MOAZIO

게임 서버 부하테스트 및 최적화

수행 내역서

1. 현황

1.1 목표 성능

✓ DAU 10만 (동시접속자 1만 수준)

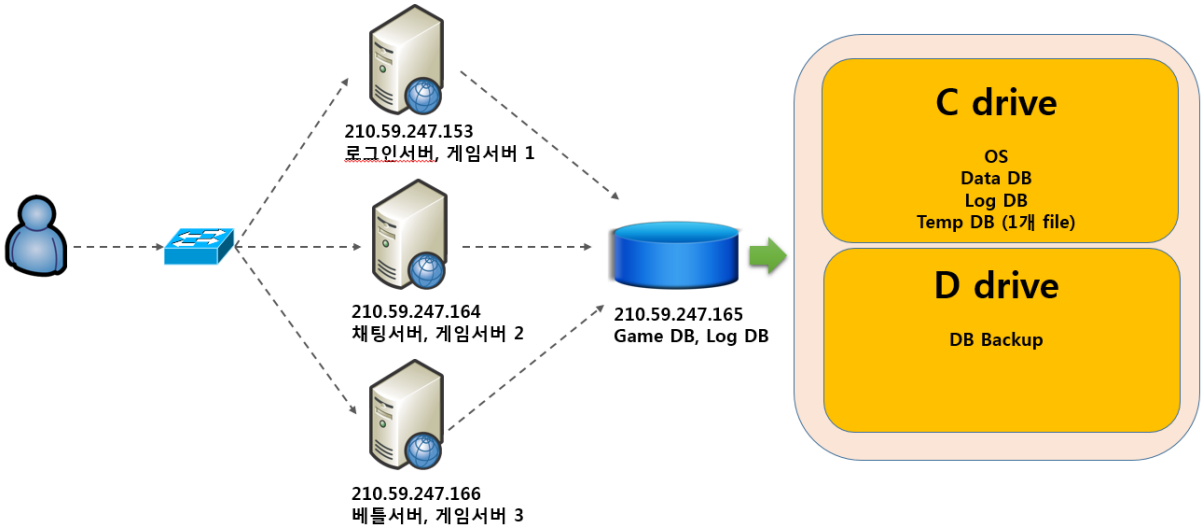
1.2 시스템 현황

✓ 총 4대의 서버 사용 (게임서버 3대, DB 1대)

✓ 특이사항

- 한대의 장비에 Game DB, Log DB 가 통합되어 있음. 향후 트래픽 대비를 위해서는 각각 별도 장비로 분리 필요.

✓ 현 시스템 구성도



1.3 기타

- SQL Injection 방지 필요

1.4 요구사항 분석

1.4.1 성능의 이슈

■ DB 에 대한 성능 의존도가 높은 게임

■ 전체 성능의 향상을 위해서는 DB 에 대한 최적화 필요

1.4.2 DB 구성 최적화 및 제안 필요

■ 현재 기본 구성된 DB 는 설정 및 구성이 Live 서비스에는 문제가 있음. 이에 대한 최적화 및 제안 필요.

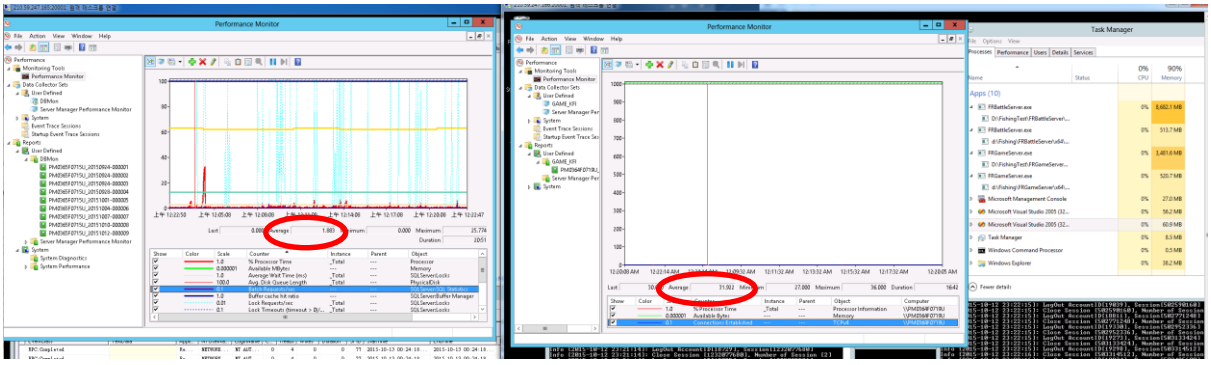
2. 설계

2.1 성능 목표 설정

➢ TCP 서버의 특성상 측정가능한 객관적인 지표로 DB 의 Batch Request 를 선정.

2.2 성능 목표 산정

A. 현재 서비스 상에서 커넥션 31에 초당 1.8 개의 Batch Request 가 발생하고 있음. 서버당 2개의 데몬이 서비스 중인 것을 감안하여 사용자 31명에 초당 3개의 Batch Request 가 발생될 것으로 예상.



B. DAU 10만명시 예상 최대 동접은 1~2만. 목표 최대 동접은 2만으로 설정.

➢ **위 A, B 두 수치를 근거로 목표 Batch Request 산정**

■ 최대 목표 동접 2만, 31명당 3개의 초당 Batch Request

■ **목표 초당 Batch Request = (20,000 / 31) * 3 = 1935**

2.3 성능 목표 수립

2.3.1 성능 목표 수립 (DAU 10만)

■ 성능 현황

Batch Request	최대 동접 / MAX DAU
32	310명 / 1550 명

■ **성능 목표**

Batch Request	최대 동접 / MAX DAU
1935	20,000명 / 100,000 명

3. 현 시스템 성능 분석

3.1 시스템 분석

3.1.1 현재 시스템의 Batch Request

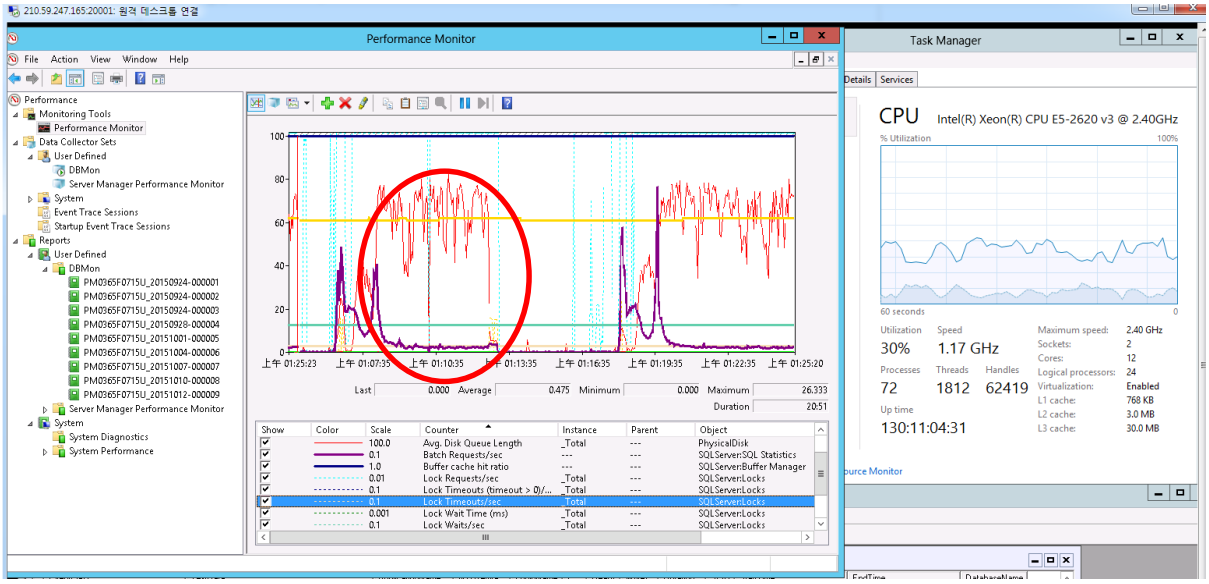
현재 시스템의 초당 처리 가능 Batch Request 는 최대 1000 이하 평균 30

A) 아래의 그래프와 같이 400개의 봇 클라이언트로 최초 부하를 주었을 때 Batch Request 가 500 까지 급증 후 수분 이후부터 20~30 으로 급락함. 즉, 대부분의 클라이언트가 응답을 받지 못하고 대기하고 있는 상태.

B) Batch Request 30 에서도 CPU 는 40% 정도를 유지

C) 20~30 Batch Requset 가 발생한 구간 : SQL Lock 개선과 튜닝이 필요

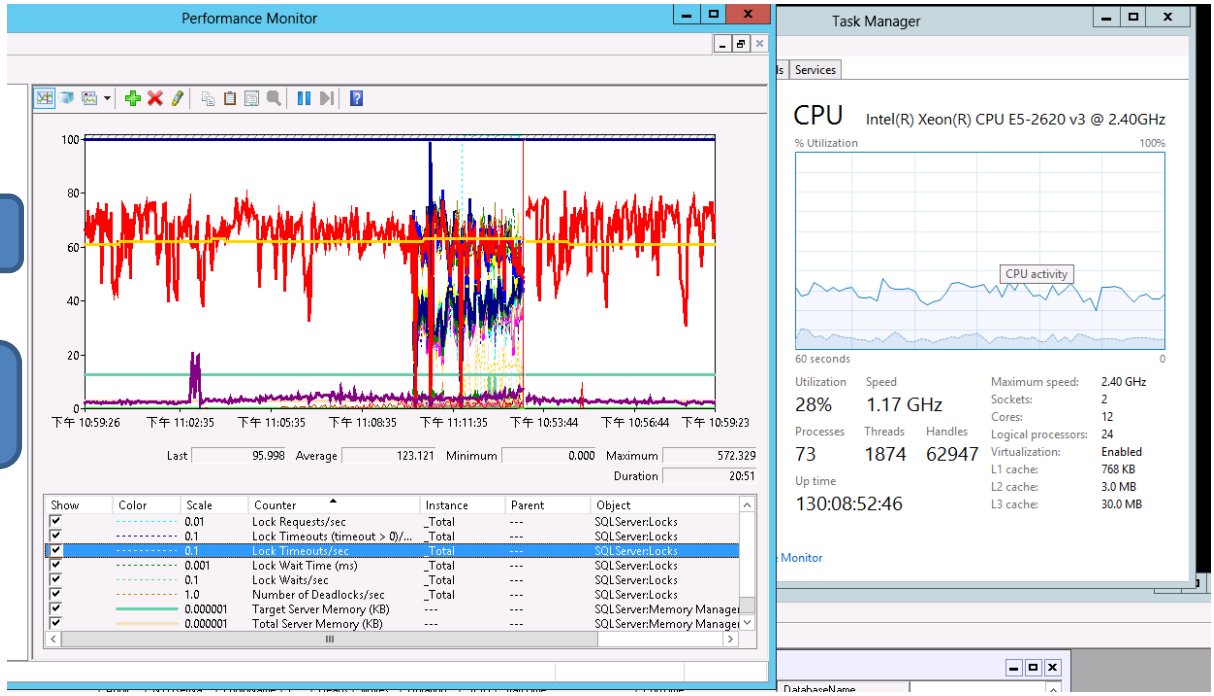
D) 500 Batch Request 가 발생한 구간 : 튜닝이 필요한 상태



E) 봇 클라이언트 개수를 늘려 보았으나 처리량이 늘어나지 않음.

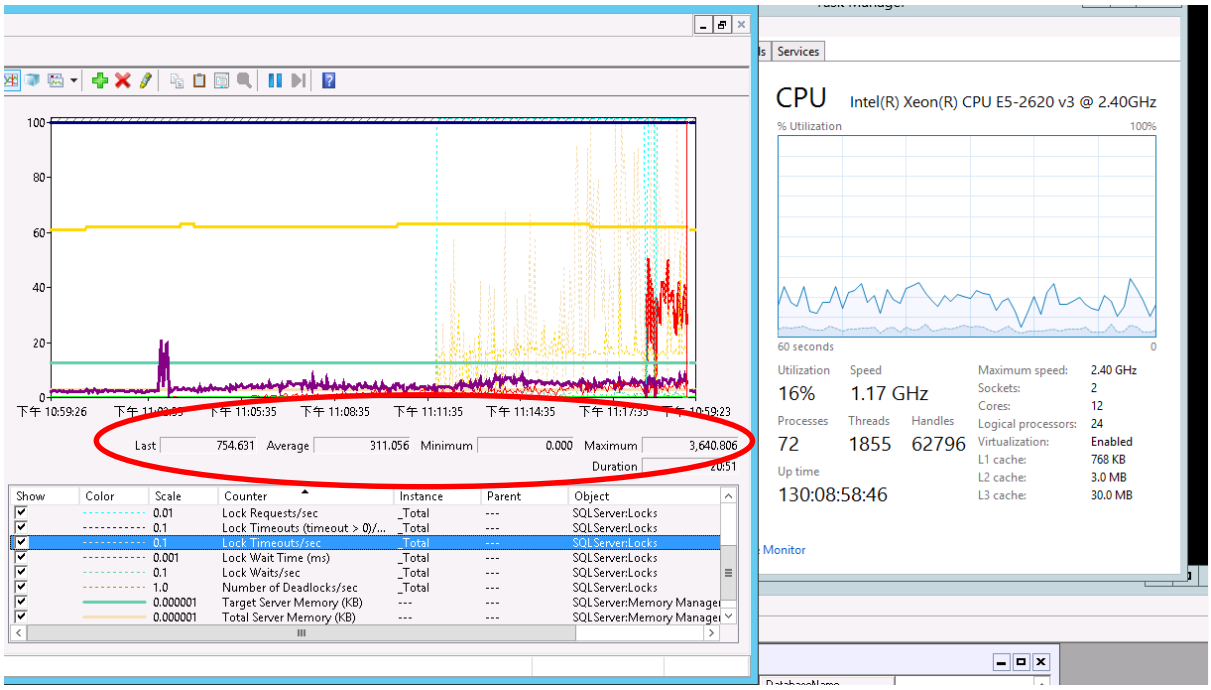
CPU

Batch Request



F) Lock 으로 인한 TimeOut 이 초당 평균 300여개 정도 발생함

➢ 클라이언트에 서버 접속 오류 등이 다수 발생되는 장애 상황

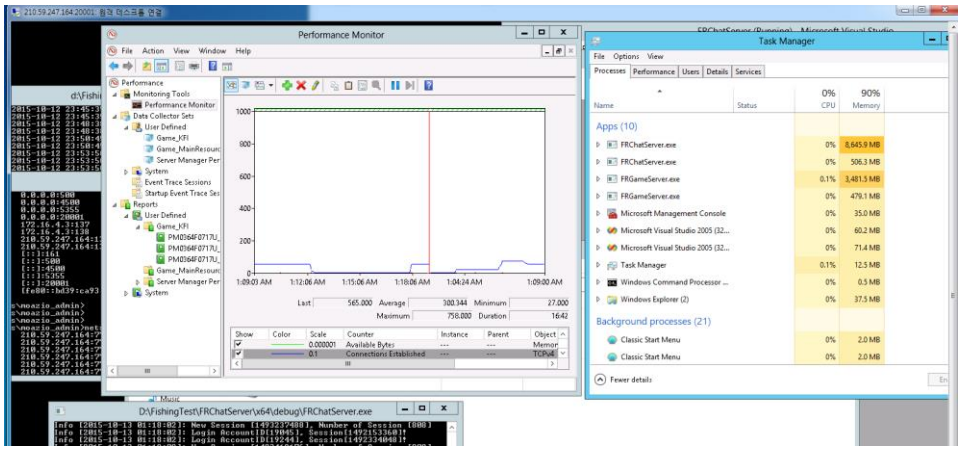


3.1.2 게임서버 장비 상황 분석

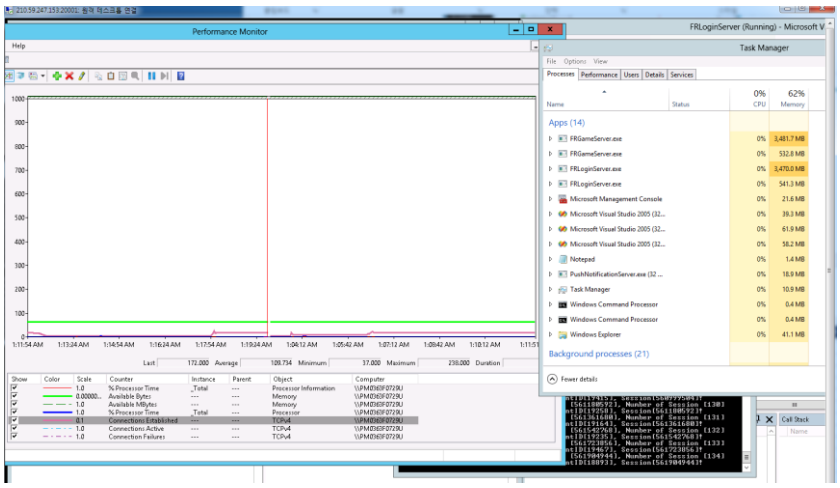
해당 시점에 게임서버 3대의 부하는 거의 없음.

해당 게임의 전체 성능은 DB에서 결정되는 것으로 보임.

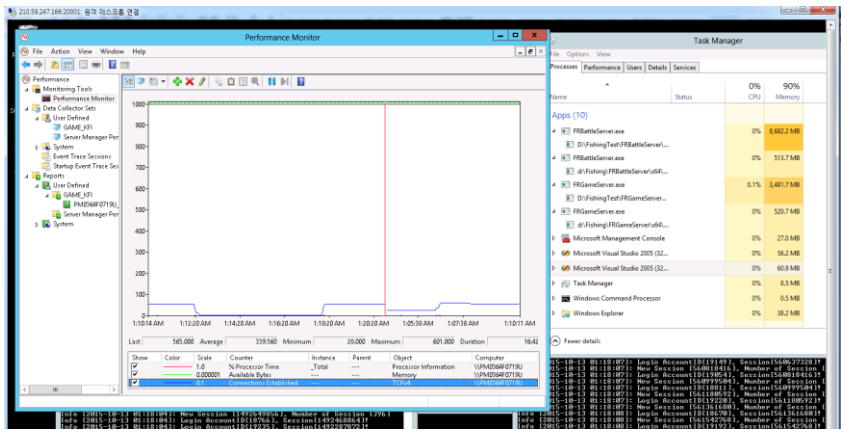
A. 210.59.247.164 (채팅, 게임서버2)



B. 210.59.247.153 (로그인, 게임서버1)



C. 210.59.247.166 (베틀, 게임서버3)



3.2 최적화

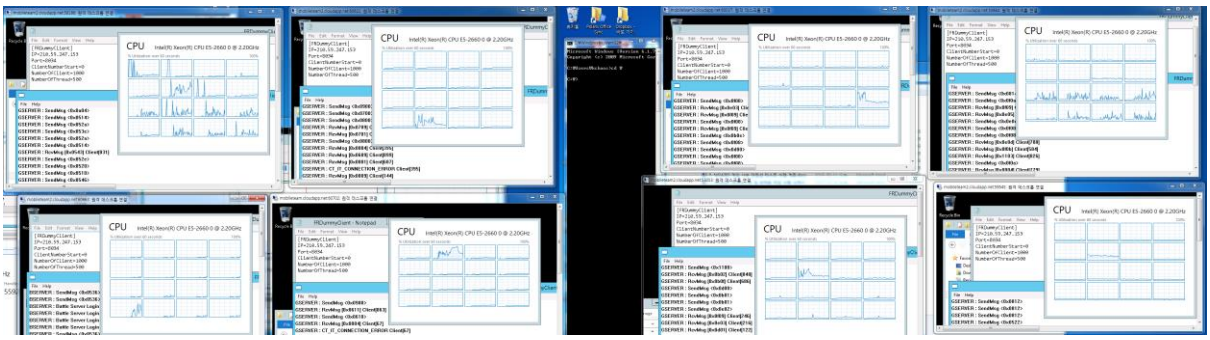
해당 게임의 전체 성능을 올리고 안정성을 확보하기 위해서는 MS-SQL 에 대한 최적화 작업에 집중하는 것이 필요한 것으로 판단되어 관련 최적화 작업 진행.
DB 최적화 관련 세부 작업 내역은 2-1, 2-2, 2-3 문서 참조.

4. DB 최적화 이후 성능 테스트

4.1 부하 테스트 클라이언트 환경

- MS-Azure D14 서버 8대 (16Core, 112GB)
- 일본 1대, 동남아 7대

mobieeteam2-1 →	실행 중	Microsoft Azure...	동남아시아
mobileteam	실행 중	Microsoft Azure...	일본 서부
mobileteam2	실행 중	Microsoft Azure...	동남아시아
mobileteam2-2	실행 중	Microsoft Azure...	동남아시아
mobileteam2-3	실행 중	Microsoft Azure...	동남아시아
mobileteam2-4	실행 중	Microsoft Azure...	동남아시아
mobileteam2-5	실행 중	Microsoft Azure...	동남아시아
mobileteam2-6	실행 중	Microsoft Azure...	동남아시아



4.2 부하 테스트 방식

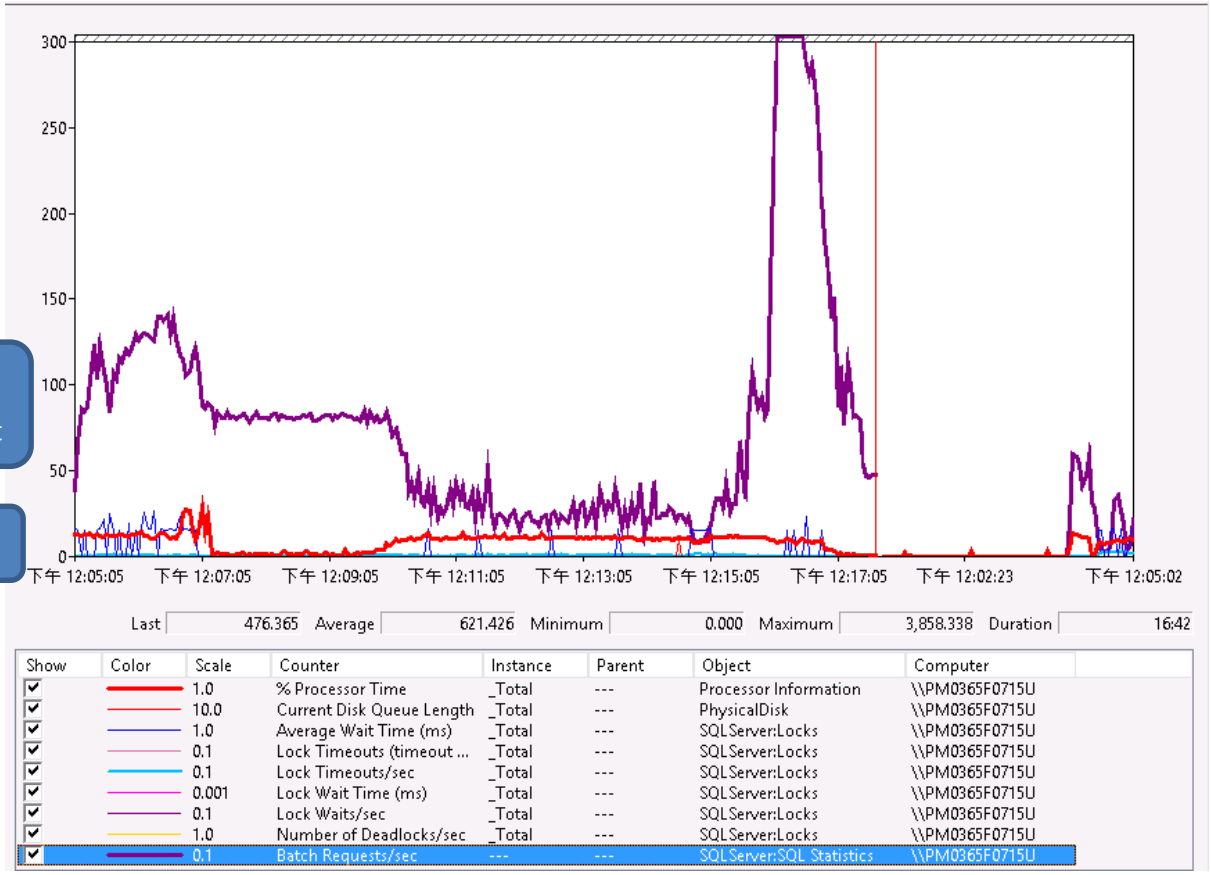
- 각 서버에 DummyClient 셋팅 (NumberOfClient=1000, NumberOfThread=500)
- 각 서버에서 순차적으로 부하테스트 클라이언트 가동

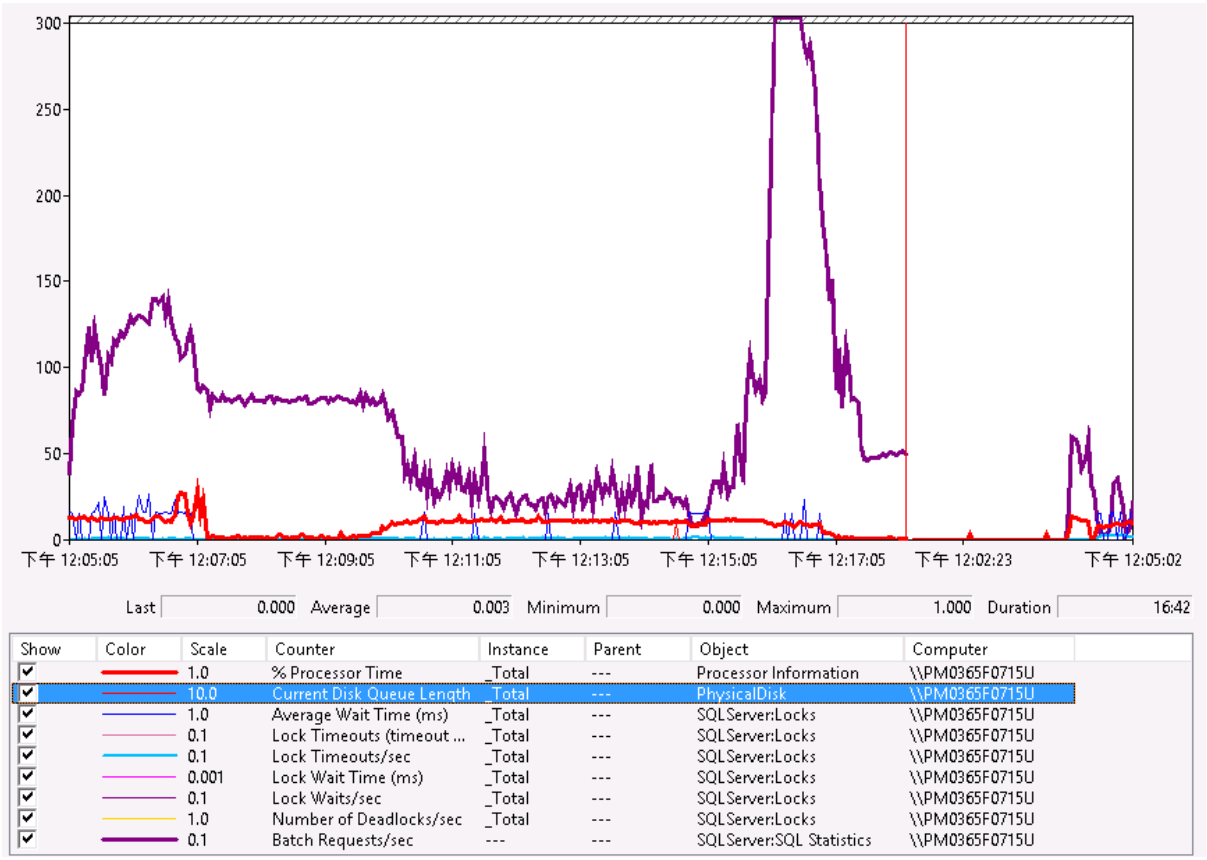
4.3 DB 성능

- 최대 3858 Batch Request 에서 CPU 20% 이내, Disk 는 Queue Length 1 (5 이하로 유지 되어야 함) 이내 에서 모두 처리됨.
- 그래프의 모습이 불규칙적인 것은 봇 클라이언트의를 통한 부하 트래픽 유입이 불규칙한 것이 원인인 것으로 보임

Batch Request

CPU





5. 결과

5.1 최종 성능 테스트 결과

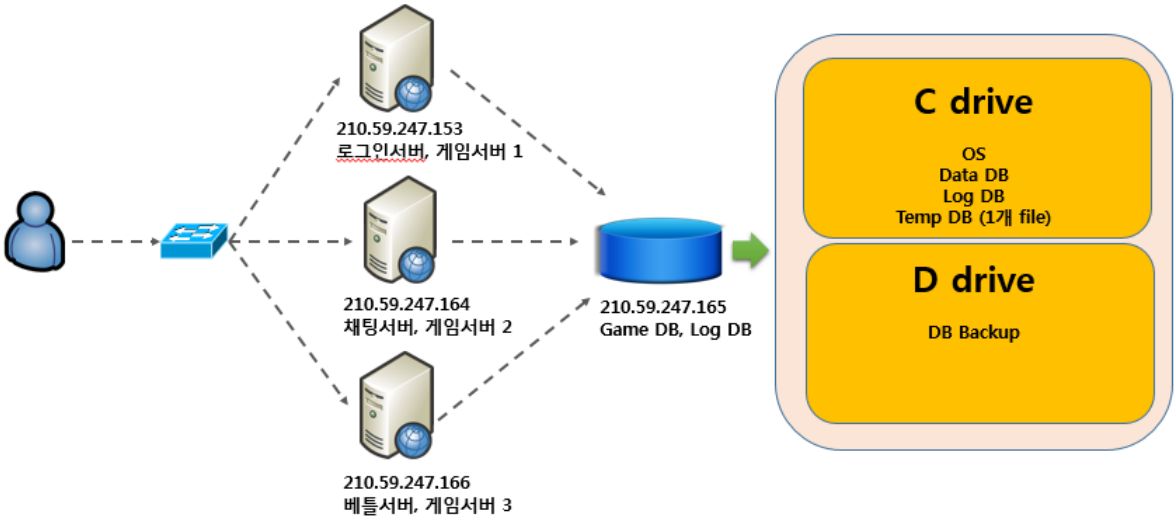
- 최초 목표했던 1935 Batch Request 달성 (DAU 10만 기준)
- 3800 Batch Request 에서도 안정적으로 처리 가능
- 불규칙한 그래프는 클라이언트의 영향으로 보임
  추가로 1회 정도 개발사와 직접 모니터링 시도 논의.

5.2 추가 개선 필요 사항

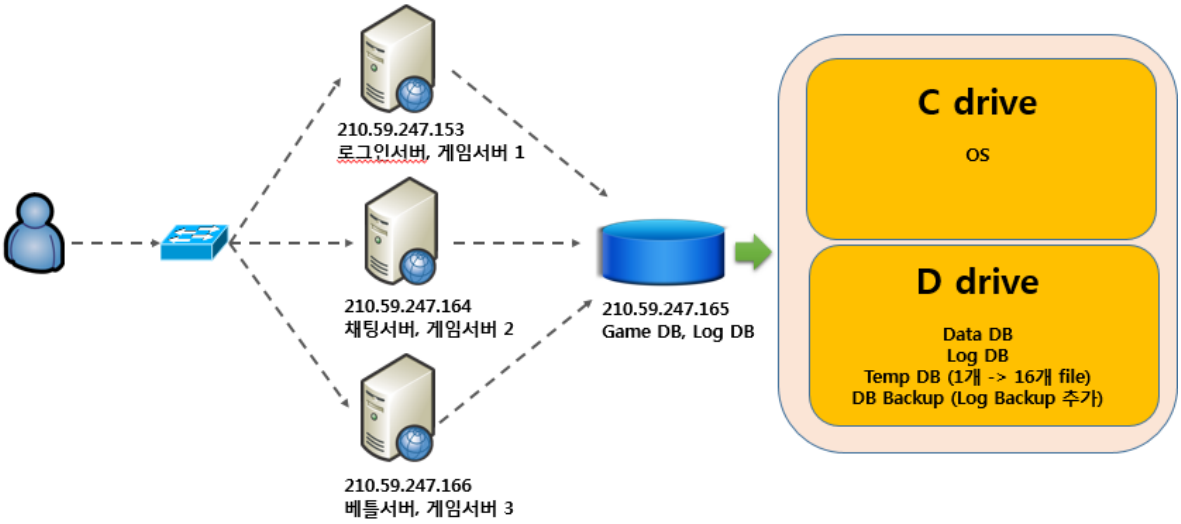
- 게임서버 에서 직접 만들어 사용하는 2개 쿼리에 대해서는 개발사에서 튜닝이 필요함.
  ("2-2. MOAZIO DB Query 튜닝 세부 내역서" 의 "1. 개발사 직접 적용 필요 query" 참조)
- SQL Injection 처리 확인
  아래 내용을 참고 하여 적용
  A. SQL Injection 을 대비하기 위하여, 사용자가 입력한 내용을 입력하는 쿼리는 반드시 프로시져로 작성
  B. 해당 프로시저에서 사용자가 입력한 값을 받는 변수는 바인딩 하여 사용할 것. (동적 쿼리 방식으로 쿼리를 생성하면 안됨)
  EX> 프로시저로 받은 @Poo 파라미터를 Where = @Poo 처럼 사용할 것. (O)
  동적 쿼리로 사용 금지. " ~~~ Where = '"+@Poo + "' " (X)

- TEST DB 는 DB 관련 파일들의 위치를 아래와 같이 변경함.
  Live DB 에도 점검 등의 시간에 동일하게 변경하는 것을 권장함.

<변경전>



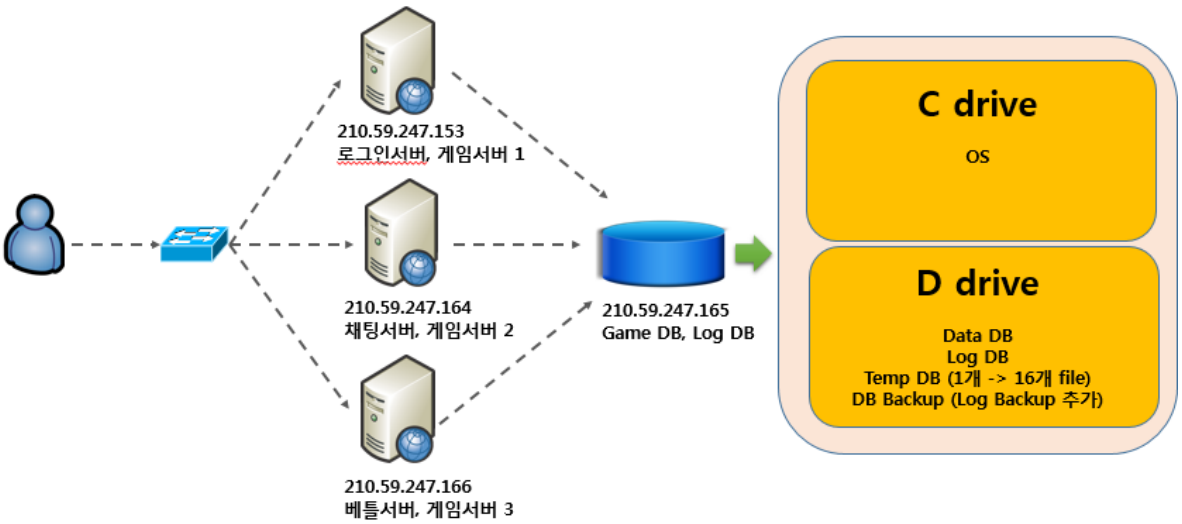
<변경후>



5.3 시스템 구성 개선안

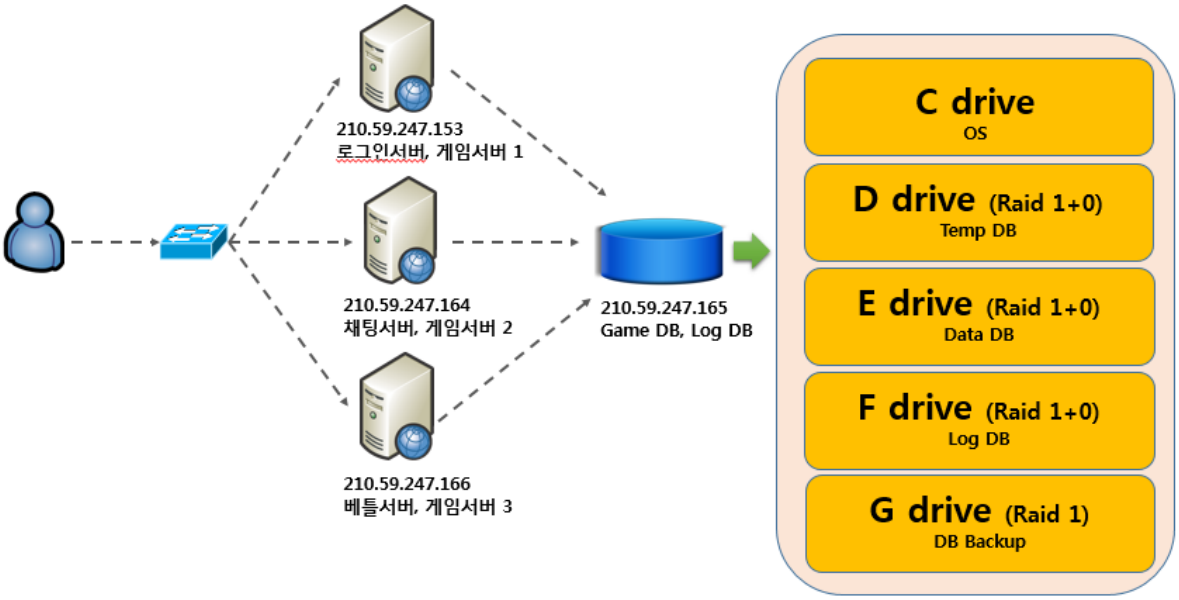
- 현재 게임서버는 더미클라이언트를 통한 부하 테스트에서는 트래픽에 비해 부하가 거의 없는 상태로 관련한 서버 증설은 필요해 보이지 않음
- 주 부하는 DB 에서 발생하고 있으며, 사용자 증가에 따른 성능의 1차 병목은 DB 에서 발생될 수 있음.
  사용자 트래픽에 따라 아래와 같은 구조로 변경하는 것을 권장함.

1) 현재



2) 1단계

물리적으로 분리된 디스크를 추가하고, 각각의 독립된 디스크로 분산 배치



3) 2단계

Data DB 와 Log DB 는 물리적으로 독립된 서버로 가동을 권장