde{k}+e_i\in\mathcal{K}} X_{\tilde{k}}(t)\exp(\frac{2\sum_\ell w_\ell s_{i\ell}}{\sqrt{a}}u_{\tilde{k}}) + X_0(t)} \quad (7)$$

Then our analysis and result for the scalar case is carried over to the multi-dimensional case. At present, it is not clear what the effect of weights w_ℓ is on the performance of the algorithm.

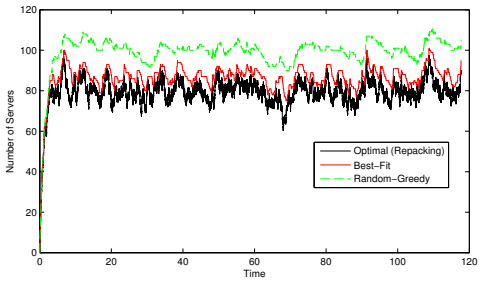

Figure 1: The number of servers used at each time: Best-Fit vs. Random-Greedy

4. SIMULATIONS

The goal of the simulations is to compare the performances of Best-Fit and the Random-Greedy algorithm [5]. In the Random-Greedy algorithm, job is placed uniformly at random in one of the feasible servers, independently of their levels. The asymptotic version of Random-Greedy is also optimal as $r \to \infty$ as shown in [5]. However, we expect that Best-Fit performs better for finite values of r as it yields less capacity waste compared to Random-Greedy. For simulations here, we considered the following scalar job sizes: $s_1 = 0.8, s_2 = 0.5, s_3 = 0.3, s_4 = 0.1$, with $\lambda_1 = 1, \lambda_2 = 4, \lambda_3 = 7, \lambda_4 = 31$, and $r = 10$. Mean service times are all one. Figure 1 demonstrates the number of servers used at each time. We have also plotted the performance of the optimal algorithm which is allowed to do repacking and solves an associated LP to find the right packing at each time a job arrives/departs. In this simulation, Best-Fit gave a saving of more than 12% in terms of the average number of servers used, compared to Random-Greedy.

5. PROOF SKETCH OF THEOREM 1

Fluid Limits: Proceeding along the lines of Stolyar-Zhong [5], we consider a sequence of systems indexed by r. For each $k \in \mathcal{K}$, let $X_k^{(r)}(t)$ be the number of servers in configuration k, starting from some initial state $X^{(r)}(0)$ in the r-th system. The goal is to minimize the number of used servers, scaled down by r. Let $A_{k,i}^{(r)}(t)$ be the number of arrivals that are placed in bins of configuration k, up to time t. Similarly, let $D_{k,i}^{(r)}(t)$ be the number of departures from such bins up to time t. Let $\dot{f}(t) = \frac{\mathrm{d}}{\mathrm{d}t}f(t)$ denote the time derivative of function f.

PROPOSITION 1. *Suppose $\frac{1}{r}X^{(r)}(0) \to x(0)$, then every*

sequence of r, has a subsequence such that

$$\frac{1}{r}(X^{(r)}, A^{(r)}, D^{(r)}) \to (x, a, d), \ \ u.o.c,$$

along the subsequence. Further, at any regular point, (x, a, d) satisfy the following fluid limit equations:

$$\begin{aligned}\dot{x}_k(t) &= [\sum_{i:k-e_i\in\mathcal{K}} \dot{a}_{k-e_i,i}(t) + \sum_{i:k+e_i\in\mathcal{K}} \dot{d}_{k+e_i,i}(t)] \\ &\quad -[\sum_{i:k+e_i\in\mathcal{K}} \dot{a}_{k,i} + \sum_{i:k-e_i\in\mathcal{K}} \dot{d}_{k,i}] \qquad (8)\end{aligned}$$

$$\dot{a}_{k,i}(t) = \lambda_i p^{(a)}_{k,i}(t); \quad \dot{d}_{k,i}(t) = x_k(t) k_i \mu_i \qquad (9)$$

$$p^{(a)}_{k,i}(t) = \frac{x_k(t)\exp(\frac{2s_i}{\sqrt{a}}u_k)}{\sum_{\tilde{k}\neq 0:\tilde{k}+e_i\in\mathcal{K}} x_{\tilde{k}}(t)\exp(\frac{2s_i}{\sqrt{a}}u_{\tilde{k}}) + x_0(t)} \qquad (10)$$

$$x_0(t) = \exp(-1-1/a)y(t) \qquad (11)$$

$$y(t) = 1 + \sum_i (y_i(0) - \rho_i)e^{-\mu_i t} \qquad (12)$$

In words, $\dot{a}_{k,i}(t)$ and $\dot{d}_{k,i}(t)$ in (9) are the rates of type i fluid arrival and departure into/from configuration k. Equation (8) is just an accounting identity for configuration k, where the first bracket is the total arrival rate into servers of configuration k and the second bracket is the total departure rate out of configuration k. $p^{(a)}_{k,i}(t)$ in (10) is the fraction of fluid arrivals of type i that are placed in a server of configuration k. $y(t)$ in (12) is the total number of jobs in the system at the fluid limit. The details are omitted due to space constraint.

Algorithm Analysis: Consider the strictly convex function

$$F^{(a)}(x) = \sum_k x_k(1 - ab_k - \sqrt{a}u_k^2) + a\sum_k x_k \log x_k, \qquad (13)$$

where $b_k = -\sum_i \log(k_i!)$ and u_k is the level of configuration k. Note that $\lim_{a\to 0} F^{(a)}(x) = \sum_k x_k = F(x)$. We show that the fluid limit of the system under the asymptotic Best-Fit converges to the optimal solution of the following static optimization

$$\begin{aligned}\min \quad & F^{(a)}(x) \qquad (14)\\ s.t. \quad & \textstyle\sum_k x_k k_i \geq \rho_i; \forall i \qquad (15)\\ & x_k \geq 0; \forall k. \qquad (16)\end{aligned}$$

The Lagrangian is given by

$$L(x,\eta) = F^{(a)}(x) + \sum_i \eta_i(\rho_i - \sum_k x_k k_i)$$

subject to $x_k \geq 0$, for all $k \in \mathcal{K}$, and $\eta_i \geq 0$ for all $1 \leq i \leq n$. Solving for $\partial L/\partial x_k = 0$ yields $x_k = c_k \exp(\frac{1}{a}\sum_i k_i\eta_i)$, where

$$c_k = \exp(-1 - \frac{1}{a} + b_k + \frac{u_k^2}{\sqrt{a}}),$$

and $x_0 := \exp(-1 - 1/a)$. By KKT, a pair (x, η) is the optimal primal-dual solution if

$$\eta_i \geq 0, \ x_k \geq 0, \qquad (17)$$

$$x_k = c_k \exp(\frac{1}{a}\sum_i k_i\eta_i), \qquad (18)$$

$$\sum_k x_k k_i = \rho_i. \qquad (19)$$

It follows from the KKT conditions that the optimal x must satisfy

$$x_{k+e_i}(k_i+1)\mu_i = \lambda_i p^{(a)}_{k,i}, \qquad (20)$$

where

$$p^{(a)}_{k,i} = \frac{x_k(k_i+1)\frac{c_{k+e_i}}{c_k}}{\sum_{\tilde{k}\neq 0:\tilde{k}+e_i\in\mathcal{K}} \frac{c_{\tilde{k}+e_i}}{c_{\tilde{k}}} x_{\tilde{k}}(\tilde{k}_i+1) + \frac{c_{e_i}}{c_0}x_0}. \qquad (21)$$

Notice that $\lambda_i p^{(a)}_{k,i}$ is the rate at which jobs of type i are placed in servers of configuration k (on the fluid scale). Noting that $u_{k+e_i} - u_k = s_i$, and $b_k = -\sum_i \log(k_i!)$, it follows that

$$\frac{c_{k+e_i}}{c_k}(k_i+1) = \exp(\frac{s_i^2}{\sqrt{a}})\exp(\frac{2s_i}{\sqrt{a}}u_k). \qquad (22)$$

Using (22) in (21), the term $\exp(\frac{s_i^2}{\sqrt{a}})$ is independent of configuration and canceled out from numerator and denominator. This is exactly the probability assignment used in the asymptotic Best-Fit algorithm at equilibrium ($t \to \infty$). Notice that the optimal x satisfies (20) and thus is the equilibrium point of the fluid limit equations (8)-(12). This proves that, starting from equilibrium, the asymptotic Best-Fit algorithm minimizes the objective function $F^{(a)}(x(t))$. It remains to show that the fluid limits indeed converge to the equilibrium (20). This follows from the arguments of Stolyar-Zhong [5]. Essentially, $F^{(a)}(t)$ (for small values of a) acts as a Lyapunov function for the system, i.e., if $x(t) \neq x$, then $\frac{\mathrm{d}}{\mathrm{d}t}F^{(a)}(x(t)) < 0$. This shows that $x(t) \to x$ (the optimal solution of the static optimization) as $t \to \infty$. The statement of Theorem 1 then follows from standard arguments for the weak limit of the associated sequence of steady-state random variables and their uniform integrability.

6. REFERENCES

[1] D.S. Johnson, A. Demers, J.D. Ullman, M.R. Garey, R.L. Graham. Worst-case performance bounds for simple one-dimensional packing algorithms. SIAM Journal on Computing vol. 3, no. 4, 1974, pp. 299-325.

[2] C. Kenyon, M. Mitzenmacher, Linear waste of Best Fit bin packing on skewed distributions. Proc. Foundations of Computer Science, 2000.

[3] A.L. Stolyar. An infinite server system with general packing constraints. Operations Research, vol. 61, no. 5, September 2013, pp. 1200-1217.

[4] A.L. Stolyar, Y. Zhong. A large-scale service system with packing constraints: Minimizing the number of occupied servers. SIGMETRICS 2013.

[5] A.L. Stolyar, Y. Zhong. Asymptotic optimality of a greedy randomized algorithm in a large-scale service system with general packing constraints. submitted to Queueing Systems, 2013.

[6] S.T. Maguluri, R Srikant, L. Ying. Stochastic models of load balancing and scheduling in cloud computing clusters. INFOCOM 2012.

[7] V. Gupta, A. Radovanovic. Online stochastic bin packing, `arXiv:1211.2687`.