Randomized Load Balancing with Non-Uniform Task Lengths

Randy Cogill
Department of Systems and Information Engineering
University of Virginia
rcogill@virginia.edu

Abstract

In this paper we consider the classical multiprocessor scheduling problem. In this problem, the goal is to assign a collection of tasks with different execution times to a set of machines so that the total load is balanced as evenly as possible across the machines. This objective is typically expressed as minimization of the *makespan*, or the load on the most heavily loaded machine. In general, the problem of minimizing makespan is known to be NP-hard [2].

Here we consider the scheduling policy that simply assigns tasks to machines uniformly at random, regardless of the task execution times and previous assignments. We show that for any problem instance, the expected makespan under this policy is no greater than a factor of $2\ln(m)/\ln(\ln(m)+1)$ from optimal, where m is the number of machines. This is an extension of existing results that establish similar bounds for the special case of loading m machines with m unit-length tasks [3, 10].

After considering the case of arbitrary task lengths, we then consider the performance of uniform random assignment when task lengths are generated randomly according to an exponential distribution. We show that if the number of tasks exceeds $m\ln(m)$, then the expected makespan under uniform random assignment is no greater than a factor of 4 from optimal. Furthermore, the expected makespan under uniform random assignment approaches the optimal makespan as the number of tasks grows beyond $m\ln(m)$.

1 Introduction

In this paper we consider the classical multiprocessor scheduling problem. This problem is the following. We have n tasks, and m machines on which we can execute these tasks. Task j requires a_j units of execution time. The goal is to assign all of the tasks to machines so that the total time required to execute all tasks is minimized. That is, we would like to partition $\{1,\ldots,n\}$ into m disjoint subsets $S_1,\ldots,S_m$ that minimize the *makespan*,

$$M(S_1,\ldots,S_m)=\max_i\left\{\sum_{j\in S_i}a_j\right\}.$$

Here, S_i gives the set of tasks assigned to machine i, the sum $\sum_{j\in S_i}a_j$ gives the total time required to execute the tasks assigned to machine i, and $M(S_1,\ldots,S_m)$ gives the time at which the last task completes execution.

In general, determining an assignment of tasks that minimizes makespan is NP-hard [2]. Although this is true, there are well known heuristics that can find a schedule with a makespan within a constant factor of the optimal makespan. Probably the most well known heuristic is the *list scheduling* policy, which appeared in the seminal papers by Graham [4, 5]. This policy assigns each task sequentially to the machine with the smallest load at the time of that task's assignment, achieving a makespan within a factor of $2 - 1/m$ of optimal. Although this policy is fairly simple, it still requires knowledge of the assigned tasks' execution times and a centralized scheduler that keeps track of the load assigned to each machine.

Here we consider an even simpler policy that requires no knowledge of task execution times and no knowledge of the states of the individual machines. We consider a scheduling strategy that assigns each task uniformly at random to a machine, independent of the execution time of the task and the assignments of other tasks. Although this strategy uses no information about the set of tasks in each problem instance, we show that the expected makespan under this strategy is surprisingly close to the optimal achievable makespan. In particular, we show that the expected makespan under random assignment, $E[M_\mathrm{r}]$, is within the following bound of the optimal makespan M_opt

$$E[M_\mathrm{r}] \leq \left(\frac{2\ln(m)}{\ln(\ln(m)+1)}\right) M_\mathrm{opt}. \tag{1}$$

In the worst case, an instance can have an assignment with a makespan that is m times larger than the optimal makespan (consider the instance with m unit-size tasks). However, the bound in (1) grows very slowly in m.

We then impose additional structure on the distribution of task lengths, and prove stronger bounds on the relative performance between random assignment and optimal assignment for this case. In particular, we model the task execution times as independent and identically exponentially distributed. For this case, we show that

$$E[M_\mathrm{r}] \leq \left(1 + \sqrt{\frac{m\ln(m)}{n}}\right)^2 E[M_\mathrm{opt}].$$

In particular, if $n \geq m\ln(m)$, then

$$E[M_\mathrm{r}] \leq 4E[M_\mathrm{opt}]$$

and

$$\frac{E[M_\mathrm{r}]}{E[M_\mathrm{opt}]} \to 1$$

as $n \to \infty$. In other words, if task lengths are identically exponentially distributed, assigning tasks to machines at random is asymptotically optimal in the number of tasks.

When all tasks are of uniform size, the makespan under uniform random assignment has been studied previously [10]. The paper [3] discusses the problem of randomly assigning m tasks to m machines, although it is presented in the context of hash code searching. The problem of randomly assigning uniform size tasks is also often associated with classical occupancy problems and urn models [6]. For uniform size tasks, there are known results stating that when assigning m tasks to m machines, expected makespan grows as

$$E[M_\mathrm{r}] = \frac{\ln(m)}{\ln(\ln(m)+1)}(1+o(1)),$$

m	10	100	1000	10000	100000
$E[M_{\mathrm{r}}]/M_{\mathrm{opt}}$	3.85	5.34	6.68	7.93	9.11

Table 1: Values of (1) for various numbers of machines.

and when assigning $n = m\ln(m)$ tasks to m machines, expected makespan grows as $E[M_{\mathrm{r}}] = O(n/m)$ (see, for example, [10]). In this paper we will provide extensions of these results to cases with non-uniform task lengths.

For arbitrary non-uniform task lengths, analysis of random assignment schemes becomes considerably more difficult. To the best of our knowledge, this paper contains the first analysis of this case. The main tool that enables the analysis of this case is the Chernoff-type bound in Lemma 3 of the Appendix. This result appears to be new as well.

For the case of non-uniform task lengths generated by a given probability distribution, there are a number of results dealing with average performance of heuristics such as list scheduling [1, 8]. A number of similar results establish various forms of asymptotic optimality of simple scheduling heuristics [7]. However, it appears that there are no existing results establishing asymptotic optimality of simple random assignment when task lengths are non-uniform.

2 Arbitrary Task Lengths

For the multiprocessor scheduling problem, we consider a simple algorithm which we call *uniform random assignment*. This algorithm simply assigns each task to a machine uniformly at random, independent of the task's execution time and the assignments of other tasks. In other words, this algorithm constructs the sets $R_1, \ldots, R_m$ by randomly assigning each element of $\{1, \ldots, n\}$ to a set R_i. The makespan under this algorithm is then described by the random variable

$$\begin{aligned} M_{\mathrm{r}} &= M(R_1, \ldots, R_m) \\ &= \max_i \left\{ \sum_{j \in R_i} a_j \right\}. \end{aligned}$$

Also, we denote the optimal achievable makespan for a problem instance by

$$\begin{aligned} M_{\mathrm{opt}} &= \min_{S_1, \ldots, S_m} \{M(S_1, \ldots, S_m)\} \\ &= \min_{S_1, \ldots, S_m} \left\{ \max_i \left\{ \sum_{j \in S_i} a_j \right\} \right\}. \end{aligned}$$

The main result of this section is the following theorem, which relates the expected value of M_{r} to the optimal makespan M_{opt}.

Theorem 1. *For $m \geq 2$, the makespan under uniform random assignment, M_r, satisfies*

$$E[M_r] \leq \left(\frac{2\ln(m)}{\ln(\ln(m) + 1)} \right) M_{opt}$$

for any problem instance.

Proof. Let W_i be the workload assigned to machine i under uniform random assignment. That is,

$$W_i = \sum_{j \in R_i} a_j$$

where each W_i is identically distributed. So for any $t > 0$,

$$\begin{aligned} E[M_{\mathrm{r}}] &= E[\max\{W_1, \ldots, W_m\}] \\ &\leq \frac{1}{t}\big(\ln(m) + \ln(E[e^{tW_1}])\big), \end{aligned}$$

where the inequality follows from Lemma 3. Let $W_1 = a_1 I_1 + \cdots + a_n I_n$, where

$$I_j = \begin{cases} 1 & \text{if task } j \text{ is assigned to machine 1} \\ 0 & \text{otherwise.} \end{cases}$$

Since assignments of tasks to machine 1 are independent,

$$\begin{aligned} E[e^{tW_1}] &= E[e^{ta_1 I_1 + \cdots + ta_n I_n}] \\ &= \prod_{j=1}^{n} E[e^{ta_j I_j}] \\ &= \prod_{j=1}^{n} \left(1 + \frac{e^{ta_j} - 1}{m}\right). \end{aligned}$$

This gives

$$\begin{aligned} \ln(E[e^{tW_1}]) &= \sum_{j=1}^{n} \ln\left(1 + \frac{e^{ta_j} - 1}{m}\right) \\ &= \frac{1}{m} \sum_{j=1}^{n} \ln\left(\left(1 + \frac{e^{ta_j} - 1}{m}\right)^m\right). \end{aligned}$$

By Lemma 4

$$\left(1 + \frac{e^{ta_j} - 1}{m}\right)^m \leq \exp(e^{ta_j} - 1).$$

This gives

$$\begin{aligned} \ln(E[e^{tW_1}]) &\leq \frac{1}{m} \sum_{j=1}^{n} (e^{ta_j} - 1) \\ &\leq \left(\frac{e^{t\max\{a_j\}} - 1}{\max\{a_j\}}\right) \frac{1}{m} \sum_{j=1}^{n} a_j, \end{aligned}$$

where the second inequality follows from convexity of the exponential. Setting

$$t = \frac{\ln(\ln(m) + 1)}{\max\{a_j\}}$$

gives

$$\begin{aligned} E[M_{\mathrm{r}}] &\leq \frac{\max\{a_j\}}{\ln(\ln(m)+1)}\left(\ln(m)+\left(\frac{\ln(m)}{\max\{a_j\}}\right)\frac{1}{m}\sum_{j=1}^{n} a_j\right) \\ &= \left(\max\{a_j\}+\frac{1}{m}\sum_{j=1}^{n} a_j\right)\frac{\ln(m)}{\ln(\ln(m)+1)}. \end{aligned}$$

Since both $\max\{a_j\} \leq M_{\mathrm{opt}}$ and $\frac{1}{m}\sum_{j=1}^{n} a_j \leq M_{\mathrm{opt}}$ by Lemma 5, we have

$$E[M_{\mathrm{r}}] \leq \left(\frac{2\ln(m)}{\ln(\ln(m)+1)}\right) M_{\mathrm{opt}}.$$

■

We now turn to the question of tightness of the bound in (1). As discussed in the Introduction, similar results have been established for the case of m unit-sized tasks assigned randomly to m machines. For this case, Gonnet [3] has shown that the expected load on the most heavily loaded machine is

$$\begin{aligned} E[M_{\mathrm{r}}] &= \Gamma^{-1}(m)\left(1+O\left(\frac{1}{\ln(\Gamma^{-1}(m))}\right)\right) \\ &= \frac{\ln(m)}{\ln(\ln(m)+1)}(1+o(1)). \end{aligned}$$

For this case, the assignment that minimizes makespan simply assigns one task to each machine, achieving a makespan of 1. Therefore, for this simple special case, the exact ratio between the expected makespan under uniform random assignment and the optimal makespan is within a small constant factor of the bound in (1).

3 Exponentially Distributed Task Lengths

In the previous section, we considered the ratio between the optimal makespan and the makespan under uniform random assignment. We gave a bound on this ratio that holds for arbitrary task lengths. If we make stronger assumptions on the distribution of task lengths, we can prove some stronger results about the performance of uniform random assignment. Specifically, if we model task lengths as being distributed according to some given probability distribution, we can perform an average-case analysis which will be tighter than the worst-case analysis of the previous section.

Here we will model task lengths as being independent and identically exponentially distributed. Assuming this model for task lengths, the main result of this section states that, on average, the makespan under uniform random assignment approaches the optimal makespan as the number of tasks grows.

Theorem 2. *Consider instances where task lengths are drawn independently from an exponential distribution with parameter λ. Over all instances with m machines and n tasks, the makespan under uniform random assignment satisfies*

$$E[M_r] \leq \left(1+\sqrt{\frac{m\ln(m)}{n}}\right)^2 E[M_{opt}].$$

In particular, if $n \geq m \ln(m)$, then

$$E[M_r] \leq 4E[M_{opt}]$$

and

$$\frac{E[M_r]}{E[M_{opt}]} \to 1$$

as $n \to \infty$.

Proof. Let W_i be the workload assigned to machine i under uniform random assignment. That is,

$$W_i = \sum_{j \in R_i} A_j$$

where each A_j is independent and exponentially distributed with parameter λ. Also note that each W_i is identically distributed as well. From Lemma 3, for any $t > 0$,

$$\begin{aligned} E[M_\mathrm{r}] &= E[\max\{W_1, \ldots, W_m\}] \\ &\leq \frac{1}{t}\big(\ln(m) + \ln(E[e^{tW_1}])\big), \end{aligned}$$

Let $W_1 = A_1 I_1 + \cdots + A_n I_n$, where

$$I_j = \begin{cases} 1 & \text{if task } j \text{ is assigned to machine 1} \\ 0 & \text{otherwise.} \end{cases}$$

Then

$$E[e^{tW_1}] = E_A\left[E_I[e^{tA_1I_1 + \cdots + tA_nI_n} | A_1, \ldots, A_n]\right].$$

Since assignments of tasks to machine 1 are independent,

$$\begin{aligned} E_I[e^{tA_1I_1 + \cdots + tA_nI_n} | A_1, \ldots, A_n] &= \prod_{j=1}^{n} E_I[e^{tA_jI_j} | A_j] \\ &= \prod_{j=1}^{n} \left(1 + \frac{1}{m}(e^{tA_j} - 1)\right). \end{aligned}$$

Also, since the task lengths are independent and exponentially distributed,

$$\begin{aligned} E_A\left[\prod_{j=1}^{n} \left(1 + \frac{1}{m}(e^{tA_j} - 1)\right)\right] &= \prod_{j=1}^{n} \left(1 + \frac{1}{m}(E_A[e^{tA_j}] - 1)\right) \\ &= \prod_{j=1}^{n} \left(1 + \frac{1}{m}\left(\frac{t}{\lambda - t}\right)\right), \end{aligned}$$

where we have used the fact that the moment generating function of the exponential random variable A_j is

$$E[e^{tA_j}] = \frac{\lambda}{\lambda - t}.$$

This gives

$$\begin{aligned}\ln(E[e^{tW_1}]) &= n\ln\left(1+\frac{1}{m}\left(\frac{t}{\lambda-t}\right)\right)\\ &= \frac{n}{m}\ln\left(\left(1+\frac{1}{m}\left(\frac{t}{\lambda-t}\right)\right)^m\right).\end{aligned}$$

By Lemma 4

$$\left(1+\frac{1}{m}\left(\frac{t}{\lambda-t}\right)\right)^m \leq \exp\left(\frac{t}{\lambda-t}\right),$$

giving

$$\ln(E[e^{tW_1}]) \leq \frac{n}{m}\left(\frac{t}{\lambda-t}\right).$$

Setting

$$t = \frac{\lambda\sqrt{m\ln(m)}}{\sqrt{n}+\sqrt{m\ln(m)}}$$

gives

$$\begin{aligned}E[M_{\mathrm{r}}] &\leq \frac{1}{t}\ln(m) + \frac{n}{m}\left(\frac{1}{\lambda-t}\right)\\ &= \frac{1}{\lambda}\frac{n}{m}\left(1+\sqrt{\frac{m\ln(m)}{n}}\right)^2.\end{aligned}$$

For any realization of $A_1,\ldots,A_n$, the optimal makespan satisfies $M_{\mathrm{opt}} \geq \frac{1}{m}\sum_{j=1}^{n} A_j$ by Lemma 5. Therefore, over all instances with n tasks and m machines,

$$\begin{aligned}E[M_{\mathrm{opt}}] &\geq E\left[\frac{1}{m}\sum_{j=1}^{n} A_j\right]\\ &= \frac{1}{\lambda}\frac{n}{m}\end{aligned}$$

and

$$E[M_{\mathrm{r}}] \leq \left(1+\sqrt{\frac{m\ln(m)}{n}}\right)^2 E[M_{\mathrm{opt}}].$$

Finally, it is clear that for $n \geq m\ln(m)$

$$\left(1+\sqrt{\frac{m\ln(m)}{n}}\right)^2 \leq 4,$$

and

$$\lim_{n\to\infty}\left(1+\sqrt{\frac{m\ln(m)}{n}}\right)^2 = 1.$$

■

4 Conclusions

In this paper we considered the classical multiprocessor scheduling problem. We considered a scheduling policy that assigns tasks to machines at random and proved several bounds on the ratio between the expected makespan under this policy and the optimal achievable makespan. In particular, we show that some known results for the the case of tasks of uniform length essentially carry over to cases with tasks of different lengths. This is surprising for several reasons. First, optimal assignment of tasks of uniform length is straightforward, but optimal assignment of tasks with different lengths is NP-hard [2]. Therefore, although the level of complexity of the problem of optimal assignment is very different for these two cases, a very simple algorithm that doesn't distinguish between these two cases works almost equally well for both. Second, for uniform task lengths it is optimal to distribute tasks evenly between machines, and it is clear that assigning tasks to machines uniformly at random will achieve a fairly even distribution of the total load across the machines. However, when the total load is distributed unevenly among the tasks, makespan can be sensitive to the relative placement of certain tasks. For example, if a small number of tasks have lengths that are much greater than the lengths of the remaining tasks, to minimize makespan it is crucial that these tasks be placed on separate machines, and that the remaining tasks are distributed among the remaining machines. However, uniform random assignment treats all tasks equally, so it is not clear that this should lead to an even distribution of the total load.

As an additional contribution, the main proofs of this paper use what appears to be a new technique for bounding the expected value of the maximum of a collection of correlated random variables. We believe that this technique has applications well beyond the problem discussed in this paper.

References

[1] J. Bruno and P. Downey. Probabilistic bounds on the performance of list scheduling. *SIAM Journal on Computing*, 15(2):409–417, 1986.

[2] M. Garey and D. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979.

[3] G.H. Gonnet. Expected length of the longest probe sequence in hash code searching. *Journal of the ACM*, 28(2):289–304, 1981.

[4] R.L. Graham. Bounds for certain multiprocessing anomalies. *Bell System Technical Journal*, 45(9):1563–1581, 1966.

[5] R.L. Graham. Bounds on multiprocessing timing anomalies. *SIAM Journal on Applied Math.*, 17(2):416–429, 1969.

[6] N. Johnson and S. Kotz. *Urn Models and Their Applications*. John Wiley & Sons, 1977.

[7] E. Coffman Jr. and W. Whitt. Recent asymptotic results in the probabilistic analysis of makespan scheduling.

[8] E.G. Coffman Jr. and G.S. Lueker. *Probabilistic Analysis of Packing and Partitioning Algorithms*. John Wiley & Sons, 1991.

[9] M. Mitzenmacher. *The Power of Two Choices in Randomized Load Balancing.* Ph.D. thesis, University of California, Berkeley, 1996.

[10] M. Raab and A. Steger. "Balls into bins" — A simple and tight analysis. *Lecture Notes in Computer Science*, 1518:159–170, 1998.

Appendix

Proofs of the main theorems in this paper are based on the following Chernoff-type bound. This gives an upper bound on the expected value of the maximum of a set of dependent random variables.

Lemma 3. *Suppose $X_1, \ldots, X_m$ are identically distributed, but not necessarily independent random variables. Then for any $t > 0$,*

$$E[\max\{X_1, \ldots, X_m\}] \leq \frac{1}{t}\left(\ln(m) + \ln\left(E[e^{tX_1}]\right)\right).$$

Proof. For any $x_1, \ldots, x_m$ and $t > 0$,

$$\begin{aligned} \max\{x_1, \ldots, x_m\} &= \frac{1}{t}\ln\left(\max\{e^{tx_1}, \ldots, e^{tx_m}\}\right) \\ &\leq \frac{1}{t}\ln\left(e^{tx_1} + \cdots + e^{tx_m}\right). \end{aligned}$$

Therefore, for the random variables $X_1, \ldots, X_m$ we have

$$\begin{aligned} E[\max\{X_1, \ldots, X_m\}] &\leq \frac{1}{t}E\left[\ln\left(e^{tX_1} + \cdots + e^{tX_m}\right)\right] \\ &\leq \frac{1}{t}\ln\left(E[e^{tX_1}] + \cdots + E[e^{tX_m}]\right) \\ &= \frac{1}{t}\ln\left(mE[e^{tX_1}]\right) \\ &= \frac{1}{t}\left(\ln(m) + \ln\left(E[e^{tX_1}]\right)\right), \end{aligned}$$

where the second inequality follows from Jensen's inequality. ∎

The following approximation to the exponential function is very well known, but is presented here to keep our treatment self contained.

Lemma 4. *For $x \geq 0$ and any positive integer m,*

$$\left(1 + \frac{x}{m}\right)^m \leq e^x.$$

Proof.

$$
\begin{aligned}
\left(1+\frac{x}{m}\right)^m &= \sum_{i=0}^{m}\binom{m}{i}\left(\frac{x}{m}\right)^i \\
&= 1+\sum_{i=1}^{m}\left(\frac{\prod_{j=1}^{i}(m-j+1)}{m^i}\right)\frac{x^i}{i!} \\
&\leq 1+\sum_{i=1}^{m}\frac{x^i}{i!} \\
&\leq 1+\sum_{i=1}^{\infty}\frac{x^i}{i!} \\
&= e^x
\end{aligned}
$$

■

The following lower bounds for optimal makespan are well known, and are found in the original papers on multiprocessor scheduling by Graham [4, 5].

Lemma 5. *Let M_{opt} be the minimum achievable makespan over all schedules. Then we have the lower bounds*

$$M_{opt} \geq \max\{a_1, \ldots, a_n\}$$

and

$$M_{opt} \geq \frac{1}{m}\sum_{j=1}^{n} a_j.$$

Proof. Let W_i be the workload of machine i under an optimal schedule. Also, let $a_k = \max\{a_1, \ldots, a_n\}$. Under an optimal schedule, suppose task k is assigned to machine p. Then

$$
\begin{aligned}
a_k &\leq W_p \\
&\leq M_{\text{opt}}.
\end{aligned}
$$

To prove the second inequality, note that under any schedule the maximum workload over all machines is greater than its average. That is,

$$M_{\text{opt}} \geq \frac{1}{m}\sum_{i=1}^{m} W_i.$$

Since each task is assigned to one machine, $W_1 + \cdots + W_m = a_1 + \cdots + a_n$, giving

$$M_{\text{opt}} \geq \frac{1}{m}\sum_{j=1}^{n} a_j.$$

■