pepperdata

Upsight Guarantees HBase SLAs on Multi-Workload Cluster

Customer challenges:

- Standard HBase workload on shared Hadoop cluster was disrupted by batch jobs
- Manual tuning did not solve performance problems
- Splitting the cluster was not a viable solution

Results with Pepperdata:

- Stable HBase activity despite concurrently running batch jobs
- Easier internal reporting

Background

Upsight, a San Francisco-based company, allows mobile app and web developers to incorporate marketing and analytics into their applications. With over 500 billion data points processed monthly, Upsight is one of the largest platforms available for gathering mobile and web metrics, and the firm boasts a large list of enterprise companies as customers.

To facilitate the delivery of its analytics, Upsight deployed an array of technologies, including Kafka, Apache Storm, HBase, MapReduce, Hive, and Parquet. Most of these platforms, including HBase, were run on a single shared cluster with over a thousand cores. Upsight employed a devops team that maintained this cluster, but that team was also responsible for many other components and systems — so the day-to-day operation of this shared cluster wasn't their only focus.

"We use a single cluster because it reduces unnecessary overhead. A single cluster is easier for our team to manage, and easier for us to monitor — we have more centralized visibility around resources like CPU and I/O," says Michiel Frishert, Upsight's Head of Architecture.

Streaming event data emitted by Upsight's software (from 1.5 billion unique mobile devices each month) was collected by Upsight, normalized, and then written into HBase, a distributed, NoSQL, non-relational database. After some additional processing, the data underwent metrics generation, where MapReduce (Hive) jobs were run to create the mobile marketing and analytics metrics that were sent to Upsight clients. These metrics were then loaded back into HBase.

Key Challenges

Various ETL jobs were run on a variety of schedules: hourly, daily, or in real time as needed. But because the company relied on a single cluster to run multiple workloads, there was contention for resources. Jobs scheduled to run daily were impeding jobs running in real time, and mission critical HBase activity was suffering severe performance degradation — an untenable situation, since nearly all other processes got their data from the central HBase store. Time-sensitive jobs were missing their deadlines. This was a dealbreaker for Upsight, whose business model revolves around providing real-time analytical insights to customers.

> "Without Pepperdata, we would've had to stop using HBase in the way that we do."

"We run thousands of jobs daily in addition to HBase on this cluster," says Frishert. "We tried to fix the performance issue on our own. At the end of the day, it didn't make sense to spend so much of our own time troubleshooting it."

Upsight first experimented with manually allocating resources for specific jobs — but this quickly proved inefficient and time-consuming. The company also tried segregating their cluster to isolate HBase activity, but this "fix" came with a new problem. Specifically,

by isolating HBase activity, Upsight wasn't able to get cluster-wide metrics in a centralized, easy-to-access dashboard, so they couldn't easily compare across clusters.
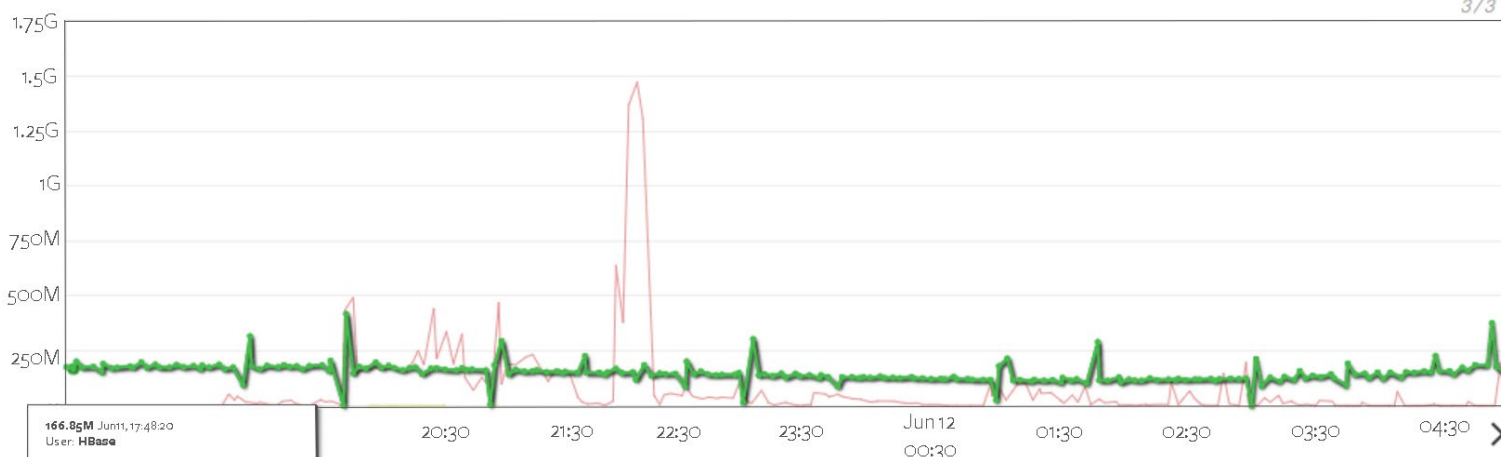
Pepperdata Protects Upsight's HBase

What Upsight needed was a solution that could dynamically handle resource contention in real time, making the thousands of necessary decisions each second to ensure specific jobs and users got guaranteed access to cluster resources. This level of control would allow mission critical jobs to complete on time even if multiple, concurrent workloads were running on the same cluster.
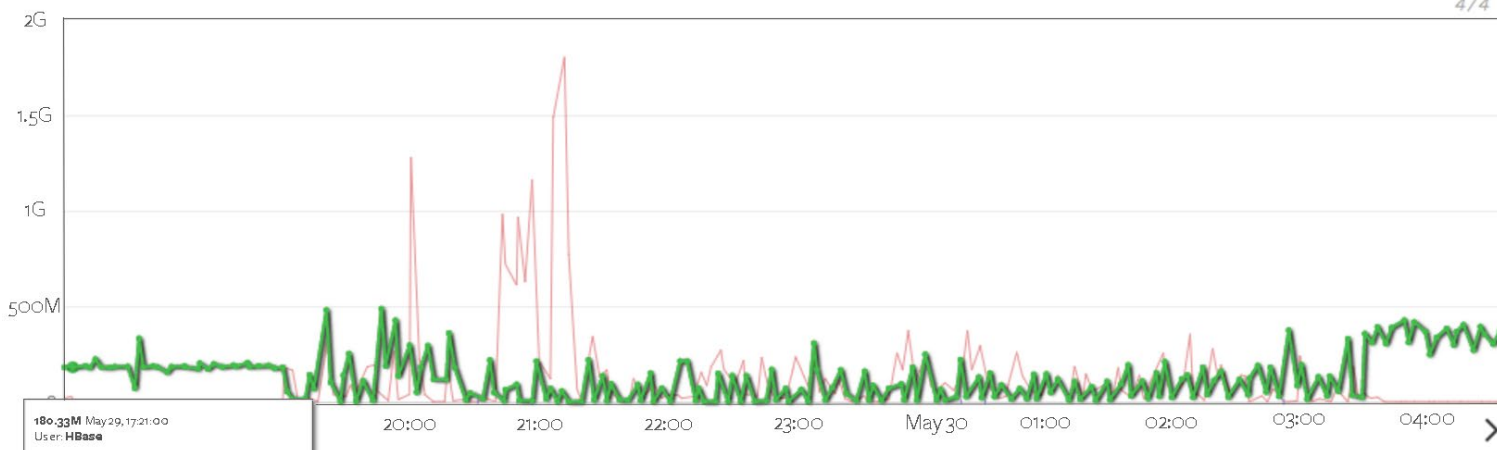
View before Upsight enabled Pepperdata policies to protect HBase activity (green) from other workloads on the cluster (red). HBase activity doesn't have consistent access to the resources it needs for predictable operation.

View after enabling Pepperdata policies to protect HBase activity (green) from other workloads on the cluster (red). HBase activity is stable and consistent, regardless of other workloads.

Upsight deployed Pepperdata policies to guarantee HBase protection in the summer of 2015. The criterion for success was having a stable HDFS write throughput for HBase, regardless of other MapReduce activity on the cluster. After configuring a Pepperdata policy guaranteeing a minimum amount of resources to be allocated to HBase jobs, Upsight saw a huge performance improvement in their cluster. Upsight was able to run an arbitrarily high number of map slots while Pepperdata's software hummed in the background, handling contention dynamically and in real time — which enabled Upsight to enforce service level agreements (SLAs).

"Without Pepperdata, we would've had to stop using HBase in the way we do," says Frishert, "and that would've required a lot of additional development and time." Pepperdata allowed Upsight to avoid having to continuously tweak their cluster and HBase settings to ensure a smooth data flow.

But that's not all. "Another benefit of having chosen to go with Pepperdata," Frishert adds, "is that because the Pepperdata software collects lots of data on MapReduce jobs, we can report on those jobs by user. And for us, a 'user' is a customer. This made internal reporting much easier; without Pepperdata, we would've had to build out that functionality ourselves." In short, Upsight saved time and development effort around creating a reporting system for just these sorts of metrics.

About Pepperdata

Pepperdata is the only solution for Hadoop and Spark that enables predictable performance and true SLA enforcement for mission critical jobs, applications, or users. With Pepperdata, you can run multiple, concurrent workloads on a single cluster with consistent, reliable performance.

Pepperdata software is trusted on over 10,000 production nodes in firms of all sizes — including 3 Fortune 50 companies.