MRM: Delivering Predictability and Service Differentiation in Shared Compute Clusters

Masoud Moshref, Abhishek B. Sharma, Harsha V. Madhyastha,
Leana Golubchik, Ramesh Govindan

moshrefj,leana,ramesh@usc.edu, absharma@nec-labs.com, harsha@cs.ucr.edu

Computing-as-a-service has been evolving steadily. Today, private clouds (e.g., Google's internal shared computing cluster) as well as public clouds (e.g., Amazon's web services (AWS), Microsoft's Azure) provide computing abstractions at various levels: bare virtual machines, specialized languages and runtimes (e.g., for massively-parallel data processing—MapReduce, Dryad), web services. For example, Amazon offers bare virtual machines as well as MapReduce clusters.

However, despite the emergence of several computing services and the wide range of abstractions they offer, little attention has been paid to the *service model*: the interface between the user and the operator that determines the type of service provided. Currently, relatively simplistic models seem to be the norm, where the operator undertakes to provide resources to complete a job, but does not provide any assurance of when the job will be completed (*predictability*) or provides limited ways in which users can ask for different levels of service (*service differentiation*). For instance, AWS and Azure use a "rental" based service model, in which users can choose from a range of virtual machine instances (of different sizes) and pay a fixed rate for each instance; the system makes no statement about when jobs finish. On the other hand, most grid-computing infrastructures charge users based on resource usage (e.g., node hours), and provide differentiation using a few discrete priority queues that are differentiated by job size and duration; low priority jobs have no guarantees on when they finish.

Our contributions. We explore the design of a service model that attempts to provide both *predictability in finish times* and the capability for *differentiation* by giving options to users to *select* a desired finish time for their jobs. Our approach, called MRM (for Map-reduce Market), achieves these goals by enabling a service model in which a user is presented with a price-deadline curve at the time when she submits a job. The user can choose an appropriate point on this curve based on the delay-tolerance of her job and her current wealth. This, in combination with deadline-based scheduling, ensures both predictable finish times and service differentiation.

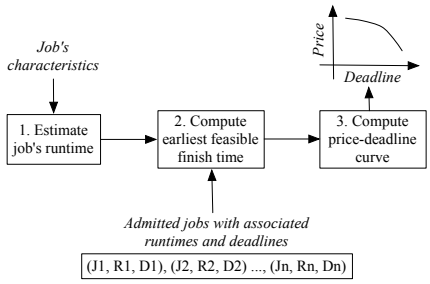

Figure 1: Overview of MRM.

To enable such a service model, we develop two key components. First, to ensure predictability, we estimate *a priori* the processing time of a job when it is submitted. We do so using Gaussian Process modeling that uses information from prior executions of similar jobs as input. Using job processing time estimates, MRM computes a range of feasible finish times for a job based on current system load while taking into account the computing resources required by the job. MRM then restricts the choice of deadlines for a job to be within the range of feasible finish times. Second, to achieve service differentiation, MRM *prices* deadlines. By charging more for earlier deadlines, MRM encourages users with delay-tolerant jobs to select later deadlines (than the earliest possible finish time). Thereby, MRM incentivizes users to offer slack in the execution schedule. This slack can be used to accommodate earlier finish times (than possible with FCFS) for later arrivals of delay-sensitive jobs.

We have instantiated MRM in Hadoop for shared MapReduce clusters. Figure 1 shows the main components. Experiments using our MRM prototype on a 40 server cluster show that MRM can achieve near-perfect predictability in realistic scenarios. Our trace-driven simulations show that by incentivizing users submitting delay-tolerant jobs to offer slack, MRM reduces the waiting time of delay-sensitive jobs. We also show that MRM can achieve deadline violation rate comparable to FCFS while providing service differentiation; in contrast, Hadoop's priority scheduler provides differentiated service but its deadline violation rate is significantly higher than MRM. The MRM technical report is available at `www.cs.usc.edu/assets/001/82853.pdf`

Copyright © 2013 by the Association for Computing Machinery, Inc. (ACM). Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.
Copyright is held by the owner/author(s).

SOCC '13, Oct 01-03 2013, Santa Clara, CA, USA
ACM 978-1-4503-2428-1/13/10.
http://dx.doi.org/10.1145/2523616.2525934