3. 연구내용

3.1 Prefetch Thread 구성

async하도록 thread를 구성

이미 heartbeat thread도 async하긴 한데 heartbeat thread는 이미 하는 일이 많아서 이거까지 시키면 오히려 악영향을 끼친다..(이걸 굳이 넣을 필요까진 없을 듯)

Prefetch api를 만든 것..?

3.2 데이터 prefetching 방법

Heartbeat thread에서 돌린다..

Prefetch 구현한 방법

Alluxio worker <-1-> Alluxio Master <-2-> Client(사용자)

1. BlockMaster에서 최종반영

workerHeartBeat에서 주기적으로 worker에 반영된 정보를 master에 기록함.

1. RPC통신(네트워크로 코드, data를 전송, 수신)

Worker에 실질적인 data가 존재하기 때문에 worker에서 copyBlock을 해야 함. 따라서 client에서 내린 prefetch 명령은 master로 감. 근데 문제가 master에서 worker로 명령을 직접 내릴 수가 없어서 default file system master에서 block master(worker가 master에 최종적으로 update된 정보를 반영하는 class)에 prefetch 해야 할 blockID를 저장. 그리고 worker에서 worker hearbeat를 해서 block master에 접근할 때 만약 prefetch 해야 할 block이 있으면 받아와서 worker로 가져옴. 그리고 worker에서 copy block 실행

실험 & motivation -> 서론 약간 고치기..?

1.Spark만

2.Alluxio 대체

3.Alluxio 대체+prefetch (large scale일때, MEM이 한정적이니까 drop이 많이 일어날 때 유리하다..)