



국민대학교  
전자정보통신대학  
컴퓨터공학부

# 캡스톤 디자인 I

## 종합설계 프로젝트

프로젝트 명	<i>HTTP 프로토콜 기반 ARCUS Proxy 서버 개발</i>
팀 명	<i>이다숨(JaM2in)</i>
문서 제목	중간보고서

Version	1.0
팀원	이다숨
지도교수	이시윤 교수
Date	2019-04-19

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	HTTP 프로토콜 기반 ARCUS Proxy 서버 개발	
	<b>팀 명</b>	이다숨(JaM2in)	
	Confidential Restricted	Version 1.0	2019-APR-19

## 문서 정보 / 수정 내역


<b>Filename</b>	중간보고서_20153207이다숨
<b>원안작성자</b>	이다숨
<b>수정작업자</b>	

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2019-04-17	이다숨	1.0	최초 작성	

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	HTTP 프로토콜 기반 ARCUS Proxy 서버 개발	
	<b>팀 명</b>	이다숨(JaM2in)	
	Confidential Restricted	Version 1.0	2019-APR-19

## 목 차

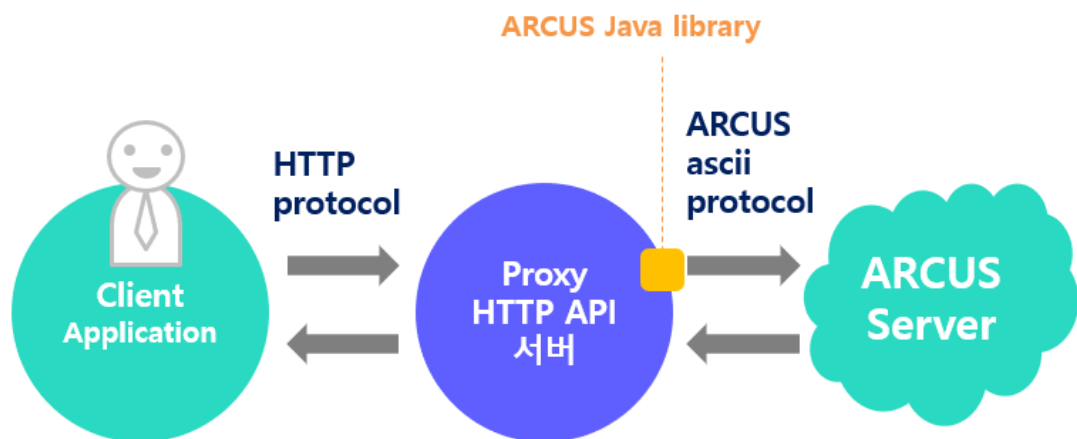
1	프로젝트 목표 .....	4
2	수행 내용 및 중간결과 .....	5
2.1	계획서 상의 연구내용 .....	5
2.2	수행내용 .....	6
3	수정된 연구내용 및 추진 방향 .....	8
3.1	수정사항 .....	8
4	향후 추진계획 .....	9
4.1	향후 계획의 세부 내용 .....	9
5	고충 및 건의사항 .....	10


 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	HTTP 프로토콜 기반 ARCUS Proxy 서버 개발	
	팀 명	이다숨(JaM2in)	
	Confidential Restricted	Version 1.0	2019-APR-19

## 1 프로젝트 목표

ARCUS는 응용 서비스의 급변하는 사용자 요청에 유연히 대응하기 위해 NAVER에서 개발된 메모리 캐시 클러스터이다. 자주 접근되는 데이터의 일부를 Remote Cache 인 ARCUS에 적재 함으로써 백엔드 저장소인 DBMS의 부하를 경감, 대용량 데이터 캐싱, 데이터간 consistency 유지 등의 장점을 가지게 된다.

하지만 ARCUS는 Java와 C 클라이언트만을 지원하고 있는 상태이다. ARCUS의 사용 환경에 대한 제약을 줄여 사용성을 증대시킬 수 있도록 <HTTP Protocol 기반의 ARCUS Proxy 서버>를 개발한다.



 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	HTTP 프로토콜 기반 ARCUS Proxy 서버 개발	
	팀 명	이다숨(JaM2in)	
	Confidential Restricted	Version 1.0	2019-APR-19

## 2 수행 내용 및 중간결과

### 2.1 계획서 상의 연구내용

#### 2.1.1 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.


#### 2.1.2 ARCUS Proxy 서버 목록 및 기능 개발

구현하는 API의 Item data type은 Key-value와 B+Tree Type으로 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, Postion과 Element 동시 조회

#### 2.1.3 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 Opertion 수행 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.)

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	HTTP 프로토콜 기반 ARCUS Proxy 서버 개발	
	<b>팀 명</b>	이다숨(JaM2in)	
	Confidential Restricted	Version 1.0	2019-APR-19

## 2.1.4 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.

## 2.2 수행내용

### 2.2.1 Key-value item 연산의 구현

- 저장 연산
  - ◆ <key, obj>의 key-value item을 저장하는 set, add 연산
  - ◆ 주어진 key의 value 부분을 교체하는 replace 연산
  - ◆ 주어진 key의 value 부분에서 가장 앞쪽 또는 뒤쪽에 새로운 value를 추가하는 prepend, append 연산
  - ◆ 한번의 API호출로 다수의 key-value items를 저장하는 asyncSetBulk 연산
- 조회 연산
  - ◆ 하나의 key에 대한 value를 조회하는 asyncGet 연산
  - ◆ 여러 key들의 value들을 한번에 조회하는 asyncGetBulk 연산
  - ◆ 주어진 key의 속성(데이터 타입, expire Time 등)을 조회하는 asyncGetAttr 연산



## ■ 증감 연산

- ◆ 주어진 key의 value 값을 증가시키거나 감소시키는 `asyncIncr`, `asyncDecr` 연산

## ■ 삭제 연산

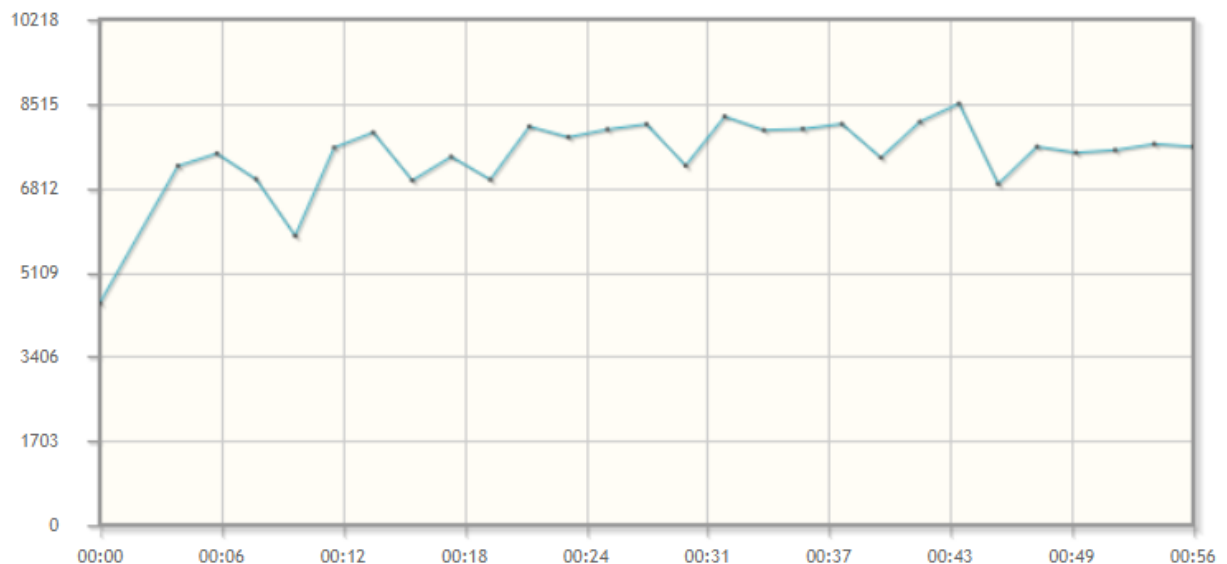
- ◆ 주어진 key를 가진 item을 cache에서 삭제하는 `delete` 연산

## ■ 개인 Github에 전체 코드에 대한 리뷰를 요청한 상태

## 2.2.2 Key-value item 연산까지 구현한 서버에 대해 성능테스트 완료

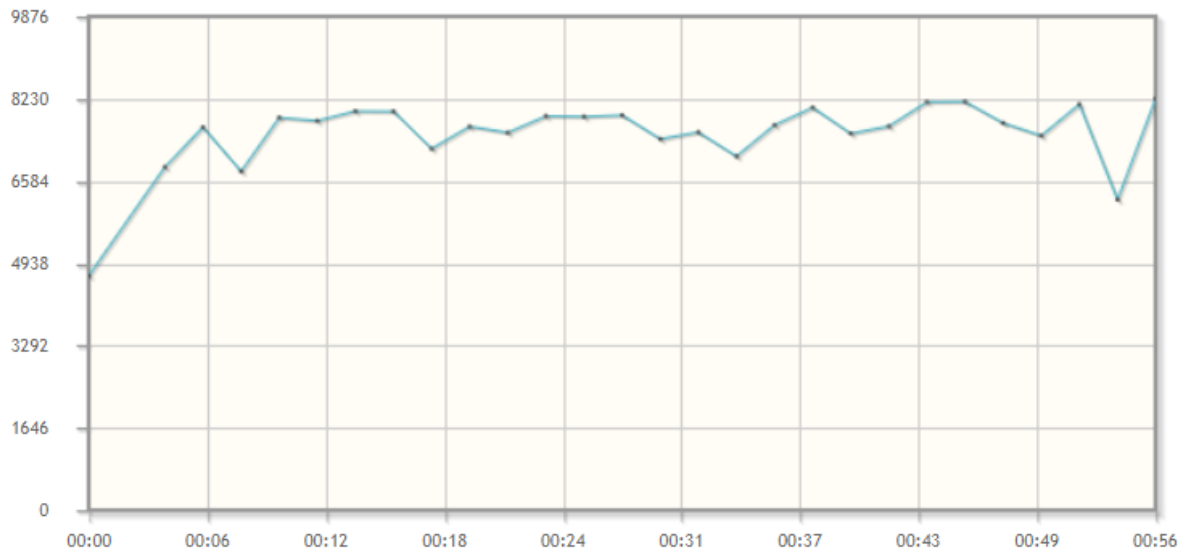
### ■ 성능 테스트 : 현재 잼투인 내부 장비에 배포 후 성능 측정 진행

- ◆ 저장 연산 – 평균 7,483 TPS(Transaction Per Sec), 최고 8,514 TPS





◆ 조회 연산 – 평균 7,527.1 TPS, 최고 8,169.5 TPS



### 2.2.3 B+Tree item 서버 명세 작성

- 현재 <Arcus Server Ascii Protocol 문서의 내용>, <arcus-java-client/b+tree Item> 문서 공부 중
- 해당 문서들이 길어 명세 작성까지 하려 하였으나 미뤄짐

## 3 수정된 연구내용 및 추진 방향

### 3.1 수정사항

없음.



## 4 향후 추진계획

### 4.1 향후 계획의 세부 내용

#### 4.1.1 개발 일정

항목	세부내용	4 월	5 월	비고
요구사항 분석 및 설계 (B+Tree)	HTTP API(B+Tree) SRS 작성			
구현 (B+Tree)	.properties 파일 구현			
	Controller/Service/DAO 구현			
	API 테스트 (Postman)			
	refactoring			
성능 측정 (B+Tree)	nGrinder 시험환경 구축			
	nGrinder 성능 테스트			

#### 4.1.2 요구사항 분석 및 설계 (B+Tree)

캡투인의 <\*\*\*Arcus Server Ascii Protocol 문서의 내용\*\*\*>, <arcus-java-client/b+tree Item> 문서를 참고하여 B+Tree에 대한 프록시 서버의 생성/삽입/변경/삭제/증감/개수계산/조회에 대한 연산을 명세한다.

#### 4.1.3 B+Tree에 대한 ARCUS Proxy Server 개발

Spring 어플리케이션 서버를 이용하여 B+Tree 아이템에 대한 HTTP 요청이 발생하면, ARCUS java client API method로 변환하여 ARCUS 서버에 요청한다. 기능 테스트는 전과 마찬가지로 Postman을 이용한다.

#### 4.1.4 B+Tree ARCUS Proxy Server 성능 테스트

Naver에서 성능 측정을 목적으로 개발된 nGrinder를 이용하여 성능 측정을 진행한다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	HTTP 프로토콜 기반 ARCUS Proxy 서버 개발	
	<b>팀 명</b>	이다숨(JaM2in)	
	Confidential Restricted	Version 1.0	2019-APR-19

## 5 고충 및 건의사항

- 교내 캡스톤 전용 프로젝트 공간 필요