



Design Document : Overview

이 프로젝트의 목표는 다음과 같은 입력, 출력을 만족하는 프로그램을 구현하는 것이다.

Input Specification

- 1 master, a fixed number of workers with their IP addresses
- Input blocks of 32MB each on each worker

Output Specification

- An ordering of workers
- An ordering of output blocks on each worker
- Sorted output blocks (of any size)

프로그램은 각각 master, worker machine에서 수행되는 두 개의 executable file로 구성되어 있다.

각 worker는 다음과 같은 알고리즘으로 동작한다.

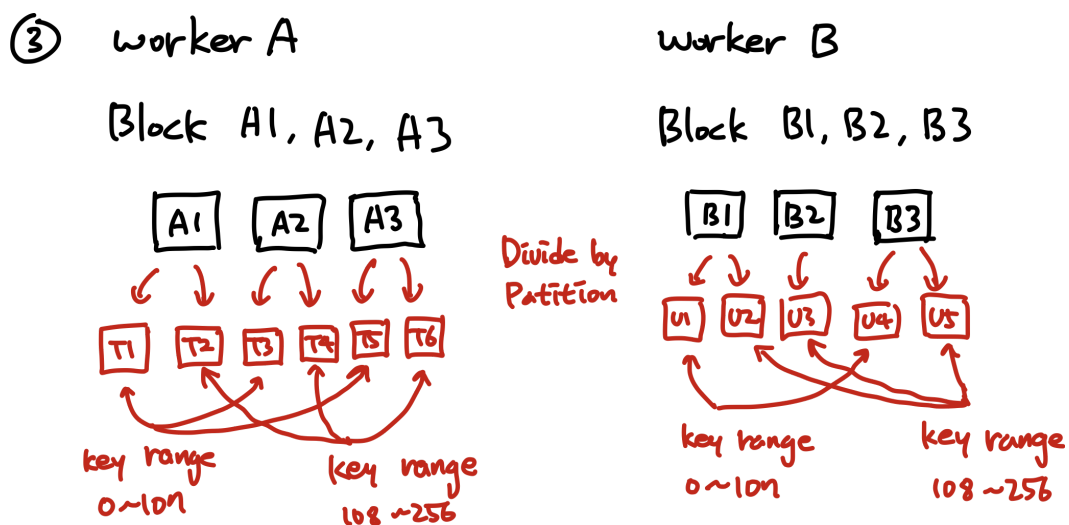
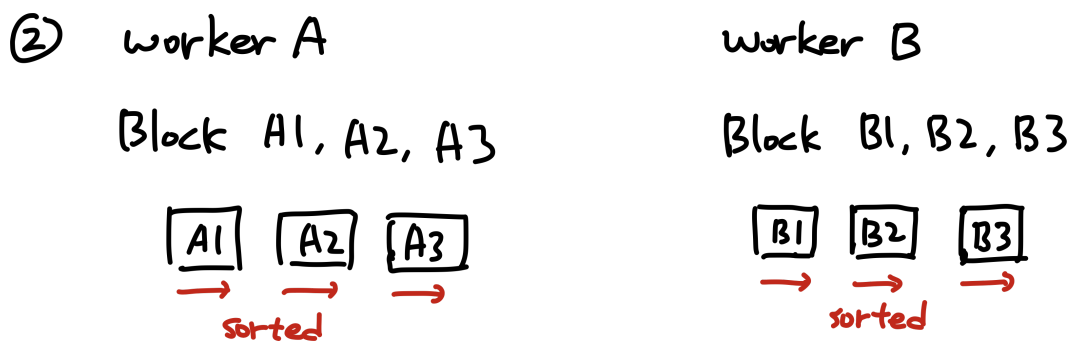
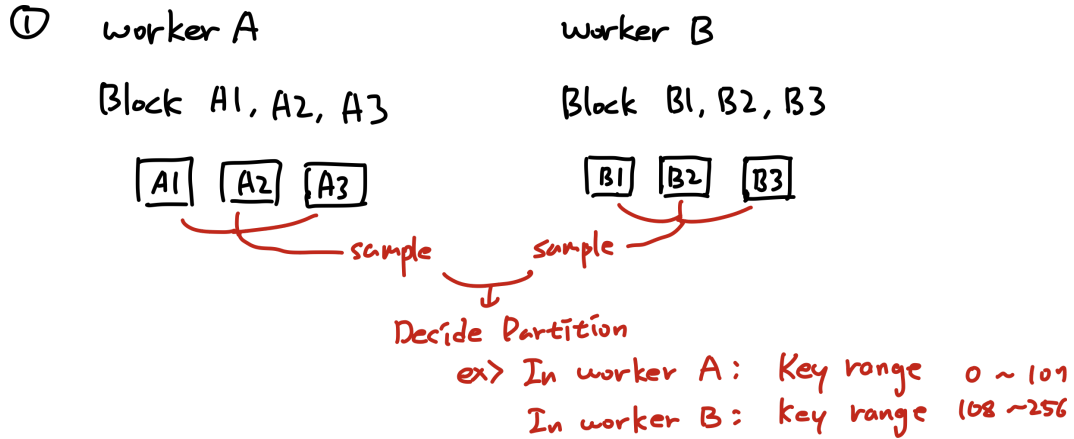
Worker Algorithm

1. 초기화 수행
2. 각 Block으로부터 몇개의 tuple을 sample하여 master로 전송하고, 각 worker가 결과적으로 저장하게 될 key range(partition)을 받아온다.
3. Machine의 모든 block을 정렬한다.
4. 정렬 이후, 각 block을 partition에 따라 block을 나눈다.
5. Partition에 따라 나뉜 machine의 블록들을 각 partition을 담당하는 machine으로 전송한다.
6. 각 machine의 tuple들을 Merge한다.

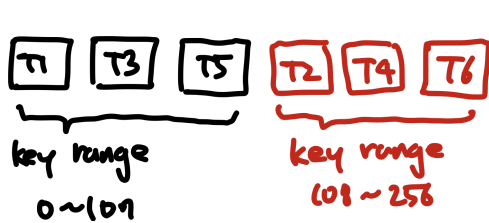
Example

- Specification Assumption

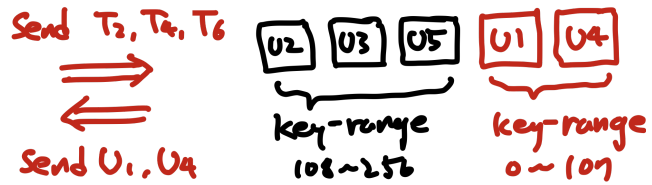
- Two workers A, B and each has three blocks.
- Tuple key range: 0~256 for simplicity



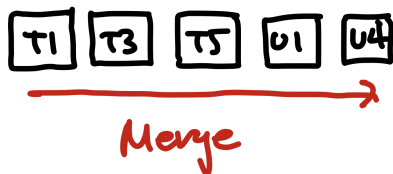
④ worker A



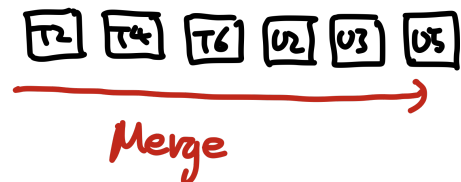
worker B



⑤ Worker A



Worker B



Worker Concept

- Worker는 여러 블록(파일)을 가지고 블록들을 리스트로 관리한다.
- Worker는 위의 알고리즘에 의해 다음의 다섯 Phase로 나뉜다.
 1. Initialization
 2. Sample
 3. Sort & Partitioning
 4. Block Transfer
 5. Merge
- Worker는 각 phase에 진입할 준비가 되었을 때, master에게 request를 보내고, response를 받은 이후 phase에 진입하여 작업을 시작한다.
 - 각 phase의 진입 시점은 master에 의해서 조정된다.
 - Master로부터의 response에는 각 phase의 작업을 위해 필요한 정보가 포함된다.

Master Concept

- Master는 다음의 세가지 작업을 한다.
 - Worker에게서 sample을 받아 partition을 나눈다.
 - 모든 worker가 phase에 진입할 준비를 마쳤을 때, 각 worker의 작업을 시작하도록 한다.
 - Block transfer phase에서 worker가 어떤 worker로 파일을 전송할지를 결정한다.