Application with Dual Approach

May 15, 2017

1 Preliminaries and Problem Definition

Consider a parallel computing cluster which consists of M worker nodes and each worker node has a single slot, namely, it can only hold one task at any time. We apply the FIFO scheduler to schedule the jobs that come into the cluster, i.e., each time, the job with the highest priority is scheduled if there are available machines, other jobs which have the lower priority should wait in the queue. For simplicity, we assume this cluster is homogeneous in the sense that all the worker nodes are identical. For ease of description, we often interchange the two notations through this paper, i.e., machine and slot.

In this cluster, we assume t_i be the duration of task i (the time between the task is initialized and the task is finished) , follows a heavy-tail distribution, i.e., $t_i \sim F(\lambda_i, t)$, where F is a specific distribution and λ_i is a parameter related to the input data size for task i. Each job consists of several tasks which can run parallel on multiple machines.

Definition 1. *The flowtime of a job J is $flow(J) = f(J) - a(J)$, where $f(J)$ and $a(J)$ denote the finish (completion) time and arrive time respectively.*

If a task i runs t_i time on a worker node, then it will consume $c * t_i$ amount of resource on this node where c is a constant number. Actually, there exists a startup time for each task. And for simplicity, we do not consider it. Without loss of generality, consider a particular job J consisting of N tasks which is submitted by a particular user to the cluster.

In the next section, we will introduce our models and present the analysis under two different regimes: i) The cluster is lightly loaded, i.e., the total number of slots is larger than the number of tasks in job J. ii) The cluster is heavily loaded, i.e., the total number of slots is smaller than the number of tasks in job J. The intuition of distinguishing between the traffic load in the cluster is that the space for making speculative copies differs a lot under different traffic loading. In the lightly load case, the scheduler can make a large number of speculative copies so as to achieve a small *flowtime* for job J. While in the heavily loaded cluster, there is only a small number of available machines and the scheduler needs to make a tradeoff between scheduling duplicates and assigning new tasks at most time. Assigning too many duplicates can easily block the scheduling of other tasks and may make the *flowtime* of job J even longer.

2 System optimization when the cluster is lightly loaded

When the cluster is lightly loaded, i.e $M > N$, all the tasks of Job J can be launched simultaneously in the cluster while some machines still remain idle. In this case, we can choose to make more copies for the tasks running in the cluster. Moreover, each task must at least maintain one copy in the cluster. Assume task i runs r_i copies and the jth copy has a duration with x_i^j who follows the same distribution as x_i. Further, we assume that all the x_i^j are i.i.d random variables for $1 \leq j \leq r_i$. In this model, we impose the constraint that job can be finished within T_d with a probability at least $1-\epsilon$. The parameters T_d and ϵ can be defined either by the user or the service provider of the cluster.

We begin to make clones for tasks at the same time as job J begins to run in the cluster. Under such a scheme, there is no need to monitor the progress of each task any more. The objective is to minimize the total resource consumption for job J while satisfying the deadline constraint with a high probability. We formulate this problem (P1) as a chance constraint problem in the following:

$$\begin{aligned} \min_{r_i} \quad & \mathbb{E}\left[\sum_{i=1}^{N} r_i \cdot c \cdot t_i\right] && \text{(1a)} \\ s.t. \quad & \sum_{i=1}^{N} r_i \leq M && \text{(1b)} \\ & t_i = \min\{x_i^1, x_i^2, \dots, x_i^{r_i}\} \qquad \forall i && \text{(1c)} \\ & r_i \geq 1 \qquad \forall i && \text{(1d)} \\ & Pr[\max\{t_1, t_2, \dots, t_N\} > T_d] \leq \epsilon && \text{(1e)} \end{aligned}$$

In the formulation, T_d is the deadline defined by the user. The first constraint states that the total number of copies of job J is no more than M and the third constraint states that as soon as one copy of task i finishes, the task completes. The last constraint states that $flow(J) = max\{t_1, t_2, \dots, t_N\}$ should not hit the deadline T_d with a certain probability $1-\epsilon$. We term this optimization problem as CSD.

2.1 Characterizing the optima of CSD

In this section, we show the convexity of Formulation 1 in Section 1 and provide characterizations of the optimal solutions to CSD.

Define $F_i(x) = F(\lambda_i, x) = Pr\{x_i < x\}$ and $H_i(t)$ as the distribution of t_i. Then, $H_i(t)$ can be expressed as

$$H_i(t) = 1 - (1 - F_i(t))^{r_i} \tag{2}$$

and the probability of not hitting the deadline is

$$Pr[\max\{t_1, t_2, \dots, t_N\} > T_d] = 1 - \prod_{i=1}^{N} H_i(T_d). \tag{3}$$

Further, the expected duration for each task is

$$\mathbb{E}[t_i] = \int_0^\infty t \cdot d(H_i(t)) = \int_0^\infty (1 - H_i(t))dt. \tag{4}$$

Thus, the expected resource consumption for all the tasks is

$$\mathbb{E}\left[\sum_{i=1}^{N} r_i \cdot c \cdot t_i\right] = c \cdot \sum_{i=1}^{N} \underbrace{r_i \int_0^\infty (1 - F_i(t))^{r_i} dt}_{\triangleq g_i(r_i)} \tag{5}$$

Lemma 1. *$r_i \cdot \int_0^\infty (1 - F_i(t))^{r_i} dt$ is a strictly convex function of r_i on the condition that $\ln(1 - F_i(t))$ is strictly convex.*

In reality, making more clones will definitely incur a larger amount of resource consumption as discussed in [1]. Thus, we assume $g_i(r_i)$ is a strictly increasing function of r_i and $g_i(\infty) = \infty$. Moreover, the last constraint implies that

$$\sum_{i=1}^{N} \ln\left(H_i(T_d)\right) \geq \ln\left(1-\epsilon\right). \tag{6}$$

Applying the inequality that $\ln(1-x) \leq -x$ when $x \in [0,1)$, we can relax the last constraint as follows:

$$\sum_{i=1}^{N} (1 - F_i(T_d))^{r_i} \leq \ln\left(\frac{1}{1-\epsilon}\right). \tag{7}$$

For ease of presentation, let $h_i(r_i)$ denote $(1 - F_i(T_d))^{r_i}$ for all i and C denote $\ln\left(\frac{1}{1-\epsilon}\right)$. Observe that $h_i(r_i)$ is a convex function of r_i and all the constraints constraints in Formulation (1) are linear. If the condition in Lemma 1 is positive, the objective in Formulation (1) is also convex. Thus, we can adopt the convex optimization technique to solve CSD.

c is a constant and it can be treated as 1. Thus, we derive the dual problem of (1) and define the Lagrangian

$$\begin{aligned} L(\mathbf{w}, l, k, \mathbf{r}) = &\sum_{i=1}^{N} g_i(r_i) + \sum_{i=1}^{N} w_i(r_i - 1) + l \cdot \left(\sum_{i=1}^{N} r_i - M\right) \\ &+ k \cdot \left(\sum_{i=1}^{N} h_i(r_i) - C\right) \end{aligned}$$

here, $\mathbf{w} = [w_1, w_2, \cdots, w_N]$, l and k are non-negative Lagrangian multipliers and $\mathbf{r} = [r_1, r_2, \cdots, r_N]$. We assume $\mathbf{r} = \mathbf{e}$ is not a feasible solution to CSD where $\mathbf{e}$ is a full one vector. Otherwise $\mathbf{e}$ is just the optimal solution and the optimization problem has been solved.

The KKT conditions of stationarity, primal and dual feasibility and complementary slackness are:

$$g_i'(r_i) + w_i + l + k h_i'(r_i) = 0 \tag{8}$$

$$w_i \cdot (r_i - 1) = 0; \quad w_i \geq 0, \quad r_i \geq 1 \tag{9}$$

$$l \cdot \left(\sum_{i=1}^{N} r_i - M\right) = 0; \quad l \geq 0, \quad \sum_{i=1}^{N} r_i \leq M \tag{10}$$

$$k \cdot \left(\sum_{i=1}^{N} h_i(r_i) - C\right) = 0; \quad k \geq 0, \quad \sum_{i=1}^{N} h_i(r_i) \leq C \tag{11}$$

$\sum_{i=1}^{N} g_i(r_i)$ is a strictly convex function of $\mathbf{r}$ for all $\mathbf{r} \geq \mathbf{0}$ due to the convexity of $g_i(r_i)$. Thus, the CSD problem has an unique optimal solution and denote it by $\mathbf{r}^*$.

Theorem 1. *When the constraints (1b) & (1d) are inactive and $F_i(t)$ is a pareto distribution for all i, the optimal solution for CSD is determined by the following equation.*

$$r_i^* = max\left\{\left[\frac{1}{\alpha} \cdot log_{\nu_i}\left(\frac{\ln \nu_i \cdot \sum_{i=1}^{N} \frac{1}{\ln \nu_i}}{\ln \frac{1}{1-\epsilon}}\right)\right], 1\right\} \tag{12}$$

where α is the heavy-tail order, μ_i is the mean of the task duration and $\nu_i = \frac{\alpha T_d}{(\alpha-1)\mu_i}$.

Proof. Let w_i and l be zero in the KKT equations and the result immediately follows. □

3 Dual Algorithm Design

Based on this Theorem 1, we can apply r^* in Equation (12) to test whether the constraints (1b) & (1c) can satisfy. If the answer is positive, then r_i^* must be the optimal number of copies each task should make. Otherwise, we adopt the dual based subgradient projection algorithm to solve CSD iteratively.

We rewrite the Lagrangian function and let the constraint (1d) remain in the dual problem.

$$L_1(l,k,\mathbf{r}) = \sum_{i=1}^{N} g_i(r_i) + l(\sum_{i=1}^{N} r_i - M) + k(\sum_{i=1}^{N} h_i(r_i) - C)$$

So the dual problem is

$$(D1) \qquad \max_{l\geq 0, k\geq 0,} D(l,k) \tag{13}$$

where the dual objective function is given as

$$\begin{aligned} &D(l,k) = \min L_1(l,k,\mathbf{r}) \\ &s.t. \quad r_i \geq 1 \quad \forall i \end{aligned} \tag{14}$$

Following the analysis in Section 2.1, the CSD problem is convex and thus the duality gap is zero. It's known [2,3] that the subgradient of $D(l,k)$ at (l,k) are

$$\nabla D(l,k) = (\sum_{i=1}^{N} r_i(l,k) - M, \sum_{i=1}^{N} h_i(r_i(l,k)) - C) \tag{15}$$

where $\mathbf{r}(l,k) = [r_1(l,k), r_2(l,k), \cdots, r_N(l,k)]$ is defined by

$$r_i(l,k) = \operatorname*{argmin}_{r_i\geq 1} g_i(r_i) + lr_i + kh_i(r_i) \tag{16}$$

Hence, $r_i(l,k) = [r_i^p]_1^+$ [1] and r_i^p is determined by

$$g_i'(r_i) + l + kh_i'(r_i) = 0 \tag{17}$$

Applying the stand subgradient projection method to the dual problem yields the following updating algorithm. Moreover, the following theorem guarantees the convergence of Algorithm 1

Algorithm 1: The dual-based subgradient projection Algorithm

```
1 Starting from a positive $l(0)$ and non-negative $k(0)$, the number of copies for each task is
  updated as
      $r_i(t) := \text{argmin}_{r_i\geq 1}\{g_i(r_i) + l(t)r_i + k(t)h_i(r_i)\}$
      $l(t+1) := [l(t) + \gamma_t(\sum_{i=1}^{N} r_i(t) - M)]^+$
      $k(t+1) := [k(t) + \gamma_t(\sum_{i=1}^{N} h_i(r_i(t)) - C]^+$
  where $\{r_i, l, k\}(t)$ denote the values of the corresponding parameters during the $t$th iteration
  of the algorithm and $\gamma_t$ is the positive stepsize.
```

for appropriate γ_t.

[1]The notation $[x]_1^+$ and $[x]^+$ stand for $\max\{x,1\}$ and $\max\{x,0\}$ respectively.

Theorem 2. *If we choose γ_t such that $\sum_{t=0}^{\infty} \gamma_t = \infty$, $\sum_{t=0}^{\infty} \gamma_t^2 < \infty$, then $\{\mathbf{r}(t)\}$ converges to the optimal solution of the CSD problem.*

Proof. By proposition 8.2.6 of [3], the sequences $\{l(t), k(t)\}$ generated by Algorithm 1 converge to some optimal solution of the dual problem. Then the sequence $\{\mathbf{r}(t)\}$ determined by Equation (16) must also converge to the unique optimal solution. Thus the proof completes. $\square$

References

[1] G. Ananthanarayanan, M. C.-C. Hung, X. Ren, I. Stoica, and A. Wierman. Grass: Trimming stragglers in approximation analytics. In *Proceedings of NSDI*, April 2014.

[2] D. Bertsekas and J. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods.* Prentice-Hall, 1989.

[3] D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar. *Convex Analysis and Optimization.* Athena Scientific, 2003.