목차

Message Queue 개념 정리 3
첫번째 의미 3
두번째 의미 3
장점 3
단점 3
사용을 고려해볼 수 있는 상황 3

Message Queue 개념 정리

첫번째 의미

메세지 큐의 한가지 의미는 IPC(Inter-Process Communication) 도구 중 하나로 시스템 프로그래밍에서 사용됨. 이는 프로세스들 간에 이산적인 양의 데이터 송수신을 위해 사용하는데, 각 데이터를 메시지 형태로 생성, 전달하고 수신이 가능하게끔 Queue 데이터구조를 이용. 메세지 큐는 커널에서 전역적으로 관리되고, 모든 프로세스에서 접근 가능하도록 구성되어있으므로 하나의 메시지큐 서버가 커널에 요청해서 메시지큐를 작성하게 되면 해당 식별자를 아는 모든 프로세서가 동일한 메시지큐에 접근하여 메세지를 공유. 리눅스, 윈도우 등 운영체제와 커널 종류에 따라서 고유한 메세지큐를 제공함.

두번째 의미

메세지 큐의 또 다른 의미는 분산 메시징 시스템. 모니터링, 로그, 이벤트 메세지 등 아주 거대한 양의 데이터를 다룰 일이 생김에 따라 만들어짐. 큐는 여러 컴포넌트들이 데이터를 생성하고 또 저장하고 사용하는 과정에서 일종의 버퍼 역할을 수행함. 큐가 있음으로 producer과 consumer간의 속도가 다를 때, 둘 중 어느 한 컴포넌트가 네트워크에 연결되지 못한 상황 등에 대응이 가능. 기본적으로 분산 메시징 시스템은 어떤 특정 메시지가 최소한 한번은 전달될 수 있도록 보장하며 다수의 reader와 writer을 하나의 큐를 통해서 공유하고 상호작용할 수 있게끔 만듬.

장점

장점	설명
확장성	기능 별로 모듈을 분리하여 구성하기 때문에 확장에 용이
데이터의 영속성	데이터를 메모리 대신에 디스크에 저장하여 데이터 유실을 방지
고가용성 유지	다소 지연시간을 희생하는 한이 있더라도 서비스 규모에 독립적으로 언제나 이용할 수 있는 상태를 유지
장애 복원력	단일 장애점 해소. 시스템 일부에 장애가 발생하더라도 전체적으로 영향을 주지 않음. 메세지가 큐에 추가되어있으므로 혹시라도 어떤 모듈에서 장애가 발생하였더라도 해당 모듈 복구 후 작업을 마저 수행하는 것이 가능.

단점

큐를 운영하기 위한 추가적인 자원 필요, 큐에 들어가고 나오는 과정에서 피할 수 없는 오버헤드가 생겨남.

사용을 고려해볼 수 있는 상황

안정성, 고가용성을 위한 설계(즉각적인 업데이트는 보장하지 않음. 시스템에 확장성 필요. 데이터의 영속성 필요. 장애에 대한 복원력이 필요.) 특정한 시간에만 tps가 튀는 등의 사례. 추가적으로 장비를 투입하는 것보다 효율적인 방안이 필요한 경우 같은 IDC 안에서 메세지가 전송되는 등의 조건 하에 HTTP 프로토콜의 오버헤드가 큐를 사용할 때 증가하는 네트워크 레이턴시보다 커지는 경우 사용할 수 있는 자원에 비해서 처리해야될 데이터 양이 지나치게 많아 대응이 필요할 때 데이터가 생성되는 양에 비해서 소모되는 양이 눈에 띄게 차이나는 경우 다수의 어플리케이션이 어떤 데이터 저장소를 공유할 필요성이 있을 때

From:
https://atl.kr/dokuwiki/ - **AllThatLinux!**

Permanent link:
https://atl.kr/dokuwiki/doku.php/message_queue_%EA%B0%9C%EB%85%90_%EC%A0%95%EB%A6%AC

Last update: **2018/09/11 01:24**