# AUTO PROVISIONING FOR LONG-RUNNING MAPREDUCE APPLICATIONS ON AWS EMR

Users can easily provision a variety of small to large scale cloud resources in order to run analytics jobs on the cloud. To ease adoption of popular programming models, cloud providers are offering easy to configure platforms for non-expert users. Such platforms include Amazon Web Services (AWS) Elastic Map Reduce (EMR) service. EMR users can upload their MapReduce code and data set, however, they also have to configure the cluster size and type of virtualized resources to run their application on. Without having deep knowledge into the capabilities of the cloud resources or the requirements of their applications, EMR users typically end up over-provisioning or under-provisioning. Over-provisioning leads to increased expenditure while under-provisioning leads to prolonged execution time.

Public cloud platforms provide several options to select virtual machine (VM) instance types from. To launch an EMR cluster on AWS, there are fifteen EC2 instance types to choose from. Each EC2 instance type varies in its processor architecture, number of virtual CPUs, memory, and I/O bandwidth. Specifically, the AWS instance types are divided into six major categories (general purpose, compute optimized, GPU instances, memory optimized, storage optimized and micro instances). Due to the variety of MR application types, provisioning the wrong resource type and size could have adverse effects on performance and cost.

In EMR, the number of mapper and reducer slots per instance is dependent on the type of instance provisioned. The number of map and reduce tasks that are run also depends on the size and number of files of input data. The performance of and the cost incurred by a MapReduce job depends on four parameters: type of instance, number of instances, application type and size of the data set. This is especially the case for long-running MapReduce applications where the cluster setup time does not dominate overall execution time. Different users run different types of MapReduce jobs and have different budget and deadline constraints. Auto-provisioning resources for different types of MapReduce applications while varying the input data set size, instance types and cluster size is not a trivial task. However, the question of selecting suitable resources when launching a MapReduce cluster is an important one.

For this project, we will limit our scope to three application types, three input data set sizes and three instance types. The task is to automatically provision resources on EMR to meet a user’s specified time and budget constraints. Students will be expected to profile representative applications into classes, classify new applications, perform load and scale testing, provision resources based on user specs, and measure the automatic provisioning system’s effectiveness at meeting its objective.

## Related Material

[1] H. Herodotou, et al, “No One (Cluster) Size Fits All: Automatic Cluster Sizing for Data-intensive Analytics”, SOCC '11 Proceedings of the 2nd ACM Symposium on Cloud Computing, 2011

[2] F. Zhang and M. Sakr, “Dataset Scaling and MapReduce Performance”, Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2013 IEEE 27th International, May 2013

[3] Z. Ren, et al, “Workload Characterization on a Production Hadoop Cluster: A Case Study on Taobao”, Workload Characterization (IISWC), 2012 IEEE International Symposium, Nov 2012

[4] S. Kavulya, et al, “An Analysis of Traces from a Production MapReduce Cluster”, Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference, May 2010

[5] C. L. Abad, et al, “A Storage-Centric Analysis of MapReduce Workloads: File Popularity, Temporal Locality and Arrival Patterns”, IISWC '12 Proceedings of the 2012 IEEE International Symposium, 2012

[6] Y. Chen, et al, “A Methodology for Understanding MapReduce Performance Under Diverse Workloads”, 2010

[7] L. Mai, et al, “Exploiting Time-Malleability in Cloud-based Batch Processing Systems”, The 7th International Workshop on Large-Scale Distributed Systems and Middleware (LADIS), co-located with ACM SOSP'13, Nov 2013

[8] J. Xie, et al, “Improving MapReduce Performance through Data Placement in Heterogeneous Hadoop Clusters”, Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium, April 2010

[9] H. Herodotou, et al, “MapReduce Programming and Cost-based Optimization? Crossing this Chasm with Starfish”, Proceedings of the VLDB Endowment, Sep 2011

[10] K. Wottrich, “The Performance Characteristics of MapReduce Applications on Scalable Clusters”, MCURCSM 2011

[11] V. Jalaparti, “Bridging the Tenant-Provider Gap in Cloud Services”, SOCC’ 12, Oct 2012

[12] S. S. Huang, J. Huang, J. Q. Dai, T. Xie, and B. Huang. “The HiBench benchmark suite: Characterization of the MapReduce-based data analysis”. Huang, Shengsheng, Jie Huang, Jinquan Dai, Tao Xie, and Bo Huang. "" In Data Engineering Workshops (ICDEW), 2010 IEEE 26th International Conference on, pp. 41-51. IEEE, 2010.

[13] W. Gao, et al. “BigDataBench: a Big Data Benchmark Suite from Web Search Engines”. The Third Workshop on Architectures and Systems for Big Data (ASBD 2013) in conjunction with ISCA 2013.

[14] Wen Xiong; Zhibin Yu; Zhendong Bei; Juanjuan Zhao; Fan Zhang; Yubin Zou; Xue Bai; Ye Li; Chengzhong Xu, "A characterization of big data benchmarks," Big Data, 2013 IEEE International Conference on , vol., no., pp.118,125, 6-9 Oct. 2013