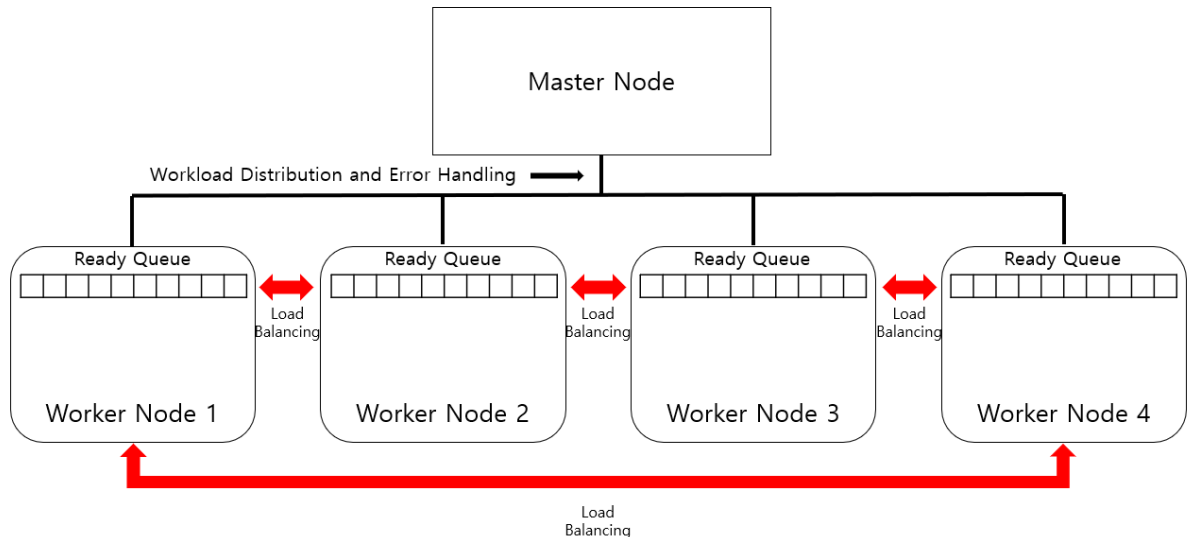


## HW#1 데이터통신 프로그래밍 과제

### 서버-클라이언트간 Multi-Thread 및 분산 처리를 활용한 대규모 행렬 곱셈 구현

- DUE: 2024.09.25(수) 23:59 (Assigned: 2024.09.12.16:30)
- How: 1개화일 제출 – G조이름HW1.zip (조별로 한명만 제출하면 됨)
  - ◆ 모든 소스파일 및 결과파일 전체를 압축한 G조이름HW1.zip으로 제출 ex) G1HW1.zip

1. (시뮬레이션 환경 및 구성요소) 다음과 같은 실험환경을 구성한다. 1개의 master node와 4개의 worker node는 thread간 Socket 통신으로 동작하며 연결 구조의 제약은 없다. 각각의 객체는 다음의 기능을 수행한다.



- ① Master node: 전역변수 혹은 임의의 객체인 System Clock을 유지, 관리, 업데이트 하고 프로그램이 수행되면 System Clock을 수행시킨다. 프로그램이 시작되면 연산 결과를 저장할 1000x1000 크기의 행렬을 2개 생성 후 각 요소들을 무작위 정수(1~99사이)로 업데이트 한다. 각 worker node에게 행렬 곱셈 작업을 분배하는 역할을 하며, worker node의 연산 처리량에 따라 동적으로 분배가 이루어져야 한다. 이때 작업 1회는 결과 행렬 C의 요소 1개( $C[i, j]$ )를 위한 연산 작업으로 정의한다. worker node에서 연산 결과를 받았을

$$C[i, j] = \sum_{k=1}^n A[i, k] \times B[k, j]$$

때 성공 시 결과 행렬의 업데이트가 이루어지고, 만약 실패 시 다른 worker node로 재할당을 통해 장애 처리를 진행하여야 한다. 이때 기존 분배 작업과 마찬가지로 공평하게

이루어져야 하고, 장애 처리를 위한 작업이 최우선 순위로 분배되어야 한다. 전체 행렬의 모든 연산이 끝났을 때, 총 연산에 소요된 system clock의 시간, 모든 행렬을 출력 후, worker node와의 연산을 종료한다. 모든 event 및 결과 정보를 log로 기록한다.

- ② Worker node: 각각의 worker node는 시작이후 master node에서 전달받은 작업의 곱셈 연산을 진행한다. 각 worker node당 최대로 할당 받을 수 있는 작업의 수는 10개이며, 그 이상의 작업은 실패로 처리한다. 연산 작업은 1~3sec 동안 무작위하게 시간이 소요된다고 가정하며, 연산의 결과와 함께 소요된 시간을 master node에 전달하여 업데이트 한다. 전달받은 작업은 80%의 확률로 연산이 성공하며, 만약 실패하게 된다면 실패되었다는 결과를 master node에 전달한다. 위의 연산 작업을 진행함과 동시에 각 worker node 간 통신을 통해 부하를 분산하고자 하여야 한다. 예를 들어, 자신이 할당 받은 작업의 대기시간이 길어진다면 작업의 대기시간이 짧은 다른 worker node로의 통신을 시도하여 작업의 재할당을 시도하여 넘겨주어야 한다. 이를 통해 최대한 공평하게 작업이 진행될 수 있도록 유기적으로 worker node가 작동해야 한다. 전체 행렬의 연산이 완료되어 master node와의 연결이 중단되기 전 각각의 worker node는 “연산 성공 및 실패 횟수, 작업 처리량, 작업의 평균 대기시간, 전체 수행시간”을 출력한다. 각 worker node간 모든 event를 Log로 기록한다.

※ Master node는 반드시 AWS나 구글클라우드 등 Physically 외부서버에 구현 되어야 함!!

※ Master node와 모든 worker node의 통신은 반드시 Socket으로 구현되어야 함!!

※ 모든 worker node는 Thread를 통해 구현되어야 함!!

2. (시뮬레이션 시나리오) 다음과 같은 실험 시나리오를 수행하여 각각의 구성요소 별 Log를 기록한다. 즉, master node와 4개의 worker node는 각각 모든 이벤트와 연산 정보를 Master.txt, Worker1.txt, Worker2.txt, Worker3.txt, Worker4.txt에 기록한다.

- ① 시나리오: 프로그램 시작 후, worker node는 모든 worker node와의 통신 환경을 초기화 후 가상의 system clock을 생성한다. System clock은 worker node에서 관리하며, 실제 시간이 아닌 operation에서 소요되는 가상의 연산 시간과 가상의 network delay를 측정한다. worker node는 연산에 사용될 1000x1000 크기의 행렬을 2개 생성 후 모든 요소에 무작위 난수를 저장한다. 그 다음 system clock 0sec에서 시작하며 master node는 모든

worker node에 행렬 연산 작업을 각 worker node가 가진 연산 queue에 최대로 분배한다. 이때 작업의 분배는 worker node의 처리 속도에 따라 동적으로 이루어진다. 작업의 분배는 최대한 공평하게 이루어져야 하고 분배에 대한 규칙 및 알고리즘 (전송할 행렬의 요소, 전송할 worker node의 선택 방식)은 제약이 없으며 제출 시 규칙 및 알고리즘에 대해 명시하여야 한다. 각 worker node는 전달받은 작업의 연산을 진행 후, 연산 결과를 master node에 전송하고 system clock을 업데이트 한다. 이때 모든 worker node는 연산 작업을 80%의 확률로 성공할 수 있다. 따라서 20%의 확률로 실패 시, master node는 실패되었다는 결과를 전달받아 다른 worker node로 재할당을 통해 장애 처리를 시도해야 한다. 위의 작업이 진행되는 동시에 모든 worker node들은 서로 간 통신을 이용하여 부하의 분산을 유도해야 한다. 즉, 모든 worker node는 연산 queue에 대기 중인 작업량이 공평하게 유지하도록 해야한다. worker node간 부하 분산에 사용되는 방법론 및 알고리즘 또한 제약이 없으며, 제출 시 사용된 알고리즘에 대해 명시하여야 한다. 전체 행렬의 곱셈 연산이 완료되면, Master node는 "worker node 별 작업 처리량, 각 node별 작업 수행시간 및 대기시간, 전체 연산의 수행시간을 log에 작성하고 master node는 행렬 3개 (곱셈에 사용될 행렬 2개, 결과 행렬 1개), 전체 수행시간을 log에 저장 후 모든 worker node와의 연결을 종료한다(모든 node는 Gracefully Termination 되어야 함). worker node에서 연산에 소요되는 가상의 시간은 1~3sec에서 작업마다 랜덤하게 설정하여야 하며, 모든 통신의 delay는 1sec로 지정한다. 프로그램 수행동안 모든 node들은 모든 event(연산 작업 분배, 작업 송수신 메시지, 연산 수행 내용, 결과 송수신 메시지, 장애 처리, 연산 부하 관리 등)를 각 system clock과 함께 log에 기록한다.

### 3. (과제제출) 다음 화일을 조별로 기한안에 제출한다.

- G조이름HW1.zip (ex. G1HW1.zip) //제출된 압축화일 문제시 0점처리
  - ◆ 모든 소스 화일들
  - ◆ AllDefinedLogs.txt
    - 조별로 프로그램에서 정의한 모든 Log 메시지 명세 및 설명
  - ◆ Node 별 출력된 모든 Log 화일들
  - ◆ download.txt
    - G조이름HW1.mp4 (ex. G1HW1.mp4) : 5분이내의 설명 동영상을 제작하여, 해당동영상을 다운로드 할 수 있는 link를 포함하는 화일

- 반드시 동영상이 다운되는 링크를 삽입할 것
  - 포함된 링크로 동영상 다운이 안되거나, 공유권한이 없거나, 다운 후 동영상 실행이 안되거나 하는 등의 에러는 프로그램 수행이나 포함내용의 미포함처럼 모두 해당조의 과실로 감점대상임
- ◆ Readme.txt
    - 조 이름, 모든 조원 학번&이름
      - 학생 별 역할 명시
    - 프로그램 구성요소 설명
    - 소스코드 컴파일방법 명시
      - 컴파일 및 실행이 되지 않으면 전체과제 점수 0점 처리
    - 프로그램 실행환경 및 실행방법 설명
    - 작업 분배 및 부하 분산에 사용한 알고리즘 설명 (장점 및 단점 분석)
    - Error or Additional Message Handling에 대한 사항 설명
    - Additional Comments: 추가로 과제제출관련 언급할 내용 작성