Hummer: Mitigating Stragglers with Partial Clones

Jia LI, Changjian WANG, Dongsheng LI, Yiming ZHANG

National Laboratory for Parallel and Distributed Processing, School of Computer Science,
National University of Defense Technology, Changsha, China
{josephlijia, lds1201}@163.com, c_j_wang@yeah.net,
sdiris@gmail.com

Abstract. Small jobs typically run for interactive data analyses in datacenters, and often are delayed by long-running tasks called stragglers. Many efforts, like Blacklist, speculative execution, proactive mitigation, have been devoted to the solutions. However, they either consume too much time or waste too many resources. In this paper, we propose a new proactive method to mitigate stragglers by performing partial clones, which improves job average duration by 48% and 18% compared to LATE and Dolly.

Keywords: small jobs, stragglers, partial clones

1 Introduction

Straggler mitigation [3] is vital to data analyses since it can decrease data analyses performance obviously. Lots of efforts have been devoted to the solution of stragglers and some feasible approaches have been proposed, such as Blacklist, speculative execution [1, 4], Dolly [2].

For Blacklist, failed or slow nodes will be put into blacklist and no new tasks will be assigned to them. Speculative execution launches multiple copies for slow tasks and selects the earliest one. Dolly clones each task in small jobs to mitigate stragglers. Though these above-mentioned approaches are efficient for straggler mitigation, they waste too much time or too many resources.

In this paper, we propose a new approach for straggler mitigation, named Hummer. In Hummer, only the tasks with high delaying risk in a job will be cloned, which can save lots of cluster resources and can reduce job average duration obviously.

2 Design of Hummer

Due to the shortcomings of LATE [1] and Dolly [2], we propose a trade-off approach Hummer. Hummer launches clones for part of tasks in one job to mitigate stragglers proactively, just when they are submitted. Hummer picks up the earliest clone as the result. The default number of clones for one task is three, which has been proved to be the best value. Because there exist differences among nodes in cluster, one task dispatched to high risk node (stragglers appear more often) turns out to be a straggler

with high probability. Thus, in Hummer, we maintain the risk of straggler for all nodes in our cluster according to historical data. Hummer eliminates the waiting time in speculative execution and saves more resources than Dolly, meanwhile improving job average duration to a large extent. Fig.1 shows the process of partial clones.

3 Experimental Evaluation

In our experiment, we find small jobs (with 0-100 tasks) occupy almost 80% cloning resources consumption in cluster. From the results as Fig.2 shows, we can see that Hummer is more suitable for small jobs. Hummer reduces job average completion time by 46% and 18% compared to LATE and Dolly. We bin the jobs as Table 1.

Table 1. Job bins by the number of tasks

Bin	1	2	3	4
Tasks	0-50	51-100	101-200	201-500

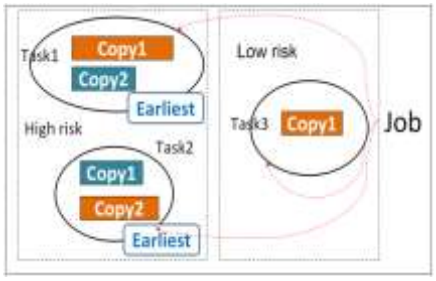

Fig. 1. The process of partial clones

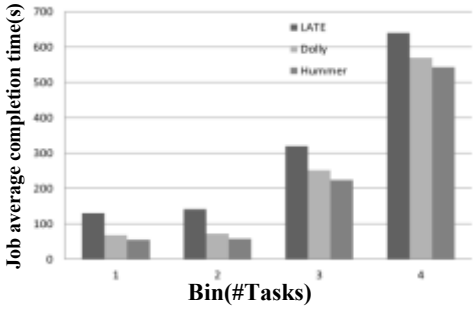

Fig. 2. Job Average Duration in Hummer

4 Conclusions

This paper presents a proactive approach to mitigate stragglers, called Hummer. Experimental results show that Hummer reduces job average duration by almost 46% and 18% compared to LATE and Dolly for small jobs.

References

1. M. Zaharia, A. Konwinski, A. D. Joseph, R. Katz, I. Stoica. Improving MapReduce Performance in Heterogeneous Environments. In Proc. of the USENIX OSDI, 2008.
2. G. Ananthanarayanan, A. Ghodsi, S. Shenker, and I. Stoica. Effective Straggler Mitigation: Attack of the Clones. In Proc. of the USENIX NSDI, 2013.
3. Y. Kwon, M. Balazinska, B. Howe, and J. Rolia. SkewTune: Mitigating skew in MapReduce applications. In Proc. of SIGMOD
4. G. Ananthanarayanan, S. Kandula, A. Greenberg, I. Stoica, E. Harris, and B. Saha. Reining in the Outliers in Map-Reduce Clusters using Mantri. In Proc. of OSDI, 2010.