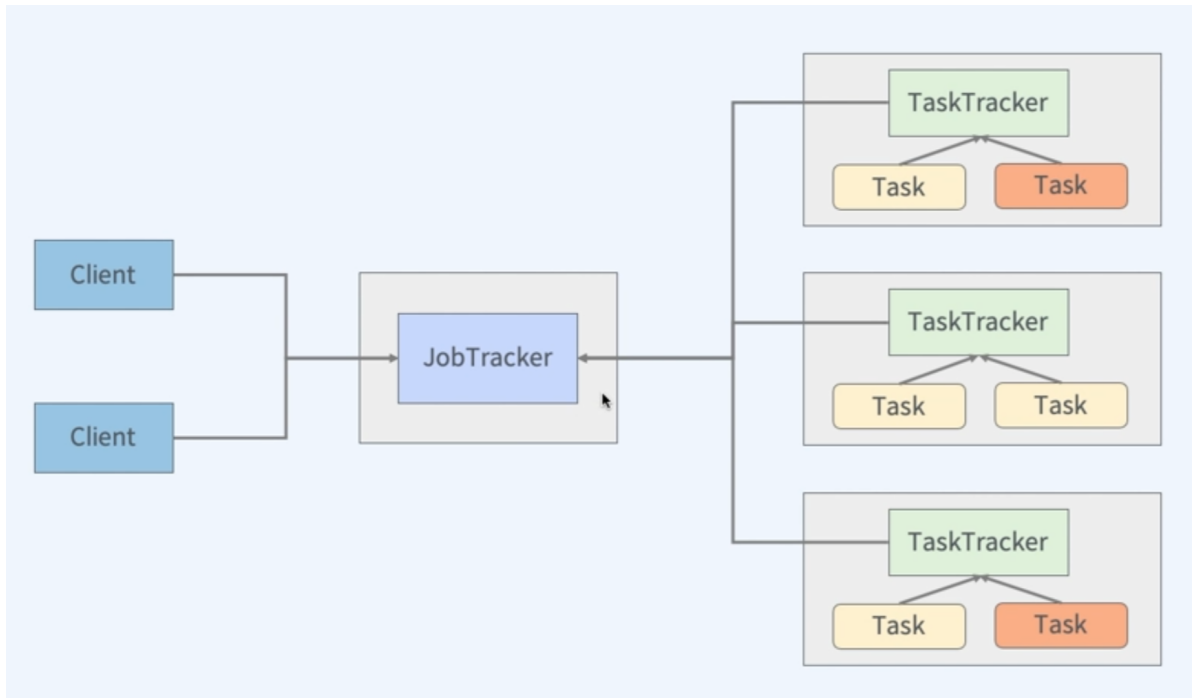


Yarn

MapReduce1



- Job Tracker의 부담이 큼
 - 리소스 매니징
 - 스케줄링
 - 태스크 모니터링

MapReduce1의 문제점

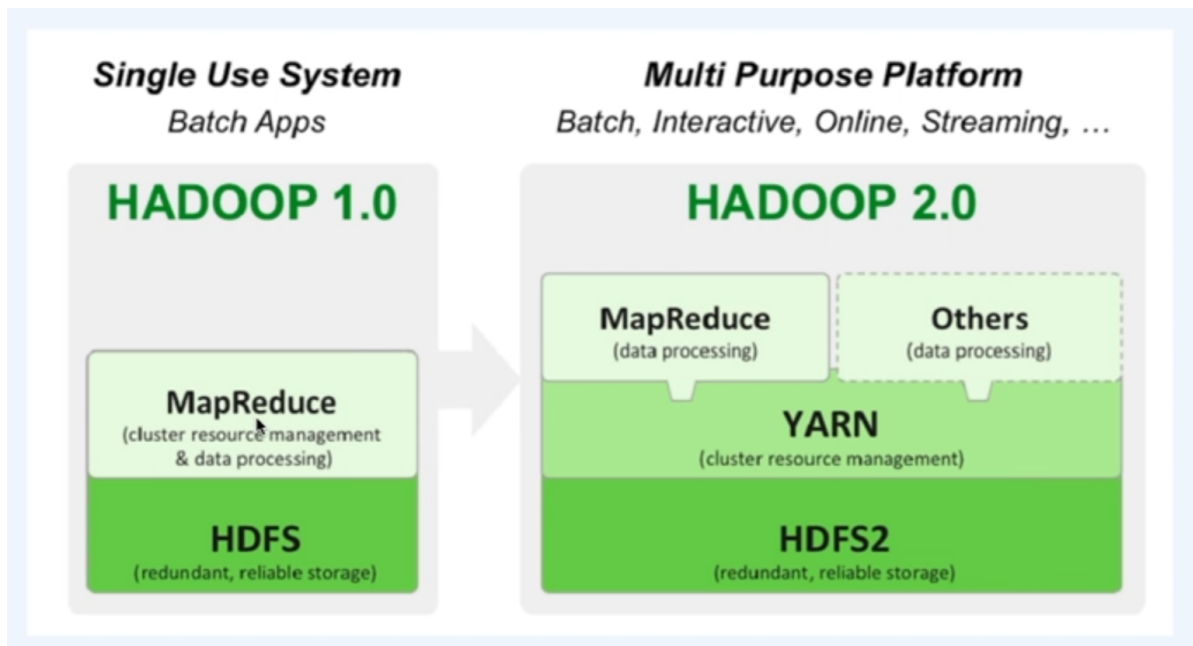
- 클러스터의 확장성 병목
- 신뢰성과 가용성 문제
 - 잡트래커에 문제가 생기면 서버 전체 문제 발생
- 맵리듀스 프로그래밍 모델만 지원
- 클러스터 이용률 문제

YARN(Yet Another Resource Negotiator)의 등장

특징

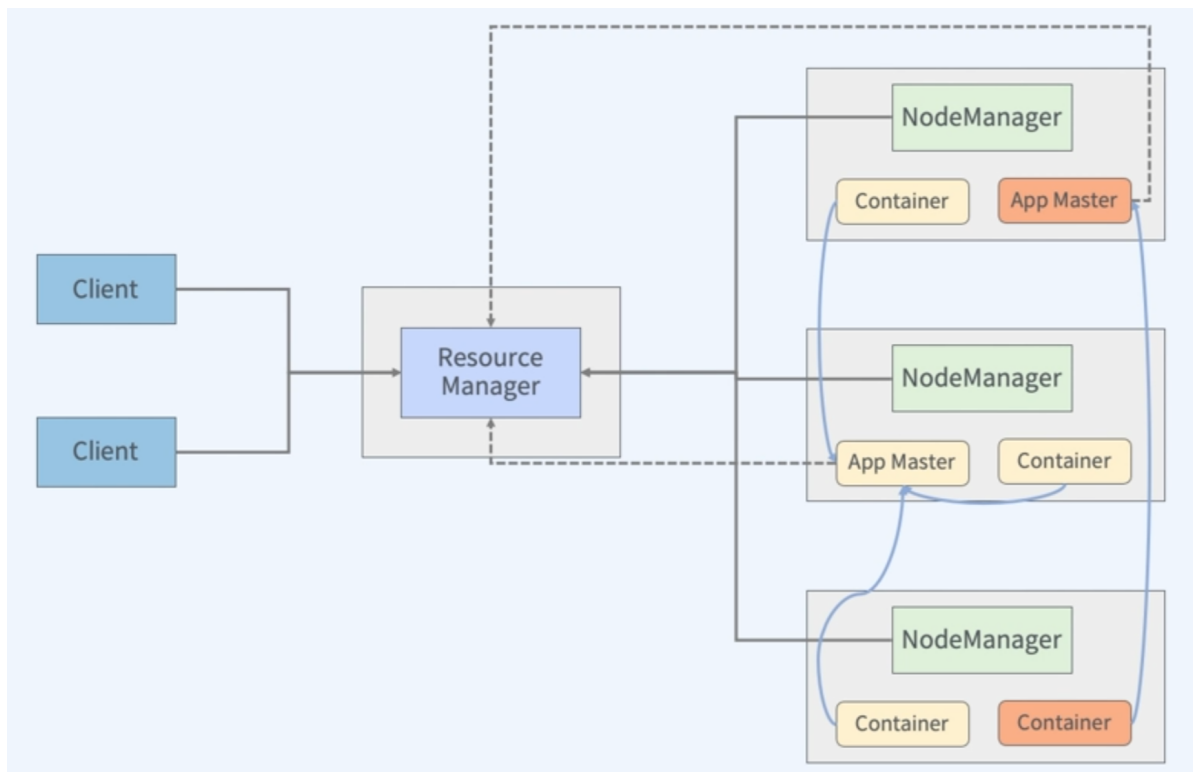
- JobTracker의 두 가지 중요한 부분의 책임 분리
 - Resource Manager
 - 리소스 컨테이너 단위로 추상화 하여 배분
 - 클러스터 이용률 개선
 - Application Master
- 확장성 개선

- 다양한 워크로드 지원
- 클러스터 이용률 개선
- 기존 맵리듀스 호환성 지원



- MapReduce 이외에도 다른 프로그래밍 모델을 이용해 데이터 처리가 가능하도록 함

Yarn Architecture



- Resource Manager
 - 모든 클러스터의 자원을 중재
 - 플러그인 가능한 스케줄러, Job관리하는 ApplicationsManager로 이루어져 있다.
- Node Manager
 - 각 노드를 관리

- 리소스 매니저에 노드의 상태를 공유
 - application container의 라이프 사이클 관리
 - 로그 관리 등
- AppMaster
 - 리소스 매니저와 자원을 협력하여 Task 실행
 - 하트비트 전송해서 어플리케이션의 상태를 갱신
- Container
 - 단일 노드에서 물리적인 리소스의 단일

MapReduce1 vs YARN 컴포넌트 비교

MapReduce 1	YARN
JobTracker	ResourceManager, ApplicationMaster, TimelineServer
TaskTracker	NodeManager
Slot	Container

YARN 컴포넌트

- Resource Manager
 - 클러스터 리소스를 중재하는 마스터
 - 주요 컴포넌트
 - Scheduler
 - FIFO
 - Capacity
 - Fair 중 선택 가능 - 평균적으로 할당
 - Applications Manager
 - 다수의 App들의 유지
- Node Manager
 - 하둡 클러스터의 노드를 관리하는 역할
 - 시잘할 때 Resource Manager에 등록
 - heartbeat를 통해 Resource Manager에게 상태를 보냄
 - kill
- Application Master
 - 실행 상태를 모니터링 하면서 상태 추적
 - application - 사용자가 제출한 단일 작업
 - Resource Manager와 자원 협상 후 Node Manager와 일함
- Container
 - 단일 노드의 자원
 - Resource Manager가 스케줄링
 - Node Manager가 관리 감독
 - 동적 할당이 가능한 형태

YARN의 동작방식

