

Accommodate Apache Yarn to Long-lived Services

Huiyi Li, Ruonan Rao
Shanghai Jiaotong University

Abstract

Large-scale cluster scheduling and management systems are becoming dominant platforms for a variety of applications and services. These complex “cloud” systems often run on clusters of thousands of unreliable commodity machines and must handle all kinds of failures, schedule and deployment requirements while achieving the maximum resource utilization. We take Apache Yarn to a full analysis and argue that Yarn currently excels in handling batch jobs without interacting with outside systems or clients. In this paper, we propose a solution to service registration, log management, node reallocation problem that Yarn had and accommodate Yarn to long-lived services.

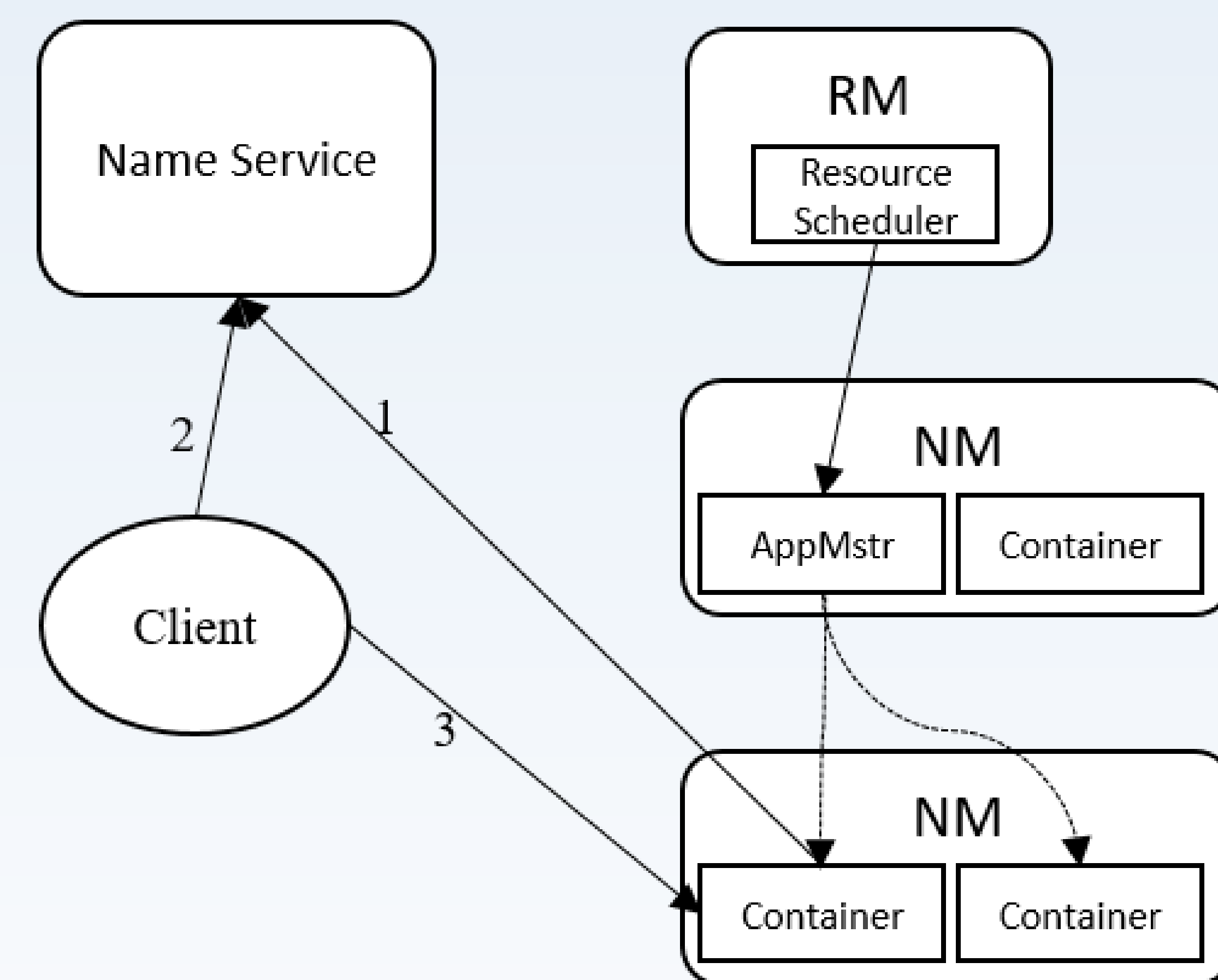
Objectives

To effectively accommodate Apache Yarn to long-lived services. In order to do that, we need to make adjustment to Apache Yarn from three aspects: service registration, log management and container resource reallocation.

Methods

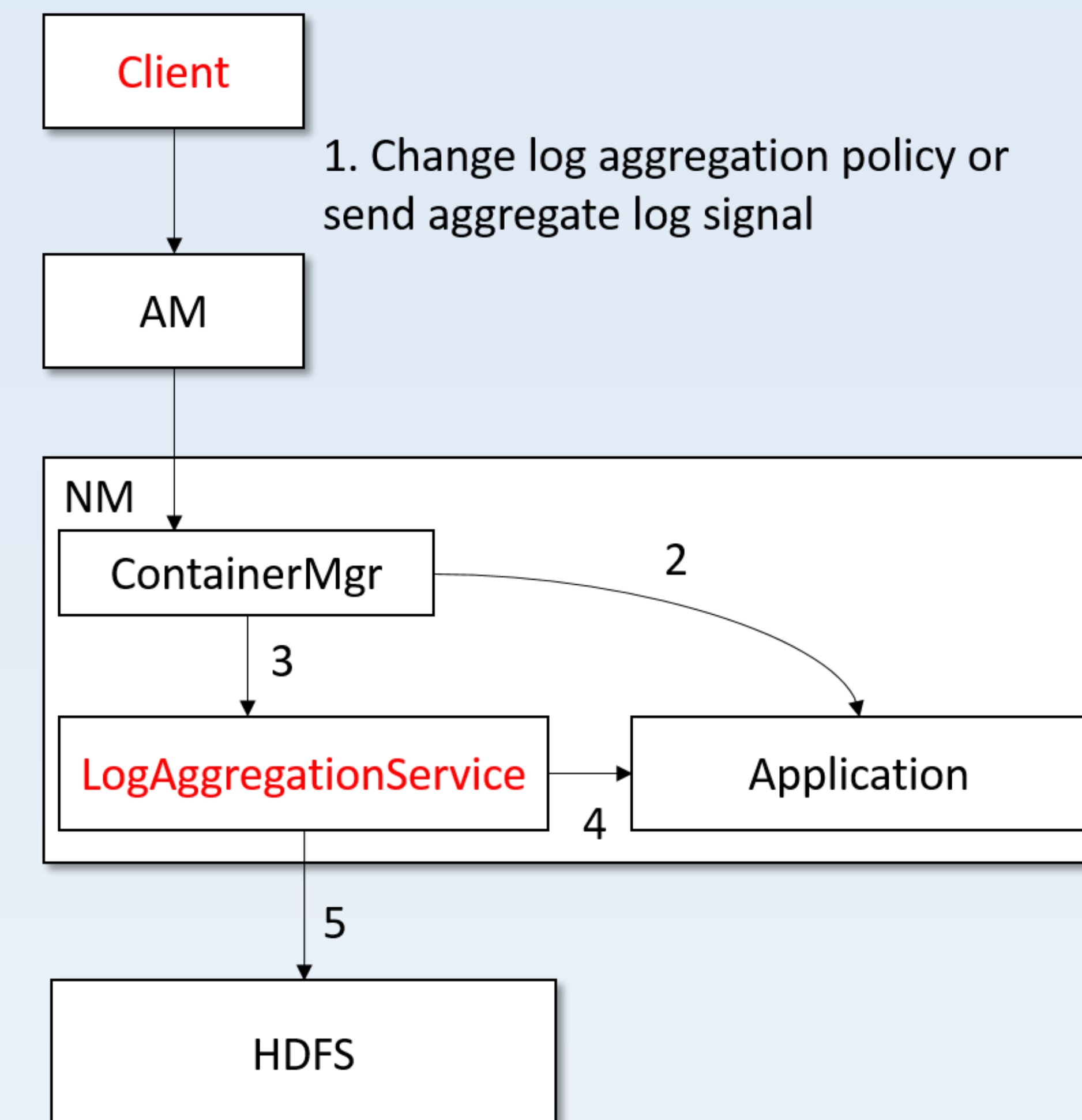
Services registration:

- 1) Register both internal (e.g. Storm Nimbus node and bolts rpc services) and external services
- 2) Use zookeeper as base name service

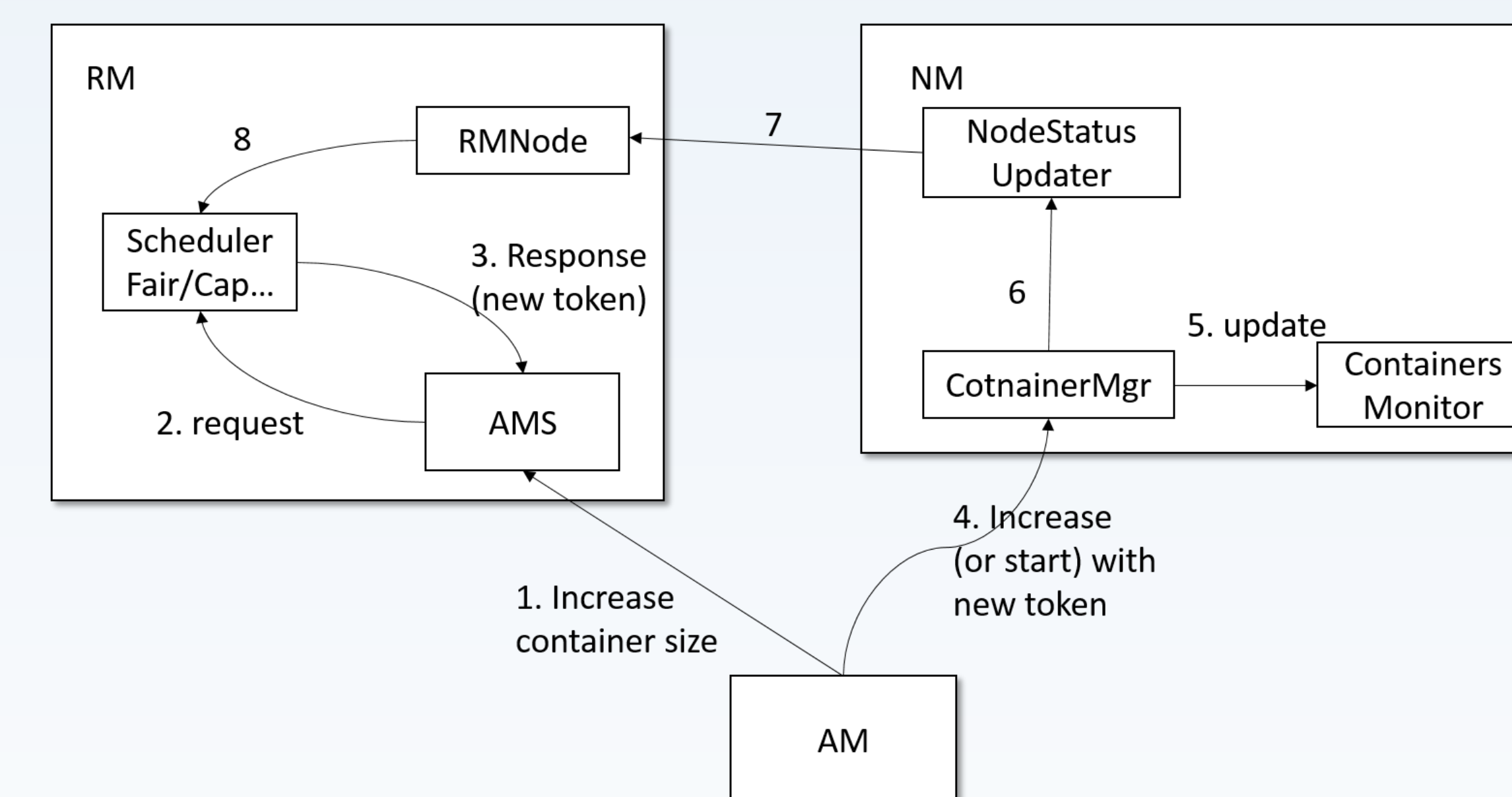


Log management:

- 1) Enable clients to dynamically change or choose log aggregation policy or send log aggregation signal
- 2) Time-based & size-based log aggregation policy



Container resource reallocation:

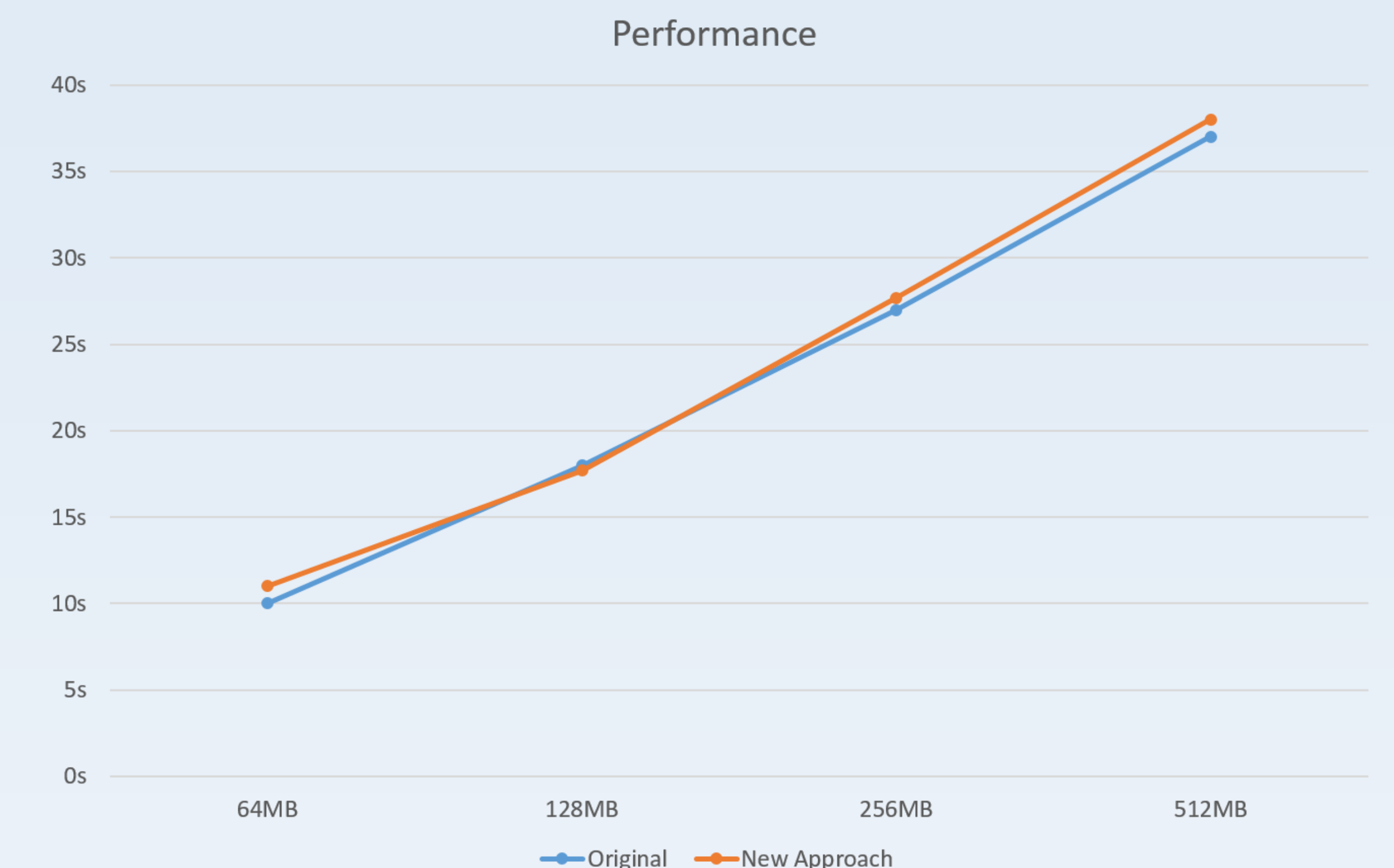


Results

Is our design successfully accommodate Apache Yarn to long-lived services?

Yes, test results shows that we could deploy any distributed or non-distributed services such as Apache Storm, Spark, MongoDB, a pool of tomcat servers to Apache Yarn.

How does our design affect the short-lived (map/reduce) services running in Yarn?



Conclusion

Our design is mainly based on the work of Apache Slider , which is a project that aims to allow long-lived services to run on YARN cluster and minimizes the management trouble. Slider did this by add additional components and agents in Application Master and Node Manager which was used to track the deployed service information and allow changes with command line interface. Our approach uses zookeeper to exploit its strength on service registration instead of reinventing the wheel.