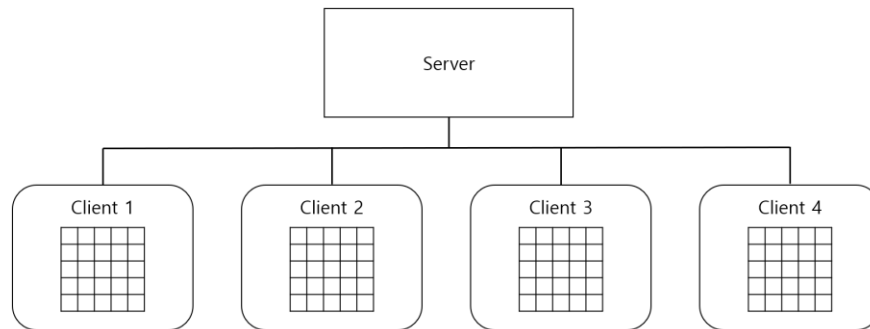


HW#2 데이터통신 프로그래밍 과제

서버-클라이언트간 Multi-Thread/Process 행렬 곱셈 계산 구현

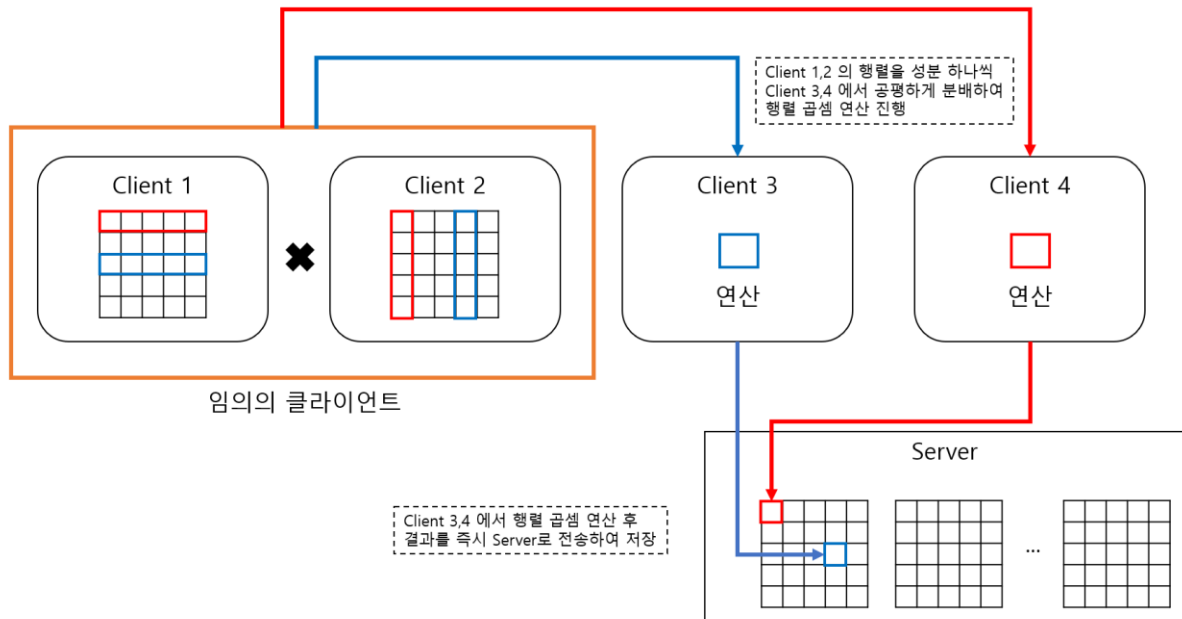
- DUE: 2023. 11. 14(화). 23:59 (NO Late Submission Allowed)
- How: 1개화일 제출 – G조이름HW2.zip (조별로 한명만 제출하면 됨)
 - ◆ 모든 소스화일 및 결과화일 전체를 압축한 G조이름HW2.zip으로 제출 ex) G1HW2.zip

1. (시뮬레이션 환경 및 구성요소) 다음과 같은 실험환경을 구성한다. 4개의 클라이언트와 1개의 서버는 thread/process간 Socket 통신으로 동작하며 연결 구조의 제한은 없다. 각각의 객체는 다음의 기능을 수행한다.



- ① 서버: 전역변수 혹은 임의의 객체인 System Clock을 유지, 관리, 업데이트 하고 프로그램이 수행되면 System Clock을 수행시킨다. (System Clock은 operation당 1sec씩 증가한다.) 프로그램이 시작되면 연산 결과를 저장할 10x10 크기의 행렬을 6개 생성한다. 전송받은 각 클라이언트 쌍의 연산 결과를 각각 6개의 행렬에 저장하며, 6개의 행렬의 연산이 끝날 때까지를 round 1이라고 한다. 해당 round가 끝난다면 새로 빈 행렬 6개를 생성하여 새로운 연산 결과를 저장한다. 동일한 과정을 round 100까지 진행하며, round 100이 끝났을 때 서버는 각 round별 총 연산에 소요된 시간 및 각 round의 연산 결과 행렬들을 출력 후 모든 클라이언트 connection을 종료한다. 서버는 모든 event 및 결과 정보(round별 결과행렬 및 소요시간)를 Log로 기록한다.
- ② 클라이언트: 각각의 클라이언트는 시작이후 10x10 크기의 무작위 정수기반(0~100사이) 행렬을 생성한다. 그 후, 아래의 그림과 같이 4개의 클라이언트 중 임의의 클라이언트 2개를 선택하여 해당 클라이언트가 보유한 행렬의 곱셈연산을 진행한다. 이때 곱셈 연산을 진행할 클라이언트는 선택되지 않은 나머지 클라이언트 2개에서 공평하게 분배하여

진행한다. 한번의 연산에 제공되는 정보는 성분 하나의 연산에 필요한 행과 열만 전송할 수 있으며 성분 하나의 연산이 끝난다면 그 즉시 서버로 결과를 전송하여 결과행렬을 update한다. round 1회에 임의의 클라이언트를 선택할 모든 경우의 수(6번)를 시행하며 모든 경우의 수의 연산을 동시에 진행한다. round가 끝난 후(모든 행렬연산 결과 종료시), 새로운 round가 시작하며 각 클라이언트는 새로운 행렬을 다시 생성한다. round 100이 끝났을 때 각각의 클라이언트는 각 round별 자신의 행렬 및 연산을 진행한 정보를 출력해야 하며, 각 클라이언트별 모든 event는 클라이언트 Log에 기록되어야 한다.



- ※ 서버는 반드시 AWS나 구글클라우드 등 Physically 외부서버에 구현 되어야함!!
 - ※ 서버와 모든 클라이언트의 통신은 반드시 Socket으로 구현되어야 함!!
 - ※ 모든 클라이언트는 Thread를 통해 구현되어야 함!!
- (해당 세가지 조건은 **과제수행의 전제조건**으로 만족되지 않으면 과제수행 인정불가)

2. (시뮬레이션 시나리오) 다음과 같은 실험 시나리오를 수행하여 각각의 구성요소 별 Log를 기록한다. 즉, 4개의 클라이언트와 서버는 각각 모든 이벤트와 연산 정보를 Client1.txt, Client2.txt, Client3.txt, Client4.txt, Server.txt에 기록한다.
 - ① 시나리오: 프로그램 시작 후, 서버는 연산 결과를 저장할 10x10 크기의 행렬 6개를 생성한다. 시작 직후 모든 클라이언트 및 round 1을 진행하며 System Clock 0sec에서 시작된

다. 4개의 클라이언트는 임의의 클라이언트 2개를 선택하여 쌍을 이루어 총 6가지 경우의 수의 행렬 곱셈 연산을 동시에 진행한다. 임의의 쌍을 이루고 있는 행렬 두개를 사용하고 곱셈 연산은 나머지 클라이언트 2개에 성분 하나씩 공평하게 분배하여 연산을 진행한다. 모든 경우의 수 연산이 끝난다면 새로운 행렬로 업데이트 하여 다음 round를 진행한다. 총 round 100까지 진행하며 round 100이 완료되면 모든 클라이언트는 보유한 행렬 및 연산 기록을 출력, 서버는 서버 종료 전 모든 round의 연산 결과 및 round별 소요 시간을 출력 후 모든 클라이언트와의 connection을 종료한다 (서버 및 클라이언트는 Gracefully Termination 되어야 함). 서버 및 클라이언트의 operation (각 요소별 메시지 전송, 행렬계산, 행렬 update 등)당 1sec 이외의 다른 delay는 없다고 가정하며, 구현하는 프로그래밍 언어의 제한은 없다. 프로그램 수행동안 각 요소들은 모든 event 및 결과 정보를 각각의 Log에 기록한다.

3. (과제제출) 다음화일을 조별로 기한안에 제출한다.

➤ G조이름HW2.zip (ex. G1HW2.zip)

- ◆ 모든 소스 파일들
- ◆ AllDefinedLogs.txt
 - 조별로 프로그램에서 정의한 모든 Log 메시지 명세 및 설명
- ◆ 서버 및 클라이언트별 출력된 모든 Log 파일들
- ◆ download.txt
 - G조이름HW2.mp4 (ex. G1HW2.mp4) : 5분이내의 설명 동영상을 제작하여, 해당동영상을 다운로드 할 수 있는 link를 포함하는 파일
 - 반드시 동영상이 다운되는 링크를 삽입할 것
 - 포함된 링크로 동영상 다운이 안되거나, 공유권한이 없거나, 다운후 동영상 실행이 안되거나 하는 등의 에러는 프로그램 수행이나 포함내용의 미포함처럼 모두 해당조의 과실로 감점대상임
- ◆ Readme.txt
 - 조 이름, 모든 조원 학번&이름
 - 학생별 역할 명시

- 프로그램 구성요소 설명
- 소스코드 컴파일방법 명시
- 프로그램 실행환경 및 실행방법 설명
- Server-Client 및 Client간 synchronization 및 serialization 수행방법 설명
(동영상 설명에서도 해당부분 설명 포함해야 함)
- Error or Additional Message Handling에 대한 사항 설명
- Additional Comments: 추가로 과제제출관련 언급할 내용 작성