USING PEPPERDATA WITH YARN

whitepaper

May 2015

pepperdata

SUMMARY OF WHAT'S WRITTEN IN THIS DOCUMENT

If you are short on time and don't want to read the whole document, here's what you need to know:

- YARN is the framework for Hadoop 2 that makes it more scalable and introduces the ability to run more diverse workloads beyond just Maps and Reduces, like Apache Spark, Storm, Solr, and Cloudera Impala.
- YARN currently supports CPU and memory. Other resources, such as disk and network I/O, are envisioned for future versions of YARN.
- The YARN settings describe the upper limits of the physical resources that can be used, based on theoretical maximums, but not the actual second-by-second real usage of the physical resources.
- Pepperdata augments YARN by monitoring all facets of Hadoop performance, CPU, memory, disk I/O, and network by user, job, and task in real time and dynamically adjusts cluster utilization so that diverse workloads can coexist on the same cluster without the risk of job failures or missed SLAs.
- Pepperdata also gives YARN a boost by increasing cluster throughput by 30-50%.

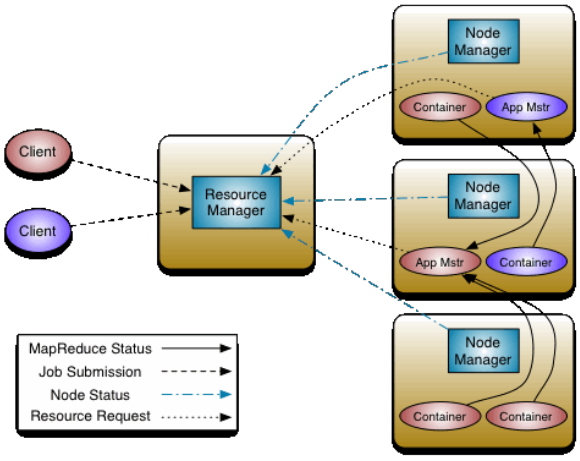
Now read on for more detail...

INTRODUCTION

The introduction of YARN in Hadoop 2 represents a major step forward in the evolution of Hadoop. While Hadoop 1 has proven itself as a powerful platform for high-speed processing of massive amounts of data on low cost commodity hardware, it is limited to batch-oriented jobs. YARN enables batch, interactive, and streaming jobs to run simultaneously on Hadoop clusters. Enterprises can now use Hadoop for more applications and new use cases. At the same time, heavier workloads increase the prospect of jobs competing with each other for physical cluster resources. This can result in missed service level agreements (SLAs) and inefficient utilization of cluster capacity. This paper provides a brief overview of YARN, identifies its limitations in production environments, and explains how Pepperdata works with YARN to overcome those limitations so Hadoop jobs can run faster, more reliably, and more efficiently.

YARN OVERVIEW

YARN, ("Yet Another Resource Negotiator") is the framework for Hadoop 2 and features a complete overhaul of MapReduce. The JobTracker in Hadoop 1 provides resource management and job scheduling services. Hadoop 2 breaks this functionality into separate daemons. The new framework features a global ResourceManager and an ApplicationMaster that is associated with individual jobs. The ResourceManager is a scheduler that allocates resources (containers) to the various running applications, subject to constraints of total capacity, queues etc. It has a pluggable scheduler that allows for various algorithms, including capacity and fair scheduling.



The ApplicationMaster negotiates with the ResourceManager for resources, which are represented as containers. A container is simply a collection of physical resources such as CPU and memory on a single node. Every node has one or more containers. Every application has its own instance of ApplicationMaster, which negotiates resources from the ResourceManager and works with the NodeManager to execute the containers.

Collectively, the new framework in YARN provides two important benefits. The first is scalability. By separating the ApplicationMaster and the ResourceManager and designing the ResourceManager as a pure scheduler, the entire system can scale more dramatically.

Second, and more significantly, by moving all application framework-specific code into the ApplicationMaster, the system can now support multiple frameworks beyond MapReduce. Technologies such as Apache Spark, Storm, Solr, and Cloudera Impala can now run side-by-side with MapReduce in a cluster. The ability to run batch, interactive, and streaming applications simultaneously means you can use Hadoop for many more things.

LIMITATIONS OF YARN

The schedulers and ResourceManager in Hadoop control when jobs start running on the cluster, but do nothing to control jobs when they are running. As a result, jobs frequently compete with each other for cluster resources. During run time, low-priority jobs can consume resources that high-priority jobs need in order to complete on time and meet service level agreements (SLAs).

Although YARN provides greater flexibility in the preallocation of resource containers, the amount of allocated memory and CPU in each container is still fixed. The YARN configuration settings describe the upper limits of the physical resources that tasks will use, based on theoretical maximums and not the actual second-by-second real usage of the physical resources in use. The net effect is that YARN allows Hadoop operators to more effectively pre-plan when and how jobs run on the cluster, but cluster resources can still end up being oversubscribed.

It should also be noted that in the current version of YARN, only CPU cores and memory can be assigned to a container. Other resources, such as disk and network I/O, are envisioned for future versions of YARN, but are not supported today.

Additional features are being added to YARN to overcome these limitations. For example, YARN-569 introduced the concept of preemption in the form of a capacity monitor to the Capacity Scheduler. It is a broad mechanism to assist the Application Manager and Resource Manager better utilize the cluster resources. Capacity management needs to be set by an administrator on a fixed time interval to calculate the free resources in the cluster. The larger the cluster the greater the time interval must be set. There are only four actions that the capacity manager can institute to improve resource utilization:

1. Container de-reservations
2. Resource-based preemptions
3. Container-based preemptions
4. Container killing

Due to the lag in instituting the progressively more expensive actions it is recommended that the policy “should operate at the macroscopic level … and not [try] to tightly and consistently micromanage container allocations.”

GREATER NEED FOR SLA ENFORCEMENT

The expanded range of Hadoop use cases that are enabled by YARN increases the potential for resource contention as more diverse workloads are added to the cluster. For administrators, it can become more difficult to ensure that low-priority jobs don’t prevent mission-critical jobs from completing within required time frames. With YARN, the ability to enforce SLAs with these diverse workloads, particularly in multi-tenant environments, becomes significantly more challenging.

PEPPERDATA OVERVIEW

Pepperdata is an enterprise software product that installs in about 20 minutes on your existing Hadoop cluster without any modifications to scheduler, workflow, or job submission processes. It is compatible with all of the Hadoop distributions (Cloudera, Hortonworks, MapR, IBM BigInsights, Pivotal PHD), and it runs on the Apache version as well. It augments YARN (as well as Hadoop 1) by monitoring all facets of Hadoop performance, including CPU, memory, disk I/O, and network by user, job, and task in real time. In addition, it dynamically adjusts cluster utilization based on user-defined policies and priorities so that high-priority production jobs, ad-hoc jobs, and HBase can coexist without risk of job failures or missed SLAs. Once installed, Pepperdata provides you with three immediate benefits:

VISIBILITY

Captures an unprecedented level of detail on cluster resource usage

Pepperdata collects 200+ metrics in real time for the four resources CPU, RAM, disk I/O, and network for any given job or task, by user or group or queue. This allows operators to quickly identify what job is causing a problem and which user submitted it. And it allows users to see what and how their jobs are doing on the cluster while they are running. Because users and operators are finally able to see what the jobs are actually doing on the cluster, the jobs can be improved.

CONTROL

Enables you to implement service-level policies that guarantee on-time completion of high-priority jobs

Pepperdata senses contention among the four resources in real time and will slow down low-priority jobs just enough to ensure the high-priority SLAs are always maintained. This SLA enforcement is ideal for multitenant environments, such as a cluster that is a central service that various business units utilize or for a cluster that has a lot of mixed workloads running. With Pepperdata there is no longer any need to isolate workloads onto separate clusters to protect high-priority production jobs. And with our HBase protection, you no longer have to worry about HBase and MapReduce jobs interfering with one another when running on the same cluster.

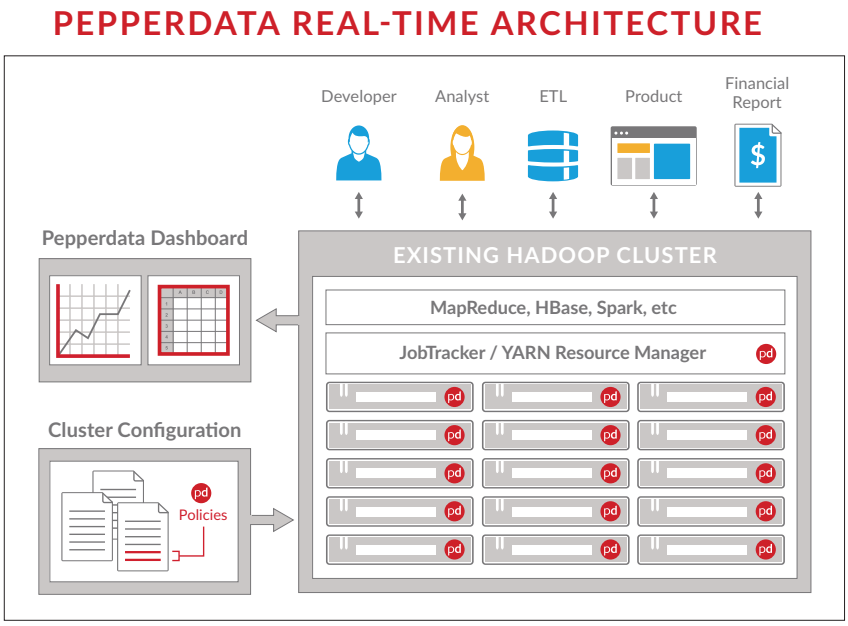
CAPACITY

Increases cluster throughput by 30-50%

Pepperdata knows the actual true hardware resource capacity of your cluster and allows more tasks to run on nodes that have free resources at any given moment. In many instances jobs will run much faster because Pepperdata will dynamically allow them to use more of the true resource on the cluster when it is available. By installing Pepperdata on your existing cluster, it will feel like you just added more servers because there will be an immediate lift in terms of the number of jobs you can run concurrently, or the same jobs run faster. And Pepperdata software costs a lot less than purchasing more servers, allowing you to take full advantage of your investments in your infrastructure.

HOW PEPPERDATA WORKS WITH YARN

Pepperdata agents, depicted by the red circles in the diagram below, run on every data node in the cluster and collect over 200 metrics related to CPU, RAM, disk I/O, and network bandwidth on a real-time basis. They communicate with each other and with the Pepperdata Supervisor, which runs on the ResourceManager node. The Supervisor has global real-time visibility into the true hardware resource utilization at any given moment in time.



Pepperdata enhances YARN by monitoring the consumption of CPU, memory, disk I/O, and network resources by every user, every job, and every task in real time, and by dynamically allocating those resources according to user-defined policies.

The net effect of using Pepperdata with YARN is the assurance that SLAs can be met, because Pepperdata ensures that high-priority applications and jobs get the resources they need, exactly when they need it. Pepperdata also provides greater capacity and throughput as a result of more efficient utilization of all resources across the cluster.

With a simple cluster configuration policies file, an administrator can specify how much of the cluster hardware to provide to a particular group, user, or job. For example, a designated production user could be guaranteed access to 80% of all cluster resources, as if they had their own cluster with that amount of capacity. Because Pepperdata monitors and controls hardware usage in real time, at any moment when that production user's jobs are not consuming the full 80% of a particular resource, other jobs on the cluster (such as ad-hoc jobs) can use the available hardware.

SUMMARY

YARN makes Hadoop a more scalable platform for distributed computing. At its core, it is a job scheduler that provides greater flexibility in terms of the types of workloads that can run on Hadoop, and more control over when and how jobs are started. Pepperdata picks up where YARN leaves off. It provides visibility and control once jobs actually start running. Organizations that use Pepperdata with YARN in production environments benefit from greater reliability and more efficient utilization of hardware resources.