

MapReduce 프레임워크를 위한 JobTracker 결함허용 메커니즘

황병현¹ 박기진²

아주대학교 산업공학과¹, 아주대학교 산업정보시스템공학부²

heal83@ajou.ac.kr, kiejin@ajou.ac.kr

A JobTracker Fault-tolerant Mechanism for MapReduce Framework

Byunghyun Hwang¹ Kiejin Park²

Department of Industrial Engineering, Ajou University¹,

Division of Industrial and Information Systems Engineering, Ajou University²

요약 - 클라우드 컴퓨팅 서비스를 제공하기 위해서는 클라우드 컴퓨팅에 적합한 데이터 분산 저장 및 병렬 처리가 가능한 IT 인프라 구축이 필수적이다. 이를 위해서 분산 파일 시스템 중 하나인 HDFS(Hadoop File System)와 병렬 데이터 처리를 지원하기 위한 MapReduce 프레임워크 관련 연구가 각광 받고 있다. 하지만 MapReduce 프레임워크를 구성하는 JobTracker 노드는 SPoF(Single Point of Failure)이기 때문에, 작업 도중 JobTracker 노드의 결함이 발생하게 되면 전체 작업이 실패하게 된다. 위와 같은 문제를 해결하기 위해서 본 논문에서는 MapReduce 프레임워크의 JobTracker 노드 결함 발생에 대처할 수 있는 결함허용 메커니즘을 제안하였다.

1. 서론

클라우드 컴퓨팅에 적합한 IT 인프라는 데이터의 효율적인 관리가 가능해야 하며 저렴하게 구축이 가능하고, 유지보수가 용이해야 한다. 클라우드 컴퓨팅 분야를 선도하는 기업 가운데 하나인 구글(Google)은 자체적으로 개발한 GFS (Google File System)를 사용하여 자사의 클라우드 서비스를 제공하고 있으며 [1], GFS 와 함께 대용량 데이터 분산 처리 시스템의 파일 시스템으로 널리 사용되는 분산 파일 시스템으로 HDFS(Hadoop File System)가 있다.

대용량 데이터 분산 파일 시스템을 효과적으로 활용하기 위해서 GFS, HDFS 을 활용하는 기업들은 MapReduce 라는 분산 프로그래밍 프레임워크 (Framework)를 사용하고 있다 [2]. MapReduce 는 데이터를 분석하여 사용자가 필요로 하는 데이터와 연관된 중간 데이터를 생성하는 맵(Map) 과정과, 중간 데이터를 정리하여 최종 결과를 도출하는 리듀스(Reduce) 과정으로 구성되며, Java 언어를 사용하여 Map 함수와 Reduce 함수를 구현할 수 있다. MapReduce 프레임워크는 각기 작성된 Map 과 Reduce 함수를 실행하고 관리하는 TaskTracker 노드와, Job 의 초기화 및 Task 스케줄링을 위한 JobTracker 노드, JobTracker 노드에 작업의 시작을 알리고 결과를 보고 받는 Client Node, 이렇게 3 가지 종류의 노드로 구성된다. 언급된 노드들 가운데 TaskTracker 노드의 경우, 결함(Fault) 발생 시 대처하기 위한 메커니즘이 존재하지만 JobTracker 노드는 Job 의 관리 및 Task 스케줄링을 수행하는 중요 노드임에도 불구하고 SPOF(Single Point Of Failure)이며, JobTracker 결함

발생에 대한 대처 메커니즘이 따로 존재하지 않는다 [3]. 이에 본 논문에서는 JobTracker 결함 발생에 대비하기 위해서 결함이 발생하기 이전에 TaskTracker 의 작업을 스케줄링하여, JobTracker 노드에 결함이 발생할 경우, 진행 중이던 MapReduce 작업이 완료 될 수 있도록 지원하는 메커니즘을 제안하고자 한다.

2. 본론

MapReduce 의 작동 과정은 그림 1 과 같다. MapReduce 작업은 MapReduce 의 runJob 메소드 실행으로 시작된다. runJob 메소드를 실행하면 JobClient 의 객체 한 개가 생성되며, 이 객체의 역할은 JobTracker 노드에서 새로운 Job ID 를 할당 받고, 해당 Job 에 대한 명세를 확인 후 HDFS 로부터 작업을 실행하는데 필요한 자원을 Client 노드로 복사한다.

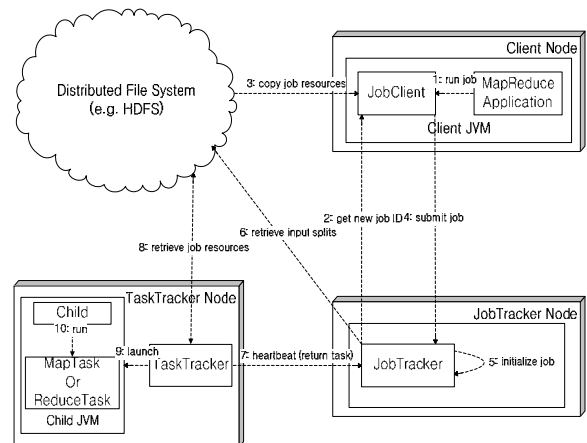


그림 1. MapReduce 작업 실행 구조

작업 실행에 필요한 자원은 Job JAR 파일(실행 소스 파일), 환경설정 파일 등이 있다. Job ID 를 할당 받고, 작업에 필요한 자원 파일의 복사가 완료되면, 'submit Job' 메시지를 JobTracker 노드의 JobTracker 객체로 보내서 작업 실행을 위한 준비가 완료 되었음을 알린다.

JobTracker 노드는 Job 의 초기화 및 Job 스케줄링을 수행한다. JobTracker 가 submit Job 메시지를 수신하면 Job 스케줄러의 Queue 에 해당 작업을 배정하고 작업을 초기화 한다. 그 다음, Job 스케줄에 따라 수행 차례가 된 Job 의 첫 input split (특정 Job 을 수행하기 위한 대용량 데이터의 조각 가운데 첫 번째 조각)을

HDFS로부터 검색한다. TaskTracker는 JobTracker에게 주기적으로 신호(Heart Beat)를 보내서 TaskTracker의 결함 발생을 확인하도록 함과 동시에, TaskTracker는 Task를 리턴 받게 된다. JobTracker로부터 부여 받은 Task를 수행하기 위해서, Task에 해당하는 자원을 HDFS로부터 검색한 후에 TaskRunner 객체를 새로운 JVM 위에 생성하여 Map 또는 Reduce 함수를 실행시키게 된다.

위와 같은 일련의 절차에 따라 MapReduce 작업이 수행되며, Client 노드와 JobTracker 노드는 물리적으로 하나의 랙(Rack)에 위치하고 있다. TaskTracker 노드는 작업의 설정에 따라 무수히 많이 존재할 수 있다. 현재 TaskTracker 노드에 결함이 발생하는 경우, 해당 노드를 작업에서 배제시키고, JobTracker에서 다시 스케줄링을 하여 작업이 끊김 없이 연속적으로 수행될 수 있도록 고안되어 있다. 하지만 JobTracker 결함이 발생하는 경우, SPOF로 설계되어 있기 때문에 전체 작업 실패로 이어지게 된다. 이와 같은 문제를 해결하고자, 본 논문에서 제안하는 메커니즘은 JobTracker의 결함 발생을 대상으로 한다. 메커니즘의 구성은 수정된 JobClient 및 JobTracker 노드에 본 메커니즘의 실행을 위한 FailSafe 객체, Cold Standby 개념으로의 결함허용 기법 적용을 위한 여분 JobTracker의 추가로 이루어진다(그림 2 참조).

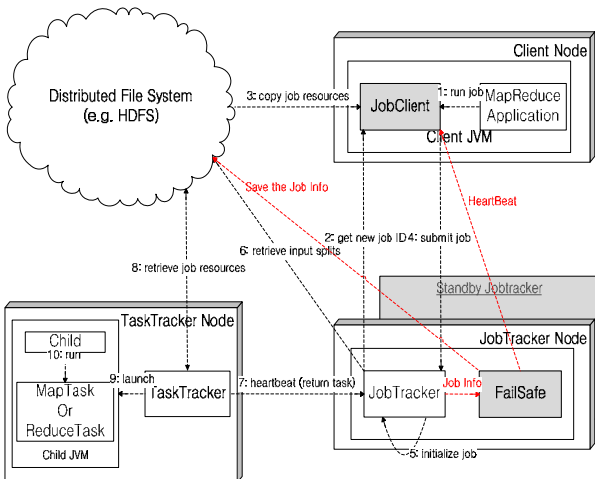


그림 2. 결함 허용 메커니즘이 적용된 MapReduce 실행 과정

FailSafe 객체의 역할은 JobTracker 노드의 결함에 대비하여 작업 정보(Job Info)를 파일시스템에 저장해 두는 것과 JobTracker의 결함 여부를 확인할 수 있도록 JobTracker의 HeartBeat를 JobClient 객체로 전송하는 것이다. 새로이 제안하는 JobClient 객체는 JobTracker의 HeartBeat를 수신하고, JobTracker의 결함을 감지하는 기능을 추가적으로 수행한다. FailSafe 객체는, 전체 TaskTracker 노드가 HeartBeat를 JobTracker에 모두 한 번 보내는 기간을 1 주기(Period)라고 했을 때, 1 주기에 1 회 현재까지 진행된 작업의 결과 및 작업 중인 TaskTracker 정보, Job 정보 등을 분산 파일 시스템 내에 저장 및 갱신한다. JobTracker의 고장이 발생하지 않는 한, 저장된 내용은 1 주기에 한 번씩 갱신된다. 또한, FailSafe 객체는 현재까지의 작업 정보를 분산 파일 시스템에 저장함과 동시에 JobClient 객체로 HeartBeat를 송신한다. JobTracker의 복구 과정에서, FailSafe 객체는

이전에 고장이 발생한 JobTracker의 FailSafe 객체가 분산 파일 시스템에 기록해둔 결함 발생 이전까지의 작업 정보를 확인하여, 결함 발생 이전 상태로 복구를 진행한다(그림 3 참조).

기존 작업 완료 과정은 JobTracker의 상태에 따라 결정되었다. JobTracker가 자신의 상태를 'Successful'로 전환하고 작업 완료를 선언한다. 작업이 성공적으로 완수되지 못하고 실패하게 되면, JobClient는 작업 실패 메시지를 사용자에게 알리게 된다. JobClient는 작업 최종 종료 시점에 분산 파일 시스템에 접근하여 MapReduce의 최종 결과, 즉 Reduce 결과물을 사용자에게 알린다.

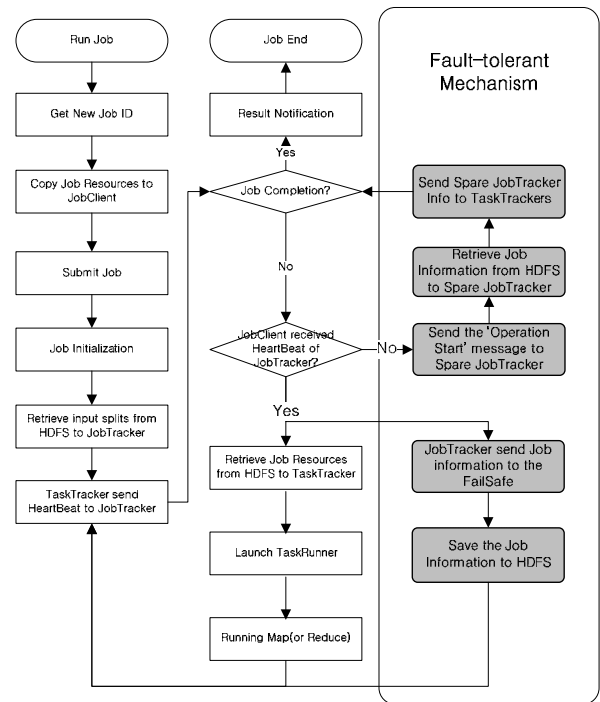


그림 3. 결함허용 메커니즘이 적용된 MapReduce 실행 과정 플로우차트

3. 결론

본 논문은 MapReduce 프레임워크를 구성하고 있는 JobTracker 노드의 SPOF 문제를 해결하기 위하여, JobTracker 노드 내에 Fail-Safe 기능을 갖춘 객체를 포함시켜 JobTracker 고장 발생에 대처할 수 있도록 하였으며, 고장 발생에 따른 작업 실행 메커니즘을 고안하였다. 향후 본 논문이 제안하는 메커니즘을 실제 클라우드 컴퓨팅 환경에 적용하여 본 메커니즘의 효과를 검증하는 연구 수행이 필요하다.

참고문헌

[1] S. Ghemawat, H.Gobioff, and S.Leung, "The Google File System," In Proceedings of the 19th ACM Symposium on Operation Systems Principle, 2003
 [2] Wikipedia, <http://en.wikipedia.org/wiki/MapReduce>, 2010
 [3] Tim White, Hadoop: the definitive guide, O'Reilly Media, June 2009