



국민대학교
전자정보통신대학
컴퓨터공학부


캡스톤 디자인 I

종합설계 프로젝트

프로젝트 명	HTTP 프로토콜 기반 ARCUS Proxy 서버 개발
팀 명	이다숨(Jam2in)
문서 제목	계획서

Version	1.2
Date	2019-MAR-29

이름	이다숨
----	-----


 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	Http 프로토콜 기반 ARCUS Proxy 서버 개발	
	팀 명	이다숨(젼투인)	
	Confidential Restricted	Version 1.2	20xx-MAR-25

CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인I 수강 학생 중 프로젝트 "**HTTP** 프로토콜 기반 **ARCUS proxy server** 개발"을 수행하는 팀 "**이다숨(Jam2in)**"의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 "**이다숨(Jam2in)**"의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.


문서 정보 / 수정 내역


수정날짜	대표수정 자	Revision	추가/수정 항목	내 용
2019-03-10	이다숨	1.0	최초 작성	

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	Http 프로토콜 기반 ARCUS Proxy 서버 개발	
	팀 명	이다숨(잼투인)	
	Confidential Restricted	Version 1.2	20xx-MAR-25

목 차

1	개요	5
1.1	프로젝트 개요	5
1.2	추진 배경 및 필요성	5
1.2.1	ARCUS 소개	5
1.2.2	ARCUS Proxy 서버의 필요성	6
2	개발 목표 및 내용	7
2.1	목표	7
2.2	연구/개발 내용	7
2.2.1	ARCUS Cluster 연결을 위한 모듈 개발	7
2.2.2	ARCUS Proxy 서버 목록 및 기능 개발	7
2.2.3	요청 및 응답 처리 모듈 개발	7
2.2.4	ARCUS proxy server 기능 및 성능 테스트	8
2.3	개발 결과	9
2.3.1	시스템 기능 요구사항	9
2.3.2	시스템 비기능(품질) 요구사항	9
2.3.3	시스템 구조	10
2.4	기대효과 및 활용방안	11
2.4.1	기대효과	11
2.4.2	활용방안	11
3	배경 기술	12
3.1	기술적 요구사항	12
3.1.1	개발 환경	12
3.1.2	결과물 확인 환경	12
3.2	현실적 제한 요소 및 그 해결 방안	12
3.2.1	하드웨어	12
3.2.2	소프트웨어	12
3.2.3	기타	13
4	개발 일정 및 자원 관리	14
4.1	개발 일정	14
4.2	일정별 주요 산출물	14

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	Http 프로토콜 기반 ARCUS Proxy 서버 개발	
	팀 명	이다숨(잼투인)	
	Confidential Restricted	Version 1.2	20xx-MAR-25

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	Http 프로토콜 기반 ARCUS Proxy 서버 개발	
	팀 명	이다숨(잼투인)	
	Confidential Restricted	Version 1.2	20xx-MAR-25

1 개요

1.1 프로젝트 개요

ARCUS는 네이버에서 개발하고 오픈한 공개 SW로, Memcached와 Zookeeper 기반의 메모리 캐시 클러스터 솔루션이다. ARCUS를 사용하기 위해 응용 서비스에서 ARCUS 클라이언트 라이브러리 메소드를 호출하는 형태로, 현재 ARCUS 클라이언트 라이브러리의 범용성 제약으로 인해 사용성이 떨어지는 상황이다. 응용 서비스의 ARCUS 사용성을 높이기 위하여 응용 서비스와 ARCUS 서버 사이에 클라이언트 라이브러리를 포함한 형태의 **HTTP protocol 기반 ARCUS Proxy 서버**를 개발한다. 클라이언트 라이브러리는 arcus-java-client를 기반으로 하여 범용적 사용을 위해 Spring 환경에서 개발하고 합리적인 성능을 목적으로 한다.

1.2 추진 배경 및 필요성

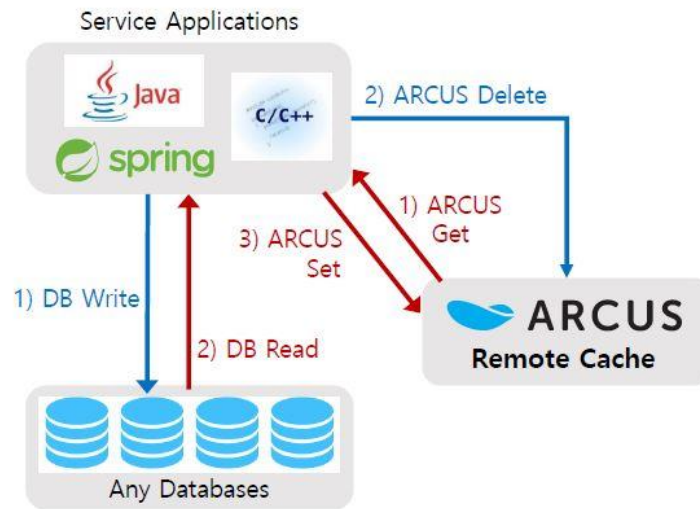
1.2.1 ARCUS 소개

ARCUS란, Memcached와 Zookeeper를 기반으로 확장한 메모리 캐시 클라우드이며 key-value 쌍의 기본적인 Memcached 데이터 모델을 지원하는 것 외에도 B+Tree, List, Set, Map과 같은 Collection 데이터 모델을 지원한다.

대규모 사용자를 대상으로 하는 최근의 다양한 응용 서비스들은 수많은 사용자와 환경에 의해 수없이 생산되는 콘텐츠들을 저장 관리하면서 사용자가 요구하는 시점에 그 사용자와 관계된 모든 정보들을 신속히 간추려 제공해야 한다. 이런 응용 서비스들의 데이터 저장 및 조회 처리를 위하여 기존 RDBMS(Relational Database Management System)만을 사용하는 경우, 해당 응용 서비스는 반드시 성능적 한계를 겪게 된다.


이러한 성능적 한계를 극복하고자 다수의 고비용 RDBMS 장비를 사용하여 데이터를 나누어 저장 및 관리하는 샤딩(Sharding) 기술을 이용하지만, 고비용을 지불해야 하는 문제 외에도 정적 구조의 RDBMS 샤딩 기술만으로는 여러 이벤트나 사회적인 이슈 등에 의해 실시간으로 급변하는 사용자 요청량에 유연하게 대응하기 힘든 문제가 있다. ARCUS는 RDBMS에 존재하는 데이터 중 자주 접근되는 데이터 일부(Hot spot data)를 메모리에 적재하여 사용자의 요청에 대한 빠른 응답과 높은 처리량을 제공할 수 있으면서 백엔드 저장소인 RDBMS에 들어가는 질의 부하를 감소시킬 수 있는 시스템이다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	Http 프로토콜 기반 ARCUS Proxy 서버 개발	
	팀 명	이다숨(젼투인)	
	Confidential Restricted	Version 1.2	20xx-MAR-25



1.2.2 ARCUS Proxy 서버의 필요성

ARCUS는 공식적으로 Java 클라이언트와 C 클라이언트만을 지원하고 있는 상태로, 응용 서비스에서 ARCUS 사용이 제한적이다. 가장 잘 알려진 캐시 솔루션인 Redis는 거의 대부분의 언어에 대해 라이브러리를 제공하고 있지만 ARCUS를 개발한 젼투인의 인력 상황에서 이러한 개발 작업은 매우 큰 비용이 발생한다. 따라서 다양한 클라이언트들을 수용하고 정보를 더욱 자유롭게 송수신 하기 위해 HTTP 프로토콜 기반의 Proxy 서버를 필요로 하게 되었다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	Http 프로토콜 기반 ARCUS Proxy 서버 개발	
	팀 명	이다숨(잼투인)	
	Confidential Restricted	Version 1.2	20xx-MAR-25

2 개발 목표 및 내용

2.1 목표

ARCUS의 언어와 환경에 대한 제약을 줄여 사용성을 증대시키는 ARCUS Proxy 서버를 개발한다.

2.2 연구/개발 내용

2.2.1 ARCUS Cluster 연결을 위한 모듈 개발

Java Web Framework인 Spring 애플리케이션 서버를 이용하여 Application으로부터 HTTP 요청이 발생하면, ARCUS java client API method로 변환하여 ARCUS 서버에 요청하게 한다.

ARCUS Cluster 연결을 위해 properties 생성 방식의 connection 모듈을 개발한다. ARCUS 서버로 요청하기 위해 arcus-java-client library에서 제공하는 ArcusClient 객체를 생성하는 과정이 필요하다. ARCUS Cluster를 식별하는 Service code와 Zookeeper 주소 등의 정보를 담은 arcus.properties를 생성하고 그 정보를 ArcusConfig.java에서 읽어 ArcusClient 객체를 생성하는 과정을 통해 ARCUS Cluster로 요청을 보내기 위한 준비 작업을 수행한다.


2.2.2 ARCUS Proxy 서버 목록 및 기능 개발

구현하는 API의 Item data type은 key-value와 B+Tree Item 연산으로 한정한다.

- Key-value Item 연산 : 저장, 조회, 값의 증감, 삭제
- B+Tree Item 연산 : 생성, 삽입, Upsert, 변경, 삭제, 값의 증감, 개수 계산, 조회, 일괄 삽입, 일괄 조회, Sort-Merge 조회, position 조회, position 기반의 Element 조회, Position과 Element 동시 조회


2.2.3 요청 및 응답 처리 모듈 개발

HTTP protocol proxy server와 ARCUS Cluster 간 연산을 요청하거나 연산 결과에 대한 응답을 처리하기 위해 요청 및 응답처리 모듈을 개발한다. 다양한 형식의 메시지 포맷(xml, json, ..)으로 요청받거나 응답할 수 있도록 하기 위해 POJO 형태의 클래스를 인자로 받고 리턴시킨다. 또한 ARCUS Operation 중 발생한 모든 실패 상황을 Exception으로 Spring의 ExceptionHandler에서 처리하도록 하여 클라이언트에게 에러 원인을 정확히 알려줄 수 있도록 한다. (예 : 같은 HTTP 500 에러 코드이지만 발생한 원인을 OperationTimeoutException, MessageExecutionException 등으로 나눌 수 있다.)

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	Http 프로토콜 기반 ARCUS Proxy 서버 개발	
	팀 명	이다솜(잼투인)	
	Confidential Restricted	Version 1.2	20xx-MAR-25

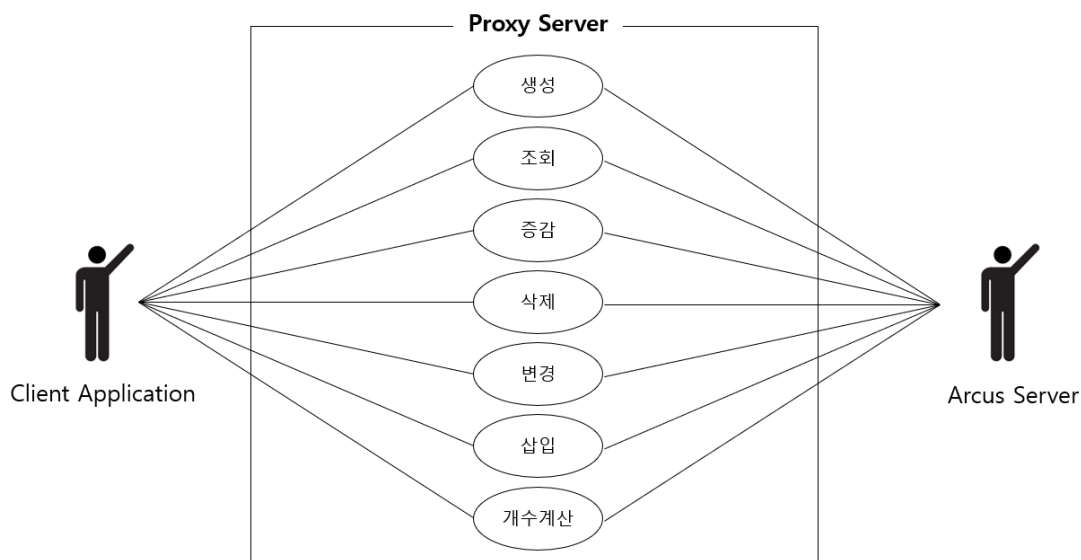
2.2.4 ARCUS proxy server 기능 및 성능 테스트

개발한 ARCUS Proxy 서버는 Postman을 이용하여 요청을 보내고, 결과를 확인하는 방법으로 기능 테스트한다. 성능 테스트는 nGrinder를 기반으로 구성하여 수행한다. nGrinder는 네이버에서 성능 측정 목적으로 jython으로 개발된 오픈소스 프로젝트로 ARCUS proxy server의 성능을 테스트한다. Jam2in VM 장비에 ARCUS proxy server를 배포하고, nGrinder Controller, nGrinder Agent 2개를 이용해 proxy server에 스트레스 부하를 주어 ARCUS cluster에 대해 요청 처리 성능 테스트를 진행한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	Http 프로토콜 기반 ARCUS Proxy 서버 개발	
	팀 명	이다숨(젼투인	
	Confidential Restricted	Version 1.2	20xx-MAR-25

2.3 개발 결과

2.3.1 시스템 기능 요구사항




클라이언트는 Proxy Server에 ARCUS에서 지원하는 데이터 모델(Key-value, B+Tree) 중 원하는 데이터 모델에 대해 동작을 요청하고, Proxy Server는 요청을 ARCUS Server에 전달한다.

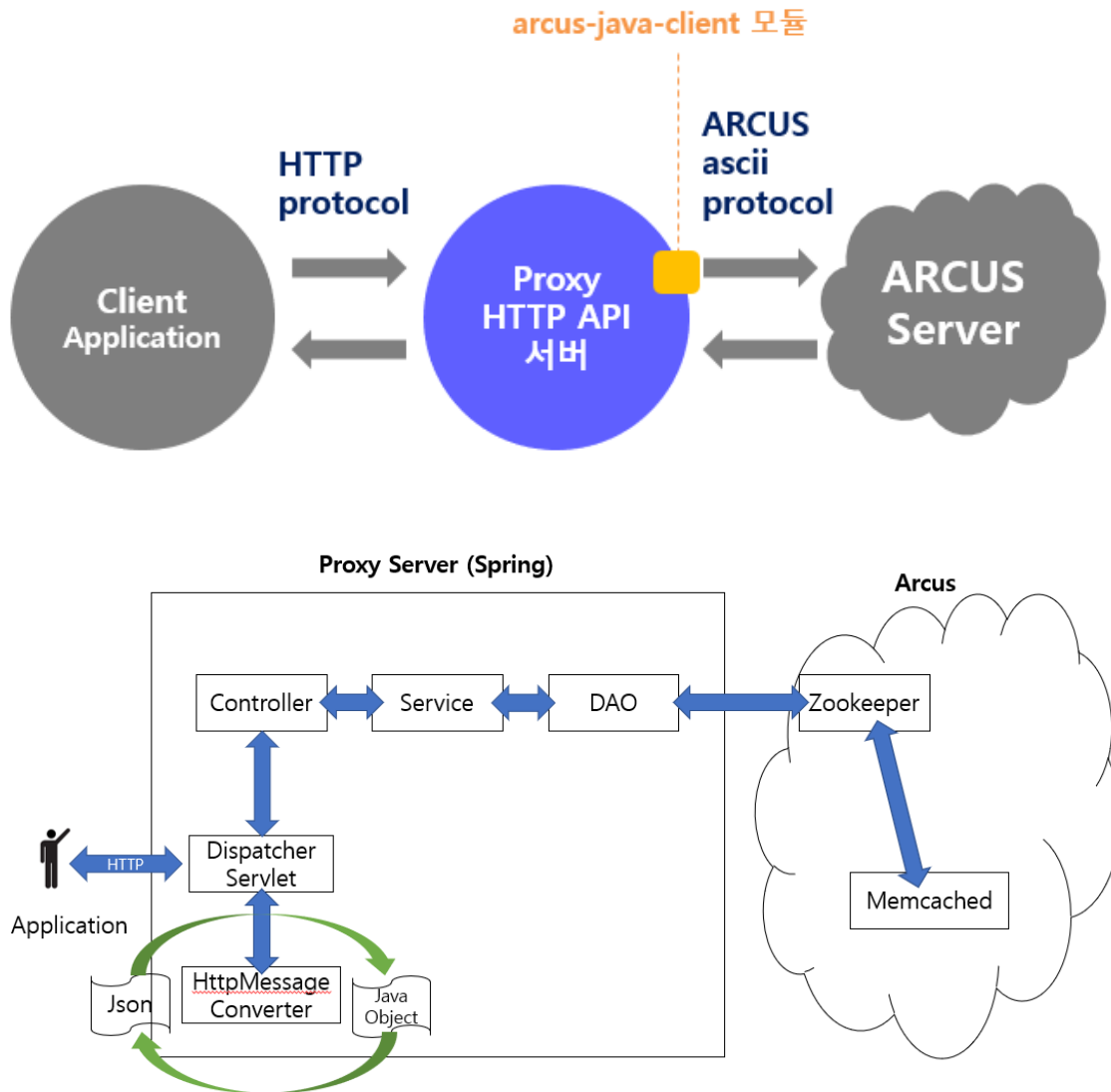
2.3.2 시스템 비기능(품질) 요구사항

[Performance]

ARCUS HTTP API는 nGrinder로 성능측정을 진행한다. Agent는 두 개를 서로 다른 장비에 설치하여 코어 수를 늘린다. 동시접속자 6000명일 경우 적어도 초당 8000번의 트랜잭션을 처리할 수 있어야 한다. (HTTP API의 특성상 성능이 높지 않을 것으로 예상)

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	Http 프로토콜 기반 ARCUS Proxy 서버 개발	
	팀 명	이다숨(젼투인)	
	Confidential Restricted	Version 1.2	20xx-MAR-25

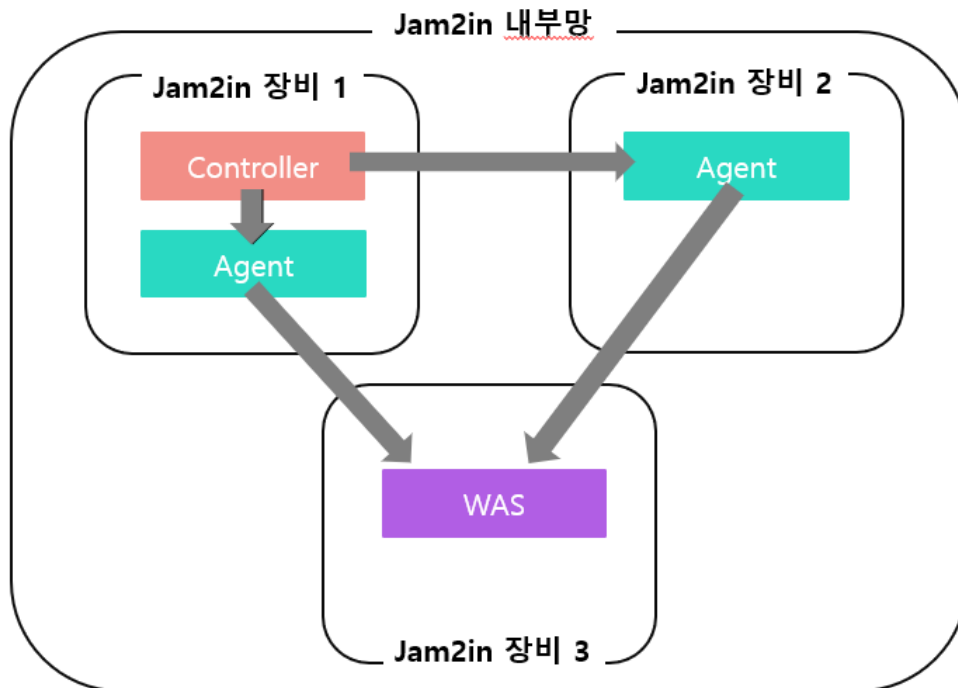
2.3.3 시스템 구조



- Client Application에서 HTTP 요청을 하면, ARCUS용 HTTP 기반 Proxy Server(이하 프록시 서버)는 import한 arcus-java-client 모듈을 이용하여 ARCUS 커맨드로 변환한 뒤 ARCUS Server에 보낸다.
- ARCUS 서버는 요청 결과를 프록시 서버에 응답하고, 프록시 서버는 Client Application에 결과 데이터를 가공하여 반환한다.
- 요청 중 JSON 포맷은 HttpMessageConverter를 통해 Json을 Java Object로 변환하고, 응답 시에도 마찬가지로 Java Object를 Json으로 변환한다.

[성능 테스트]

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	Http 프로토콜 기반 ARCUS Proxy 서버 개발	
	팀 명	이다숨(젼투인)	
	Confidential Restricted	Version 1.2	20xx-MAR-25



Jam2in VM 장비에 ARCUS proxy server 를 배포하고, nGrinder Controller, nGrinder Agent 2 개를 이용해 proxy server 에 스트레스 부하를 주어 ARCUS cluster 에 대해 요청 처리 성능 테스트를 진행한다.

2.4 기대효과 및 활용방안

2.4.1 기대효과

Java 와 C 클라이언트만 지원되었던 ARCUS 를 HTTP API 를 이용해 더욱 유연하고 쉽게 활용할 수 있다.

2.4.2 활용방안

Spring Framework 를 이용하는 기업의 ARCUS HTTP API 로 배포되어 활용된다. 다른 Data Collection Type 까지 완성되고 성능 테스트를 마치면 롯데시네마에 우선적으로 적용될 예정이다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	Http 프로토콜 기반 ARCUS Proxy 서버 개발	
	팀 명	이다숨(젼투인)	
	Confidential Restricted	Version 1.2	20xx-MAR-25

3 배경 기술

3.1 기술적 요구사항

3.1.1 개발 환경

- 운영체제 : Windows 64-bit
- 개발언어 : Java (jdk1.8.0_191)
- 컴파일 환경 : Tomcat v9.0
- IDE : Eclipse 2018-12 (Java EE)
- 애플리케이션 프레임워크 : Spring 3.9.7
- 필요 모듈 : arcus-java-client (<https://github.com/jam2in/arcus-java-client>)
- 언어의 문법적 요구사항 : arcus-java-client 기본 Command에 대한 이해

3.1.2 결과물 확인 환경

- 운영체제 : Windows, Linux, Mac OS X
- 서버 환경 : Tomcat
- API 배포 장비 : jam2in-s001

3.2 현실적 제한 요소 및 그 해결 방안

3.2.1 하드웨어

서버와 클라이언트를 연결하는 proxy 역할을 하는 HTTP API이므로 하드웨어 제한 요소는 없다.

3.2.2 소프트웨어


성능이 낮거나 아예 나오지 않는 원인은 아래와 같다.

- 배포한 WAS(ARCUS용 HTTP API)가 느리다.
Refactoring 과정을 통해 코드의 가독성을 높이고 Log 레벨을 초기 설정 값인 DEBUG에서 INFO로 조절하는 방법으로 해결할 수 있지만, 가능성은 낮다.
- HTTP 특성(Stateless) 상 TPC를 이용해 연결하고 통신 후 끊는다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	Http 프로토콜 기반 ARCUS Proxy 서버 개발	
	팀 명	이다숨(잼투인)	
	Confidential Restricted	Version 1.2	20xx-MAR-25

- c. Agent Socket가 Server Socket에 접속할 때 포트는 OS에서 자동으로 바인딩해준다. nGrinder는 서버에 대해 부하를 빠르고 많이 주게 되므로 port의 create와 close 또한 빠르게 일어난다. Port가 close가 완전히 되기 전에 port를 재사용하려 하므로 오류가 발생할 수 있다. 커널 파라미터를 net.ipv4.tcp_tw_reuse = 1, net.ipv4.tcp_tw_recycle = 1로 수정해준다.

3.2.3 기타

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	Http 프로토콜 기반 ARCUS Proxy 서버 개발	
	팀 명	이다숨(젼투인)	
	Confidential Restricted	Version 1.2	20xx-MAR-25


4 개발 일정 및 자원 관리

4.1 개발 일정

항목	세부내용	1 월	2 월	3 월	4 월	5 월	비고
요구사항분석 및 설계	REST API 조사						
	HTTP API(KV) SRS 작성						
	HTTP API(B+Tree) SRS 작성						
구현 (Key-Value)	스프링 프레임워크 설치						
	Arcus-java-client 환경 적응						
	Config, .properties 파일 구현						
	Controller/Service/DA O 구현						
	API 테스트 (Postman)						
	refactoring						
성능 측정 (Key-Value)	nGrinder 시험환경 구축						
	nGrinder 성능 테스트						
구현 (B+Tree)	.properties 파일 구현						
	Controller/Service/DA O 구현						
	API 테스트 (Postman)						
	refactoring						
성능 측정 (B+Tree)	nGrinder 시험환경 구축						
	nGrinder 성능 테스트						

4.2 일정별 주요 산출물

마일스톤	개요	시작일	종료일
계획서 발표	Key-value 데이터 타입 API 산출물 : 1. 프로젝트 수행 계획서(PDF) 2. 프로젝트 수행 계획 발표 슬라이드 쇼(PPT) 3. 프로젝트 수행 계획 발 표 슬라이드 파일(PDF)		2019-03-15

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	Http 프로토콜 기반 ARCUS Proxy 서버 개발	
	팀 명	이다숨(젼투인)	
	Confidential Restricted	Version 1.2	20xx-MAR-25

성능 측정 완료	Key-value 데이터 모델 성능 측정 완료 [Refactoring] Request Class 이름 수정 [Refactoring] Response Class 수정 (성공/실패) 산출물 : 1. 성능 측정 결과 문서		2019-03-29
설계 완료	시스템 설계 완료 산출물 : 1. ARCUS HTTP API (B+Tree) 명세서	~	2019-04-14
1 차 중간 보고	설정 arcus.properties 추가 산출물 : 1. 프로젝트 1 차 중간 보고서 2. 프로젝트 진도 점검표 3. 1 차분 구현 소스 코드	~	2019-04-19
2 차 중간 보고	Controller/Service/DAO 구현 완료 Request, Response 를 위한 클래스 생성 (Postman) API 테스트 완료 산출물 : 1. 프로젝트 2 차 중간 보고서 2. 프로젝트 진도 점검표 3. 2 차분 구현 소스 코드	~	2019-05-17
구현 완료	[Refactoring] 산출물: 1. Arcus 용 HTTP API(Arcus-http-api.war)	~	2019-05-24
성능 측정 완료	시스템 통합 테스트 산출물: 1. 성능 측정 결과 문서	~	2019-05-30
전시 및 최종 보고	최종 보고 산출물: 1. 전시용 자료(포스터 및 소개 책자) 2. 온라인 평가용 자료 3. 최종 결과 보고서		2019-05-31