Hadoop and Memory "Failures"

Users of Hadoop, reading their job logs, often report multiple and sometimes many failures per job. Cluster administrators, however, mostly do not see even a small fraction of the number of failures. Some of these apparent failures are bugs in users' code. Some are file systems overfilling. A few are actual hardware or OS failures. But many are none of these.

This project proposes that many of these problems arise because of a mismatch between the amount of memory reserved by Hadoop for each task and the amount used by the codes run by those tasks.

Hadoop v1.0 looks up a user supplied fixed amount of memory per task and only launches a task if a slot (approximately a CPU core) and enough memory "should be available". Hadoop is not monitoring actual memory use; it has a configuration parameter that the site administrator sets to the amount of physical memory that Hadoop should consider available, and simply sums the amounts of memory each launched task would be using if the task's configuration parameter were correct.

If the amount of memory used by a task is more than the task's configuration parameter, or if other jobs outside of the control of Hadoop are using significant memory on a node, then it is possible for the total memory use to exceed the amount of actual physical memory. Linux might be running virtual memory with a large swap partition on the local disk, allowing the overallocation to be tolerated by making some of the memory not resident during the period of congestion. But most administrators would see swapped out memory in a data-intensive "batch" program as pointless, because swapping/paging will slow the system down a lot. Instead they usually enable very little more virtual memory than there is physical memory, preferring to delay the launching of a task to swapping/paging tasks that have overcommitted memory.

Linux, if not Hadoop, will kill one or more tasks if the total virtual memory exceeds the maximum allowed virtual memory. This is one source of apparent failures.

The interesting side effect is that tasks that rarely exceed the configuration parameter amount of memory may often get away with it, because the other tasks running at the same time are using less than allocated memory. And because Hadoop will re-execute failed tasks, as long as the frequency of overcommitted memory failure is low, the job many finish successfully. One might even argue that this is a better strategy than setting the allocated memory size to a true maximum, because excessive memory allocation may reduce the number of tasks running, underutilize the CPU and memory and generally slow down the job.

To make everything much worse, users often copy configuration parameters from someone else's job, using different code. In this case the configuration parameters can be unrelated to the needs of the new user's job.

This project will probably construct a benchmark suite to demonstrate these types of failures, the phenomena of successful jobs suffering phantom failures, and the speed disadvantage of setting the memory allocation amount to a true maximum amount.

It should explore the differences in Hadoop v2.0 (Yarn). If the original benchmark does not cause the same effect, it should generate an alternative benchmark that generates a similar phenomena, if possible.

To assist real users, this project should suggest tools and tests users can do to determine if their code might have this problem. One of the hard factors in this will be dependence on input data; if the amount of memory is dependent on the actual values found in the input data, how can your benchmark suite show the user how bad this might be?

Finally, this project should develop monitoring methods to tell administrators when this is happening in their cluster, so they can encourage users to use the debugging tools and improve the code.

As a stretch goal, this project could imagine Hadoop scheduling extensions or changes that might exploit varying memory demands to better utilize the cluster resources and get jobs done faster.

For reading, start with online Apache Hadoop documents and the class assigned Yarn paper.