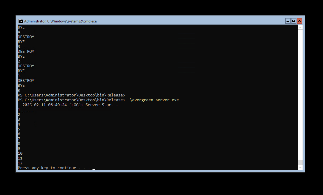
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | **작성자** | 2022180041 최재혁 2022180003 김도엽 2020184005 김나현 | **팀명** | zl존졸작전사 |
| **주차** | **1** | **기간** | **2025.01.13 ~ 2025.01.19** | **지도교수** | **정 내 훈**(서명) |
| **이번주 한일** | **최재혁: 아마존AWS 연동, Zero byte recv 도입 및 성능 프로파일링, 세션 및 소켓 풀링 적용**  **김도엽: 모션 블러 포스트프로세싱 ‘Reconstructor Filter’ 적용, ‘ImGui’를 이용한 디버깅 도입** | | | | |

**<상세 수행내용>**

최재혁

아마존 AWS 연동으로 24시간 서버 가동

페이지 락킹을 줄이기 위해 제로바이트 리시브 도입 및 프로파일링

원격 접속 테스트 컴퓨터 CPU:

13th Gen Intel(R) Core(TM) i7-13700H

기본 속도: 2.40GHz

코어: 14

논리 프로세서: 20

서버 컴퓨터 사양:

AMD Ryzen 5 3600X 6-Core Processor

기본 속도: 3.79GHz

코어: 6

논리 프로세서: 12

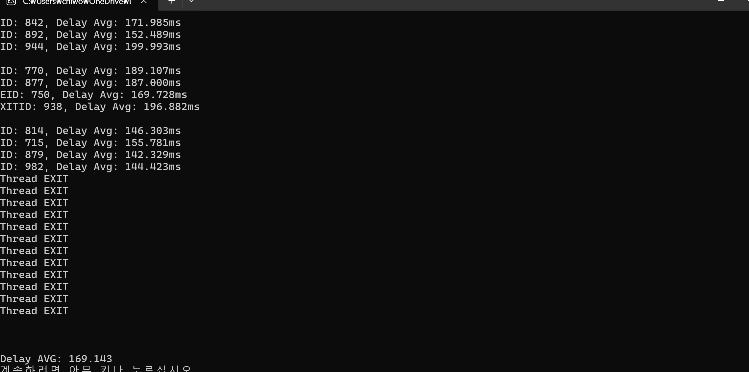
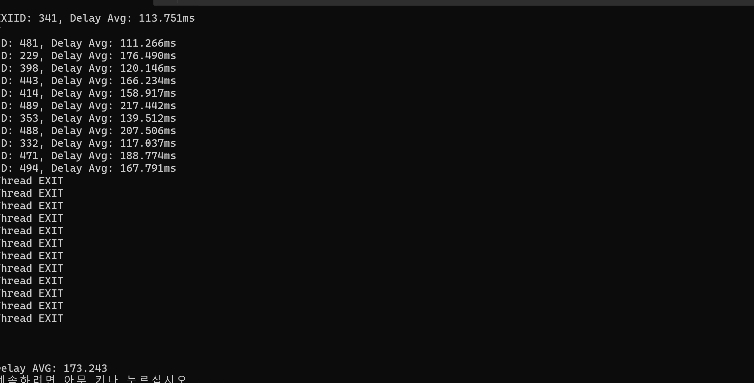
테스트 환경:

원격 – 다른 컴퓨터로 네트워크를 이용해 서버 컴퓨터로 300개의 더미 클라이언트 접속 + 본 클라

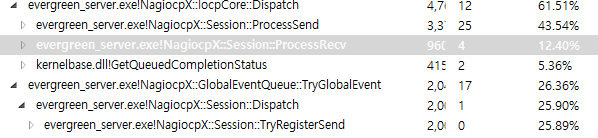
로컬 – 같은 컴퓨터에서 서버 가동 및 2000개의 더미 클라이언트 접속 + 본 클라

방법: 무브패킷에 타임스탬프를 넣어 서버에게 보낸 후 서버는 그걸 그대로 담아서 다시 클라이언트에게 전송, 클라이언트는 무브패킷 도착시 지금 시간에서 패킷에 담겨진 타임스탬프의 차이를 구함(무브패킷의 주기는 0.5초)

벤치마크 결과: (좌측이 제로바이트 리시브X, 우측이 제로바이트 리시브O)

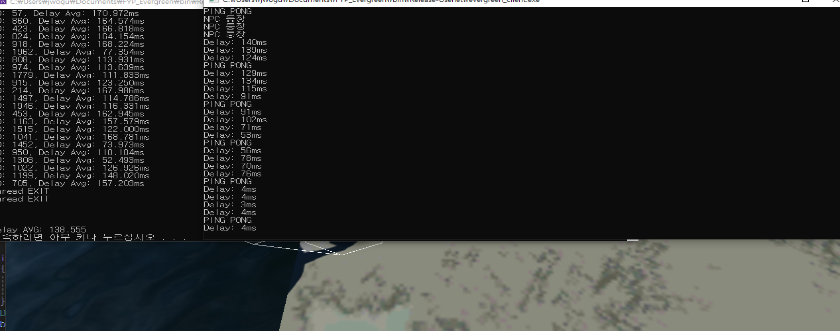
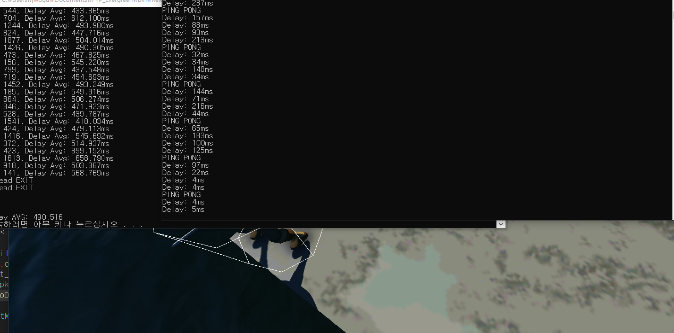


평균 173ms vs 169ms 원격 기준으로는 별 차이가 없다. -> 원인분석

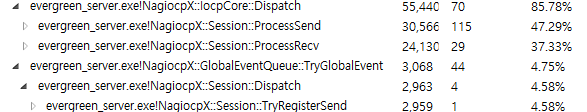


원격 테스트 기준 센드비중 43.54 + 25.89 / 리시브비중 12.4 -> 센드가 문제

로컬 테스트 결과 (더미클라이언트 2000개)



평균 490ms vs 138ms 약 4배 차이 -> 원인분석



로컬 테스트 기준 센드비중 47.29 + 4.58 / 리시브비중 37.33 리시브 지분이 커지니 유의미한 차이가 발생

결론: 센드 호출횟수를 줄여야 함 + 제로바이트 리시브 적용시 리시브 성능은 확실히 개선

(현재 지형데이터 미결정으로 인해 사실 상 N^2으로 Send중, 이 부분이 해결된다면 성능이 크게 오를 것으로 기대됨)

김도엽

* 모션 블러 렌더링 과정에 중 모션 벡터를 담은 G-버퍼에서 지배적인 모션 벡터를 구한 후, 후처리를 적용하는 ‘Reconstruction Filter’ 활용
* 피사체의 거리 차이에 따른 부자연스러운 모션 블러 효과 해결  
  〈’Reconstruction Filter’ 적용 전·후의 렌더링 화면 크롭, 좌: 적용 전, 우: 적용 후〉
* ‘ImGui’ 라이브러리 도입, 프레임 시간 Plotting 등 디버깅에 활용

김나현

|  |  |  |  |
| --- | --- | --- | --- |
| **문제점 정리** | 센드 호출 횟수, 몬스터 제대로 넣고 다시 비교필요, 세션소켓 풀링으로 인한 성능 테스트 필요 | **해결 방안** | 센드 호출 횟수는 거리 및 룩벡터 기반으로 많이 감소 시킬 예정 |
| **다음 주차** | **2** | **다음 기간** | **2025.01.20 ~ 2025.01.26** |
| **다음주 할 일** | 제안서에 작성한 내용대로, 서버단위 논타겟 시스템 추가예정 | | |
| **지도교수**  **Comment** |  | | |