

GraphQL을 이용한 가정용 에너지저장시스템 상태 API 서버 구현

이응기, 권기웅, 윤정미

전자부품연구원

{leg429, kiwoong.kwon, yunjm}@keti.re.kr

Implementation of Home Energy Storage System Status API using GraphQL

Lee Eung Gi, Kwon Ki Woong, Yun Jung Mee

Korea Electronics Technology Institute

요약

본 논문은 가정용 에너지저장시스템(ESS: Energy Storage System)의 상태 데이터를 사용자용 애플리케이션의 설정에 따라 필요한 데이터만 전달하기 위해 GraphQL을 이용하여 API 서버를 구현하였고 이에 관해 설명하고 있다. 그리고 기존에 사용 중이던 ESS RESTFul API 서버와 성능을 비교하기 위하여 다양한 상황에서의 응답속도를 측정하고 비교하였다.

I. 서론

본 논문은 GraphQL을 이용하여 가정용 ESS 게이트웨이에 저장된 ESS 상태 및 운행기록 데이터를 사용자용 애플리케이션으로 전달하기 위한 API 서버를 구현하고 성능을 분석했다.[1]

사용자용 애플리케이션은 사용자의 설정에 따라 대시보드나 ESS 상태 정보 조회 페이지에 표시해야 하는 정보가 달라진다. 따라서 사용자용 애플리케이션(클라이언트)은 API 서버로부터 필요한 데이터를 요청하는 메시지를 사용자의 설정에 따라 동적으로 변경할 수 있어야 한다. API 서버 또한 요청 메시지에서 요구하는 데이터에 맞게 응답 메시지에 필요한 데이터만 실어 보낼 수 있어야 한다. 이러한 요구사항에 따라 본 논문에서는 Web API로 흔하게 사용되는 RESTFul API 대신 GraphQL을 사용했다.

GraphQL은 Facebook에서 개발한 데이터 질의 언어이다. GraphQL Blog에서 확인할 수 있듯 GraphQL은 다양한 종류의 디바이스, 운영체제, 애플리케이션 각각의 요구사항에 따라 최적의 데이터를 제공하기 위하여 개발되었다.[2]

본 논문의 본문에서는 ESS API 서버를 GraphQL로 구현한 방식과 요청/응답 메시지의 GraphQL 질의문 구조에 관해 설명하고, 기존에 사용하던 RESTFul API 서버와 성능을 비교하였다. 결론에서는 향후 연구에 관해 설명하였다.

II. GraphQL 서버 구조

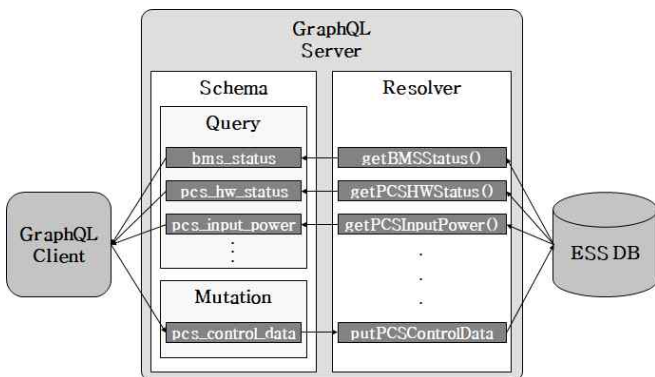


그림 1. ESS 데이터 GraphQL 서버 구조도

본 논문에서 구현한 GraphQL API 서버는 <그림 1>과 같이 Schema와 Resolver로 크게 두 부분으로 나뉜다. Schema는 서버로부터 데이터를 질의하는 Query 타입과 서버에 데이터를 입력하거나 수정할 수 있는 Mutation 타입이 있다. Query 타입은 bms_status, pcs_hw_status, pcs_input_power 등 ESS의 상태 데이터를 질의할 수 있는 다양한 종류의 객체 타입을 필드로 가진다. Mutation 타입은 pcs_control_data, sw_rest 과 같이 ESS DB에 ESS를 제어하기 위한 데이터를 입력할 수 있는 객체 타입을 필드로 가진다. 본 논문에서는 Query와 Mutation의 필드 타입을 ESS DB의 테이블 Schema와 동일하게 정의하여 모든 ESS 상태 데이터를 요청할 수 있도록 했다. GraphQL Client는 이렇게 정의된 Schema에 따라 필요한 객체 타입과 필드만 골라서 서버에게 요청할 수 있어, 원하는 데이터를 원하는 형태로 한 번에 받을 수 있다. 또한, 객체 타입마다 인자(Arguments)를 정의하여 데이터를 요청할 때 특정한 인자를 전달할 수 있다. 본 논문에서는 <표 1>과 같이 'tags' 인자를 이용하여 특정 ESS의 데이터만 요청하거나 'from', 'to' 인자를 이용하여 특정 기간의 데이터만 요청할 수 있도록 했다. 이러한 Schema는 GraphQL Client의 요청에 의해 실행되면서 각각의 필드에 맞는 Resolver를 호출한다. Resolver에는 Schema의 타입에 맞게 데이터를 가져와 가공할 수 있도록 getBMSStatus(), getPCSHWStatus(), getPCSInputPower() 등과 같이 다양한 Resolver이 있다. 각각의 Resolver은 ESS DB에 저장된 ESS 상태 데이터를 SQL 질의로 가져온다. 인자가 있을 때는 SQL을 'WHERE' 절을 이용하여 데이터를 특정한다. 해당 데이터는 바로 GraphQL로 넘겨지기도 하지만, 상황에 따라 Resolver에서 가공되기도 한다.[3]

표 1. GraphQL Schema 중 일부

```
type bms_status {  
  time: String  
  bms_mode: Int  
  normal: Int ...  
}  
type AllBMSStatus {  
  tags: ess_info  
  data: [bms_status] }  
type Query {  
  bms_status(tags: ess_input, from: String, to: String): [AllBMSStatus] }
```

III. GraphQL API 서버 성능 비교

본 논문에서는 GraphQL을 이용한 ESS 상태 데이터 API 서버와 기존의 ESS RESTFul API 서버의 응답속도를 비교하는 실험을 진행하였다. 실험은 다음과 같은 방식으로 진행되었다.

- 1) ESS DB의 여러 가지 자원 중 일정 개수의 자원을 임의로 선택한다.
- 2) 선택된 자원에 따라 GraphQL과 RESTFul API의 요청 메시지를 각각 작성한다.
- 3) 작성된 요청 메시지를 각각의 API 서버에게 요청하여 응답시간을 기록한다.
- 4) (3)번 과정을 30회 반복하여 평균값을 구한다.
- 5) (1)번 과정에서 선택 자원의 개수를 달리하여 위 과정을 반복한다.

실험을 진행한 서버의 사양과 API 패키지의 주요 모듈(버전)은 <표 2>와 같다.

표 2. GraphQL API 서버 사양 및 패키지 주요 모듈(버전)

- CPU : Intel(R) Core(TM) i5-7260U CPU @ 2.20GHz
- RAM : 16GB
- OS : Ubuntu 16.04.6 LTS
- DB : MySQL(v14.14)
- Dependencies : node.js(v12.14.0), express(v4.17.1), graphql(v14.6.0)

위와 같은 순서로 실험을 진행한 결과, <그림 2>와 같은 그래프를 얻을 수 있었다. 그래프의 파란색 막대는 GraphQL 서버의 30회 요청에 대한 응답시간(ms)의 평균을 나타낸 것이고, 주황색 막대는 RESTFul 서버에 30회 요청에 대한 평균을 나타낸 것이다. 각 막대그래프 위에 표시된 검은색 막대는 표준편차를 나타낸 것이다.

임의 자원의 개수가 작을 때는 GraphQL과 RESTFul API의 응답시간의 차이가 크지 않았지만, API 서버에게 요청하는 자원의 개수가 늘어날수록 응답시간의 차이가 점점 벌어지게 됐다. 이러한 실험결과는 GraphQL과 RESTFul API의 특징 차이에 의해 나타난 결과이다. GraphQL의 경우 여러 개의 자원이 하나의 ESS DB 테이블에 있으면 한 번의 질의로 데이터를 가져올 수 있었지만, RESTFul API의 경우 자원이 하나의 테이블에 있더라도 API를 수정하지 않는 이상 자원의 개수만큼 질의 횟수가 증가해야 한다. 그래서 RESTFul API의 요청 자원의 개수에 따른 응답속도 증가량이 더 높게 나타났다. 그러나 표준편차의 경우 RESTFul API는 거의 일정하게 증가하는 반면, GraphQL은 값의 변화가 일정하지 않은 것을 볼 수 있다. GraphQL의 경우 임의 자원이 ESS DB에서 저장된 위치에 따라 ESS DB 질의 횟수가 매번 달라지므로 자원의 종류에 따른 응답시간의 일관성을 확보하기가 어려웠다.

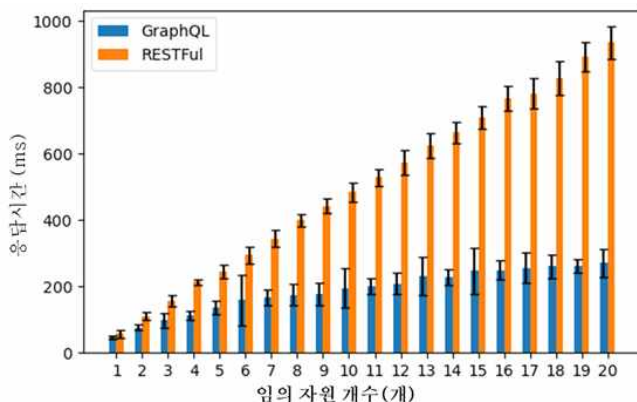


그림 2. GraphQL과 RESTFul API 응답속도 비교

IV. 결론

본 논문에서는 가정용 ESS의 상태 데이터를 요청하기 위한 API 서버를 GraphQL을 이용하여 구현하고, 기존의 RESTFul API 서버와 응답속도를 비교하였다.

추후 연구에서는 본 논문에서 사용한 MySQL; 관계형 데이터베이스 시스템 이외에 다른 데이터베이스 시스템을 사용하는 환경에서도 GraphQL과 RESTFul API의 성능을 비교하여 서비스 상황에 맞는 API를 제시할 수 있도록 하겠다.

ACKNOWLEDGMENT

본 연구는 2020년도 산업통상자원부의 재원으로 한국에너지기술평가원(KETEP)의 지원을 받아 수행한 연구과제입니다. (No. 201927101015E)

참 고 문 헌

- [1] 이응기, 하진희, 윤정미, 이상학, "가정용 ESS 자동제어를 위한 ESS 게이트웨이 개발," 2019년도 한국전파학회 하계종합학술대회 논문집, 7(1), pp. 527, Aug. 2019.
- [2] GraphQL: A query language for APIs. (n.d.). Retrieved July 30, 2020, from <https://graphql.org/>
- [3] D. A. Hartina, A. Lawi and B. L. E. Panggabean, "Performance Analysis of GraphQL and RESTful in SIM LP2M of the Hasanuddin University," 2018 2nd East Indonesia Conference on Computer and Information Technology (EIConCIT), Makassar, Indonesia, pp. 237-240, 2018.