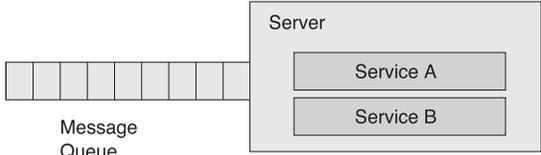
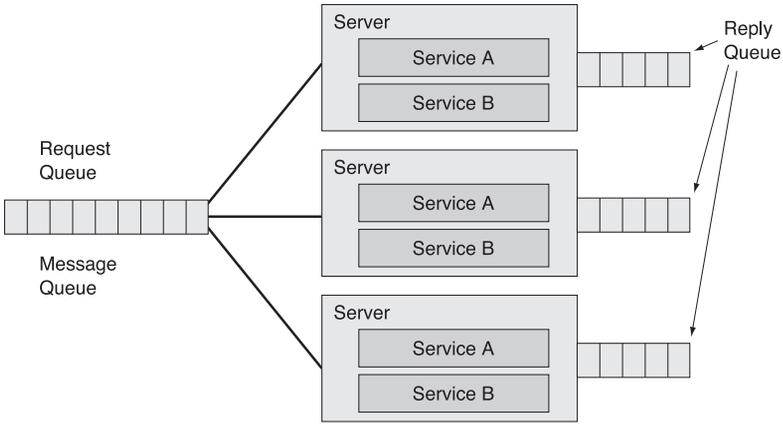
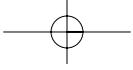




MSSQ의 구조 및 사용기준에 대해

제 품	Tuxedo(ALL Ver.)
사용 OS	모든 플랫폼(platform)
이슈사항	MSSQ 의 구조 및 사용기준에 대해서 알아본다.
해 결 안	<p>MSSQ는 텍시도에서 여러 개의 애플리케이션의 request Q를 하나로 통합하여 Q의 사용 효율성을 높이고 효과적인 부하분산을 하고자 하는데 그 목적이 있다.</p> <p>아래는 일반적인 Tuxedo application 서버의 Message queue 형태를 간단하게 나타낸 그림이다. 하나의 서버에서는 Request, Response 둘 다 처리할 수 있는 Message Queue가 하나가 존재한다.</p>  <p>그러나 아래와 같이 MSSQ를 사용할 경우는 아래 그림과 같다.</p> <p>아래 그림을 보면 Request Queue에 동일한 서비스를 가지고 있는 3개의 서버가 접근하는 것을 알 수 있다. 결국 이런 형태로 큐에 데이터가 도착할 경우 각각의 서버가 데이터를 가져가게 된다.</p> 



TUXEDO

Technical Tips

해결안

< MSSQ 사용 선택기준 >

아래와 같은 상황에 따라서 사용 여부를 선택해야 한다.

1. MSSQ를 사용할 수 있는 경우
 - 같은 MSSQ를 사용하는 모든 서버의 서비스가 동일해야 한다.
2. MSSQ를 사용하면 좋은 경우
 - 서버가 2개~12개 사이인 경우
 - 메시지의 크기가 비슷한 경우
3. MSSQ를 사용하지 말아야 할 경우
 - 하나의 MSSQ를 사용하는 서버가 12개를 넘을 경우에는 MSSQ를 분리
 - 메시지가 너무 큰 경우
 - 각각의 서버의 메시지의 크기가 상이하게 다른 경우

이러한 MSSQ 설정 시 각 Service간의 호출이 있고, 호출하는 쪽 (Caller)가 MSSQ를 사용할 때에는 REPLY_Q를 설정해야 한다. (위의 그림에서 Reply Queue)

예를 들어 서비스 A에서 서비스 B를 호출할 때, 서비스 B가 호출이 작업이 끝나면 그 내용을 서비스 A의 서버의 request Q로 return한다. 이때 이 서버가 MSSQ를 사용하고 있다면 여러 서버가 이 MSSQ를 공유하고 있기 때문에, Caller 서버가 아닌 전혀 엉뚱한 서버로 갈 수가 있다. 이런 상황을 막기 위해서 caller 쪽에 REPLYQ를 따로 지정해서 이런 현상을 방지하게 된다.

추가정보

