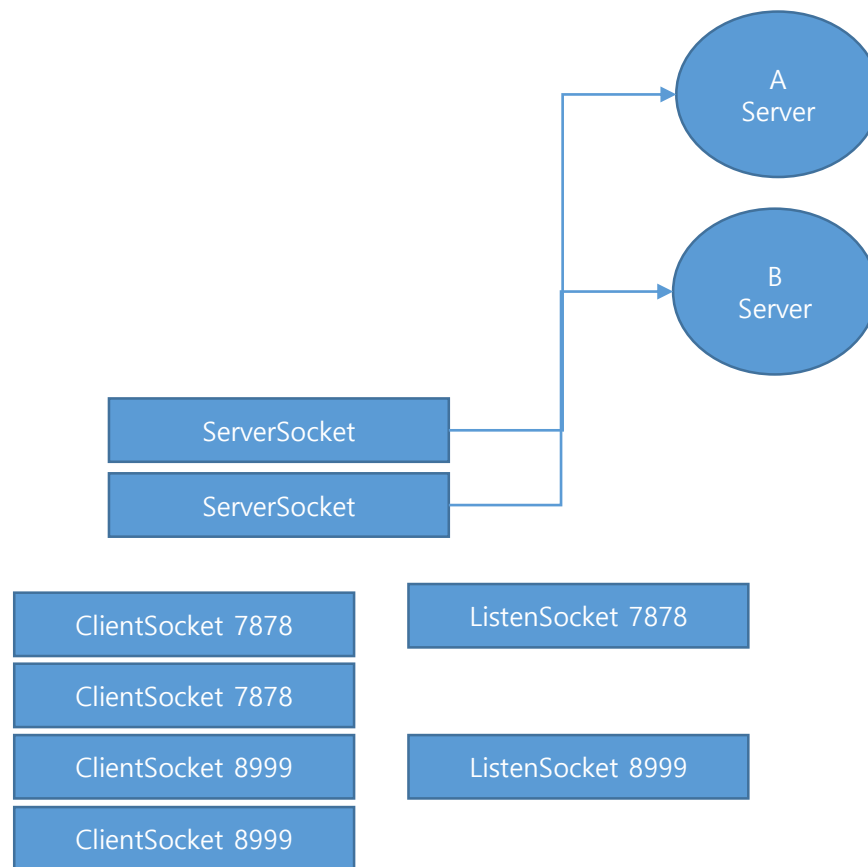
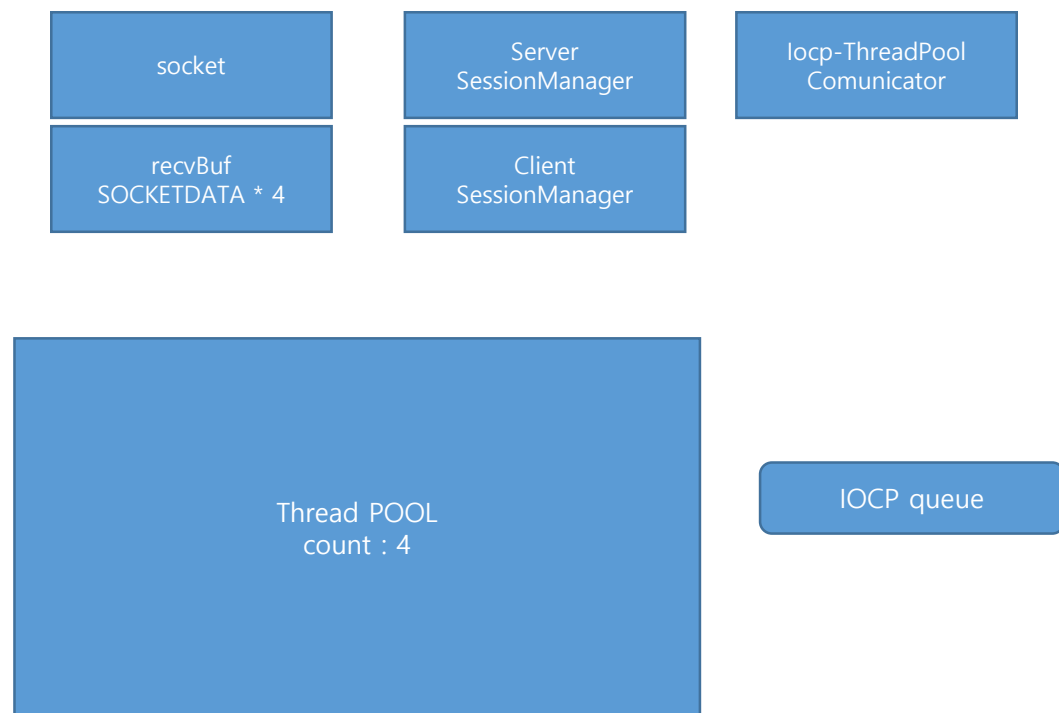


IOCP & EPOLL

아키텍처 by kim eun hye

IOCP 구조

- iocp 는 하나의 iocp 큐를 가지고 여러 소켓처리를 하도록 구성할수 있다.
- iocp 큐에 들어온 작업들은 OS 에서 자동으로 최적화된 쓰레드를 지정하여 매칭시킨다.
- 모든 소켓들을 하나의 IOCP 에 지정해 놓으면 나머지는 알아서 처리된다.



처리 매커니즘

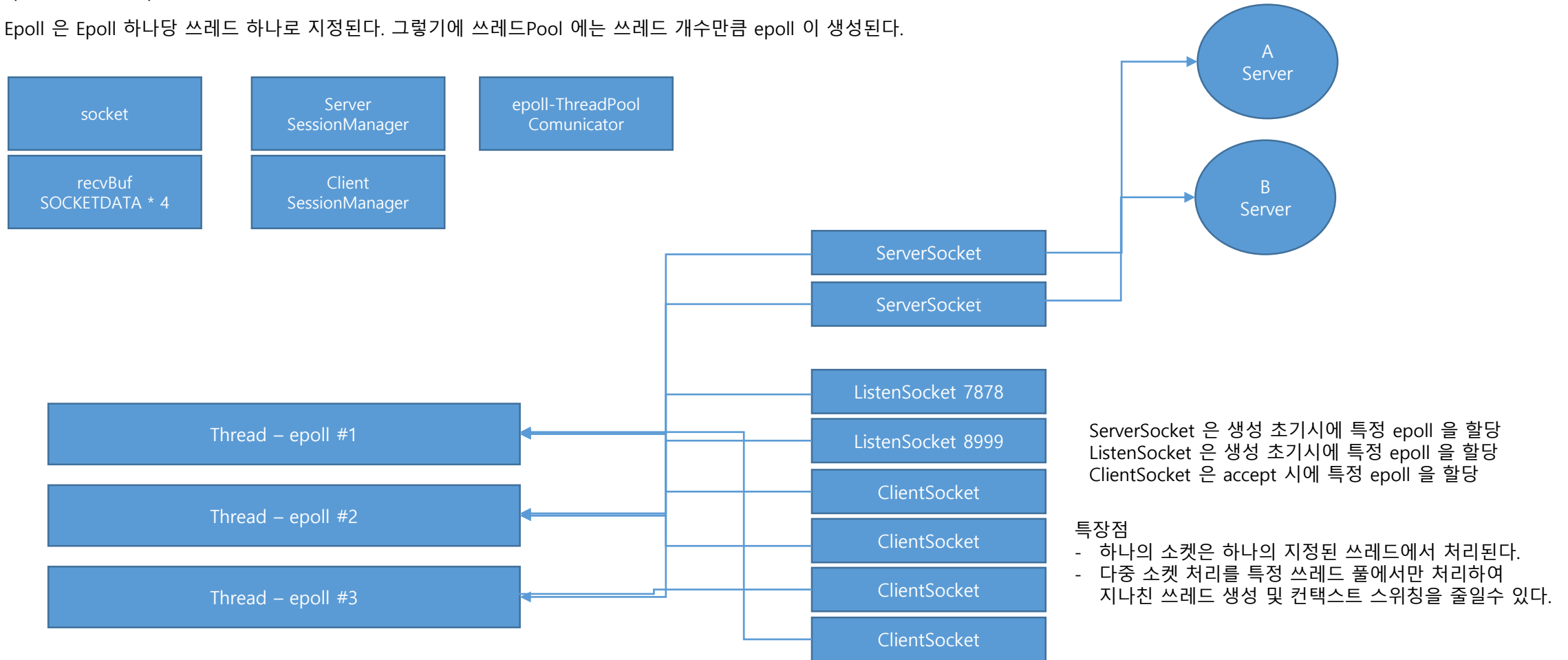
- 패킷 처리 함수를 포인터로 등록시켜 위의 thread 에서 처리한다.

장점

- 다중 소켓 처리를 특정 쓰레드 풀에서만 처리하여 지나친 쓰레드 생성 및 컨텍스트 스위칭을 줄일수 있다.
- OS 가 가장 최적화된 쓰레드 매칭을 하도록 설계되어 있다. (즉 가장 최근에 썼던 쓰레드를 계속 쓰는 경향)
- 다중 쓰레드지만, 현재 처리하고 있는 소켓의 Recv 처리가 다중으로 처리 되지 않는다. 처리 완료후 bindRecv() 되기 때문. (bindRecv 해야 OS 에서 데이터가 들어왔을때 recv 처리를 해준다.)
- 동일 소켓의 Recv 와 Send 는 다중 쓰레드에 의해 동시에 실행 가능.

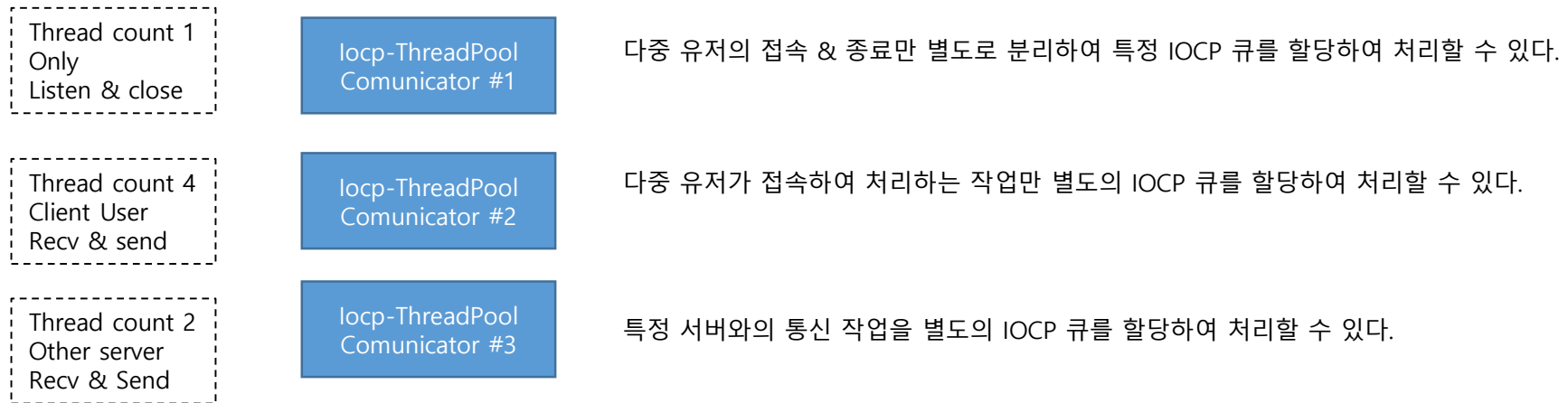
epoll 구조

- Epoll 구조는 소켓 하나당 또는 특정 이벤트처리에 대해 초기에 명확하게 epoll 을 지정해야 한다.
- Epoll 은 처음에 epoll 을 지정하는 로직이 추가로 들어간다.
- Epoll 은 Epoll 하나당 쓰레드 하나로 지정된다. 그렇기에 쓰레드Pool 에는 쓰레드 개수만큼 epoll 이 생성된다.



다중 IOCP 구조

- IOCP 큐 하나를 소켓 처리가 아닌 순수 이벤트 처리 쓰레드로 활용 가능하다. 이때는 SessionManager 를 사용안함으로 지정해야 한다.
- IOCP 큐를 작업별로 분할하여 역할을 쪼갤수 있다.



소켓 버퍼 전략

- RecvBuf , SendBuf 두개를 운영한다.
- IOCP 와 연동하는 소켓에 할당하는 recv 버퍼 사이즈는 패킷 길이에 따라 가변적으로 할당한다.
- RecvBuf : 가변적 할당은 더 큰 사이즈로 할당이 필요할 때만 변경된다. (기존 사이즈 버퍼 반환)
- SendBuf : Send 완료후 상시 반환. Overlapped 를 지원함

- 헤더파싱

1. 숫자 길이 파싱

- 숫자 byte 가변적 길이 INI 파일에서 읽어서 적용

2. Html 헤더 파싱 (이후 처리)

- CONTENTS_LENGTH, Chunk

3. 길이가 문자열 일때, 문자열 길이 헤더 파싱

- 뒤에 끝을 알리는 문자 INI 파일에서 읽어서 적용

※ 길이 Parsing 이후, 기존에 할당된 RecvBuf 사이즈가 부족하면, 반환하고 다시 할당한다.

```
typedef struct SOCKETDATA {
    STOVERLAPPED stCommon;
    CNWSocket* lpClient;
    DWORD nTotLen; // 총 목적 길이
    DWORD nCurLen; // 현재 받은/보낸 길이
    char *pData;
} SOCKETDATA, *LPSOCKETDATA;

LPSOCKETDATA m_lpRecvData;
LPSOCKETDATA m_lpSendData;
```

- HeaderPasing, RecvProc 은 함수 포인터를 사용하여 각 상황에 맞게 초기화시 대입한다.