

MongoShake 사용시 생기는 문제점과 해결 전략 수립

Mongoshake 를 사용할 시에 매우 쉽게 양방향 동기화를 사용할 수 있다.

하지만 개인적인 생각으로 고려해보아야할 것이 많다고 생각한다

첫 번째 고려 사항 → CPU 사용량 및 서버 부하

하지만 실시간 동기화에 사용되는 CPU 사용량은 무시할 수 없을 것이다.

또한 Shard 를 추가해야 하거나 삭제하여야 할 때 생기는 서버의 부하와 mongos(상대적으로 작긴 하지만) 서버 프로세스 , Mongoshake 를 통한 실시간 동기화 까지

동시에 이루어져야 한다면.. 절대 무시 못할 문제가 생길 수 있을 것이다.

Mongoshake 는 공식문서를 통해 간단한 해결 방안을 제공한다.

mongoshake 는 간단한 네트워킹 세팅과 config 파일 수정을 통해 mongoshake 용 서버를 따로 둘 수 있다. 해당 방법을 사용하면 서버의 부하를 줄일 수 있다.

두 번째 고려 사항 → Shard 추가 및 삭제 시 mongoshake 를 사용할 수 없다.

MongoDB 서버를 운영하다보면 아무래도 수직적인 확장이 아닌 수평적인 확장이 더 끌릴 것은 틀림 없다. (애초에 특화된 DBMS 이니까 ?)

하지만 mongoshake 를 통한 실시간 동기화의 한계는 sharding 진행시 chunk 들의 분배를 결정하는 Balancer 를 disable 해야지만 가능하다.

결국 동기화와 sharding 은 동시에 진행 될 수 없다는 것이다. (테스트를 통해 확인했다.)

DR 을 위해 사용한 mongoshake 이지만 이는 분명 한계가 있다. 이 때 필요한 것이 mongodump, file capture 등을 통한 백업이다.

백업 전략을 다양하게 짜놓을 수 있다면 실시간 동기화에 끊임이 생기더라도 그 공백을 메꾸기 어렵지 않을 것이다.

세 번째 고려 사항 → mongoshake...?

mongoshake 는 중국의 알리바바 사가 서비스를 만들고 github 를 통해 배포하고 있다.

물론 최근 릴리즈가 한 달 전일 만큼 활발히 사용되는 mongoDB tool 임은 틀림없다.

하지만 mongoDB 회사가 공식적으로 지원하는 tool 은 아직 아니다. 그에 따라 안정성에 의문이 생겼다.

또한 자료가 매우 적고, 중국 쪽 자료가 그나마 있는 편이다.

개발? 구성? 사용? 표현을 어떻게 해야할 지 모르겠지만 간단하지만은 않을 것이라는 거다.

이것으로 mongoshake 를 사용한 global DR 전략 스터디를 마치고 mongodump 를 사용한 백업 전략으로 넘어가 볼 까 한다.

읽어주셔서 감사합니다