**Database Proxy Server Architecture**

|  |  |  |  |
| --- | --- | --- | --- |
| **리비전** | **날짜** | **작성자** | **내용** |
| 0.1 | 2010-12-06 | 김병수 | 초안 작성 |

1. **목적**
   1. DB 요청/응답을 비동기적으로 처리하여 DB의 부하를 줄인다.
   2. DB cache 기능을 제공하여 DB 부하를 줄이고 응답 속도를 높인다.
2. **기능**
   1. In-process 및 Remote server를 지원하며, 각각 동기, 비동기 처리를 지원한다.

|  |  |  |
| --- | --- | --- |
|  | 동기 | 비동기 |
| In-process | Proxy 기능을 사용하지 않을 경우 |  |
| Remote | 설정 값 등을 읽어올 경우 | 일반적인 경우 |

* 1. Inproc/비동기는 차후에 필요 여부를 결정하여 개발한다.
  2. 쿼리의 프로파일링(profiling) 정보를 로깅(logging)한다.

1. **구조**
2. TODO:
3. **개발 기간**
   1. 분석 및 설계: 2~3일
   2. In-process/동기: 약 1주일
   3. Remote/동기: 약 1주일
   4. Remote/비동기: 약 2주일
   5. DB cache: 미정
4. **개발/서비스 환경**
5. 플랫폼: Windows 200X Server x64 or x86(x32)
6. 데이터베이스: MS SQL Server 2005/2008 or Oracle 11g/10g
7. 컴파일러: Visual C++ 9.0(Visual Studio 2008)
8. **데이터 구조 및 알고리즘**
   1. ADO를 이용하여 DBMS 구분없이 동작하도록 한다.
   2. DBMS 별 데이터 타입(특히, ADO 타입)에 상관 없이 동작하도록 한다.
   3. LOB 타입은 지원하지 않는다.
9. **제약 사항**
   1. 사용자 별로 처리 순서가 보장되어야 한다.
   2. 특정 사용자에 대한 처리가 실패할 경우 해당 사용자와 관련된 모든 처리는 보류되어야 한다. (TODO: 좀 더 고려해야 됨)
10. **장애 대처**
    1. DPS가 다운됬을 경우
       1. 각 요청자는 요청 정보를 캐싱하여 일정 시간이 되기 전까지 재요청을 시도하며, 시간이 초과하였을 때 장애로 처리한다.
    2. 응답이 기대되는 결과가 아닐 경우
       1. 로그를 남기고 해당 사용자의 접속을 강제로 해제한다.
11. **위험 요소**
    1. TODO:
12. 관련 자료
    1. <http://www.gpgstudy.com/forum/viewtopic.php?t=3188>
    2. <http://cafe.naver.com/ongameserver/477>