Paper review [1]

Yu SU 55323707

1. The Problem

There is no cluster computing framework is optimal for all application, so the general way is multiplexing a cluster between frameworks to improve utilization and allow applications to share access to large dataset, but it is costly to replicate across clusters. However, neither to run a framework per partition or to allocate a set of VMs to each framework works well, because the mismatch between the allocation granularities and framework. Although some frameworks use slot solving this problem, there is no way to perform fine-grained sharing across frameworks. As a result, the paper proposed Mesos that enables fine-grained sharing across diverse cluster computing frameworks by a common interface.

2. Challenge

The main challenge is building a scalable and efficient system. For scalability, the system should support both the current and future frameworks, which is hard to realize because different frameworks have different scheduling needs. For efficiency, the system must be fault-tolerant and highly available. Another challenge is for implementation because the architecture of Mesos is very complex and the scheduling policies are not clear.

3. Key Insight

To delegate control over scheduling to the framework, the paper proposed a new abstraction called resource offer, which encapsulates a bundle of resources that a framework can allocate on a cluster node to run tasks. This decentralized scheduling model works well in practice and is easy to implement.

The other key sight is Mesos provides some benefits to practitioners. First, each kind of organization could use Mesos no matter which frameworks you want. Second, it is much easier to develop and immediately experiment with new frameworks, which could make framework evolve faster.

And they implement Mesos with many existing good mechanisms, like providing two allocation modules for the resource allocation, using some mechanism to make resource offers scalable and robust, using ZooKeeper for fault tolerance and pluggable isolation modules for supporting multiple isolation. What's more, they built many kinds of framework on it to evaluate the system, like Hadoop, MPI and Spark.

4. Limitation

One limitation is all the task assignment is controlled by one central machine. On one hand, this will be the bottleneck of the system for the network bandwidth or the overhead of IO. On the other hand, this is a waste for other standby master. One more limitation is Mesos is not flexible for the resource allocation, because it could only track the allocation instead of the actual usage, which will cause the waste of resources.

5. Future Work

Now Mesos only could support for the task isolation, in the future they should add network and IO isolation support. And they also need to optimize their resource allocation policy to improve its efficiency of the system.

[1] A. Ghodsl, M. Zaharia, B. Hindman, A. Konwinski (2011). Mesos : a platform for fine-grained resource sharing in the data center. *Proceedings of the 8th USENIX. 295-308*