

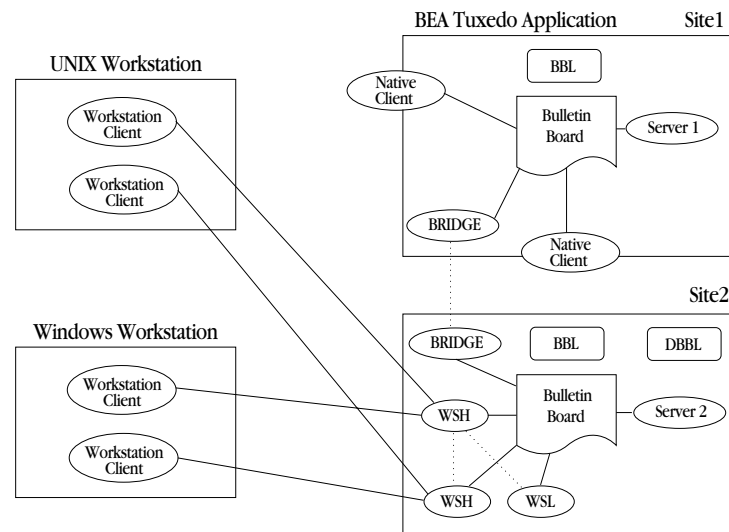
? TUXEDO - WSL 과 WSH 의 역할

제 품	TUXEDO
사용 OS	모든 플랫폼(platform)
작 성 일	2003/08/18
이슈사항	TUXEDO client 는 native client 와 /ws (Workstation) client 로 나뉘는데, native client 는 TUXEDO 환경에 직접 접속하여 서비스를 처리하는 반면 /ws client 는 WSL 과 WSH를 경유하여 서비스를 처리하게 된다. WSL 과 WSH 의 역할에 대해서 알아본다.

해 결 안

WS client 는 보통 TUXEDO 서버가 실행되는 기계와는 다른 기계에서 실행된다. PC 에 설치된 TUXEDO client 프로그램들이 모두 /ws client 들이다. TUXEDO 서버들은 유닉스 혹은 Windows NT 계열에서 message queue 라는 것을 통하여 서비스 요청을 받고 그 결과를 역시 message queue 를 통하여 반환한다.

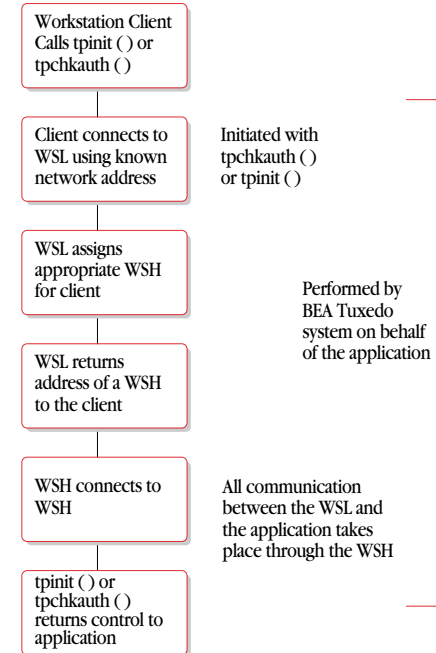
그런데, /ws client 들은 서버들과는 다른 기계에 있기 때문에 message queue 를 통하여 통신할 수는 없다. 그래서 사용되는 것이 WSL 과 WSH 이다. 간단하게 말하면 위의 두 프로세스는 TUXEDO 서버와 /ws client 들 간의 통신을 매개해주는 역할을 한다.



그렇다면 두개의 프로세스는 각각 어떤 역할을 담당하는가? /ws client 와 서버와의 통신은 크게 두가지로 나눌 수 있다. 하나는 초기 접속(tpinit)이고, 다른 하나는 서비스 수행(tpcall, tpcommit 등등)이다. 먼저 tpinit 에 대해 알아보자.

/ws client 가 서버에 접속하는 과정, 즉 tpinit 은 다음의 네단계를 거쳐 처리된다.

해 결 안

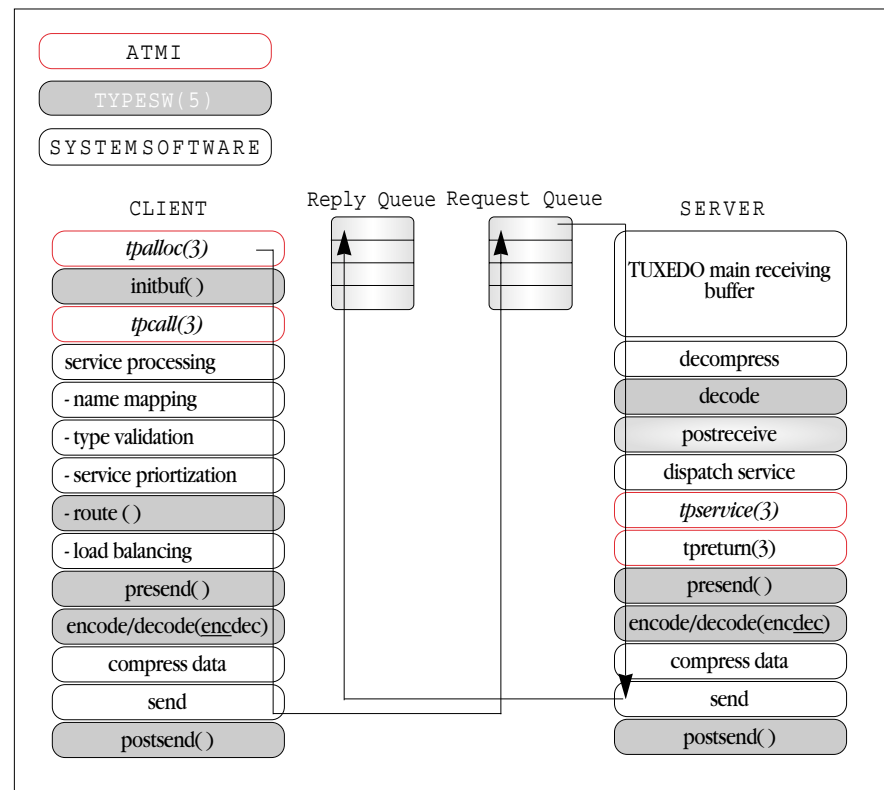


- 1단계 : /ws client WSL
/ws client 는 환경변수 WSNADDR 에 설정된 주소와 포트 번호에 해당하는 WSL 에 socket 접속을 하여 서버와의 접속을 요청한다.
- 2단계 : WSL → /ws client, WSH
WSL 은 자신이 관리하고 있는 WSH 중 하나를 선택하여 그 포트 번호를 /ws client 에게 전송하고 해당 WSH에게 /ws client 가 접속해 올 것이라고 알린다. 이 과정에서 WSL 의 옵션과 현재상황 (WSH 수와 접속된 /ws client 수)에따라 WSH 를 더 띄우기도 하고 허용된 연결 수를 초과한 경우에는 에러를 반환한다. tpinit 과정에서 WSL의 역할은 여기까지이며 WSL 과 /ws client 와의 socket 연결은 끊어진다.
- 3단계 : /ws client → WSH
/ws client 는 WSL로부터 지정받은 WSH 로 socket 연결을 하여 서버 접속에 필요한 인증정보, 즉 username, clname, password 등을 전송한다.
- 4단계 : WSH → /ws client
WSH 는 접속해온 /ws client 의 인증정보를 확인하고 TUXEDO 환경에 해당 /ws client 를 등록한다. 물론, 인증정보가 틀리면 접속을 거부한다.

해 결 안

이러한 과정을 거쳐서 접속이 이루어지게 되면 3단계에서 맺어진 socket 연결은 tpinit 과정뿐만 아니라 /ws client 가 종료할 때까지, 즉 /ws client 가 tpterm 을 호출하거나 WSH가 요청이 없는 /ws client 연결을 끊을때까지(WSL의 -T 옵션 참조) 계속 연결되어 이후의 서비스 처리(tpcall, tpcommit 등)에 사용된다.

다음은 실제 서비스 처리과정에서의 WSH 의 역할을 보자. WSL은 실제 서비스 처리과정에는 관여하지 않는다. client 의 요청 중 대부분을 차지하는 tpcall 의 처리흐름에 대해서 살펴보겠다.



위의 그림에서는 native client의 경우를 보이고 있다. /ws client의 경우에는 WSH가 /ws client를 대행하여 위와 비슷한 과정을 수행한다.

1단계 : /ws client → WSH (socket 통신)
 client 에서 tpcall 을 이용하여 서비스 요청을 하게되면 TUXEDO 서버로 보내지게 되는데 이 서비스 요청을 받아들이는 것이 WSH 이다. 위의 tpinit 3단계에서 맺어진 socket 연결을 사용한다.

해 결 안

2단계 : WSH → AP Server 의 request queue (message queue 통신)
 WSH는 요청받은 서비스 이름을 TUXEDO 환경(Bulletin Board)에서 찾아본다. 해당 이름으로 등록된 서비스가 없으면 에러(tperno = 6)를 반환한다(아래 5단계로 간다). 등록된 서비스가 하나만 있으면 해당 서버로 보내고 여러 개가 있는 경우는 load balancing 이 이루어진다(load balancing 에 대해서는 생략하겠다). 서버로 보낸다는 것은 해당 서버가 가지고 있는 request queue 에 넣는다는 의미이다.

3단계 : AP Server(서비스 처리)
 AP Server 즉 실제 업무 서비스를 처리하는 서버는 자신의 request queue에 서비스 요청이 들어오기를 기다리다가 queue에 들어온 요청이 있으면 꺼내어 처리를 한다. 당연한 말이지만 이 요청 안에는 서비스 처리 결과를 반환할 곳(해당 WSH 와 /ws client 식별자)이 명시되어있다.

4단계 : AP Server → WSH 의 reply queue (message queue 통신)
 AP Server 는 서비스 처리가 끝난 후에 결과를 반환하는데 지금의 경우는 해당 WSH 의 reply queue 로 결과메시지를 넣게 된다.

5단계 : WSH → /ws client (socket 통신)
 WSH는 자신의 reply queue 에 서비스 처리 결과 메시지가 들어오면 그 것이 어느 /ws client 가 호출한 것이지를 식별하여 해당 /ws client 에게 결과를 보낸다. 역시 위의 socket 연결을 계속 사용한다.

위의 2단계와 4단계는 상황에 따라 다를 수 있는데, 예를 들어 서버 기계가 여러 개이고 요청한 서비스가 WSH 가 있는 기계(A)가 아니라 다른 기계(B)에 있는 경우 2단계는 간단히 설명하면 다음과 같이 된다. WSH 는 A 기계에 있는 BRIDGE(A)로 보내고(message queue 통신), BRIDGE(A)는 B 기계에 있는 BRIDGE(B)로 보내고 (socket 통신), BRIDGE(B)가 해당 AP Server 로 보내게 된다(message queue 통신). 그리고, 4단계는 위의 과정의 역순이 된다. 또, 도메인을 사용하는 경우는 BRIDGE 대신 도메인 게이트웨이를 통해서 통신이 이루어진다.

WSL 과 WSH 의 역할을 살펴보았는데, 간단히 요약하면 WSL 은 초기 접속(tpinit)시에 WSH 를 할당해주는 역할을 하고, WSH 는 /ws client 의 인증과정과 이후의 서비스 요청을 처리하는 역할을 한다.

추가정보