

실시간 전기자동차 충전소 사용 로그 수집 시스템

윤주현*, 김연수*, 박선애*, 박소영*, 안진호**

Real-Time EV Charging Stations Usage Log Collection System

Yoon Ju Hyun*, Kim Yeon Su*, Park Seonae*, Park Soyoung*, and Jinho Ahn**

요 약

본 논문에서는 KECO API에서 제공하는 충전기 데이터를 이용해 전국의 충전기 사용 데이터를 실시간으로 수집하고, 충전소와 충전기 데이터를 정제 및 분석하여, 향후 사용을 위해 도출된 충전기의 사용 통계 정보를 로그화하여 저장하는 시스템을 개발하였다. 개발 시스템에서는 수십만 건 이상의 데이터를 이전 방식에 비해 벌크 쓰기와 비동기 처리기법을 적용하여 관리할 때 처리 속도가 크게 향상될 수 있음을 확인하였다. 운영 중인 개발 시스템에서 지속적으로 축적된 통계 정보를 잘 활용한다면, 현재 출시된 충전소 조회 및 추천 애플리케이션의 성능 및 사용자 편의성, 정보의 다양성을 개선할 수 있을 것으로 생각한다.

Abstract

This paper developed a system that collects national electronic vehicle charger usage data in real time using charger data provided by KECO API, cleanse and analyze them, and logs and stores deduced charger usage statistics information for further usage. In the system, it is verified that its processing speed can be significantly improved when managed with applying bulk writing and asynchronous data processing techniques to hundreds of thousands of data collected compared to the previous one. It is believed that currently released charging station inquiry and recommendation applications may enhance their performance, user friendliness, and information diversity if they well-utilize the statistical information accumulated continuously in our operating system.

Key words

EV Charging Stations, Asynchronous Data Processing, Bulk Write

1. 서 론

전기차 시대가 도래한 이후 전기자동차 충전소 정보를 제공하는 다양한 애플리케이션이 등장하였다. 그러나 대부분 충전소 위치, 충전기 단자 등과 같은 단순한 정보 조회만 가능한 수준이다. 그중 충전소별 혼잡 시간을 분석하는 기능이 있는 애플리

케이션이 존재하지만, 충전소마다 존재하는 서로 다른 종류인 충전기 단자 등을 고려하지 않고 단순히 충전소의 시간대별 혼잡도를 AI로 예상하여 제공하고 있다.

따라서 본 논문에서는 한국환경공단(이하 KECO)에서 제공하는 전기자동차 충전소 정보 API[1]를 통해 전국 충전소 및 충전기 정보 수집 및 저장하

* 소속: 경기대학교 컴퓨터공학부, email: {gabrielyoon7, yeonsu0329, sunxy5508, thdud0417}@kgu.ac.kr

** 소속: 경기대학교 AI컴퓨터공학부 교수, email: jhahn@kgu.ac.kr

는 시스템을 구현하고, 충전기의 요일/시간대별 사용 통계를 생성 및 관리하는 자동화 시스템을 신속하게 다루는 방법을 연구하려고 한다.

II. 전기차 충전소 사용 로그 수집 시스템

2.1 데이터 수집기의 전체 구조

데이터 수집기는 세 부분으로 구성되어있다. 외부로부터 데이터를 받아들이는 수신부, 받은 데이터를 가공하여 저장하는 저장부, 수신부로부터 받은 데이터 중 사용 중인 충전기의 사용 여부를 요일/시간대별로 기록하는 로그부가 있다.

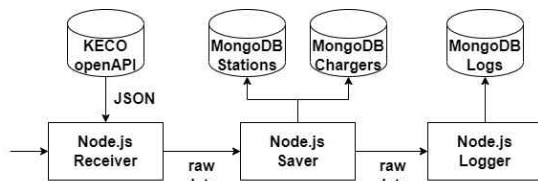


그림 1. 데이터 관리 시스템
Fig. 1. Data Management System

2.2 전기차 충전소 및 충전기 데이터 정보 수집

KECO API로부터 전국에서 데이터를 수집[2]해보면 충전기가 약 13만 건을 얻을 수 있고, 이 중 충전소를 집합 처리하면 약 4만여 장소가 존재함을 알 수 있다. 충전소와 충전기 각각이 필요한 정보가 무엇인지, 어떤 정보가 중복되는 것을 방지해야 할지 등을 고려하여 충전소와 충전기로 데이터를 나누어 저장해야 한다.

```
{
  "id": "ME00010201",
  "chgerId": "01",
  "chgerType": "06",
  "lastTsd": "20220811111543",
  "lastTsd": "20220811111543",
  "method": "단독",
  "nowTsd": "",
  "output": "50",
  "stat": "g",
  "statId": "ME000102",
  "statNm": "충소2동 주민센터",
  "statUpDt": "20220811124810",
  "id": "ME000102",
  "addr": "충청남도 천안시 동남구 천성1길 24",
  "api": "keco",
  "bom": "충청남도",
  "buscall": "1661-9480",
  "busid": "ME",
  "busNm": "충전부",
  "date": "2022",
  "delv": "N",
  "distance": 0,
  "kind": "AP",
  "kindDetail": "A002",
  "lat": "35.639283",
  "limitDetail": "",
  "limitNm": "N",
  "lng": "129.340893",
  "location": "null",
  "note": "",
  "parkingFee": "N",
  "statId": "ME000102",
  "statNm": "충소2동 주민센터",
  "statUp": "충소2동 주민센터",
  "statUp": "24시간 이용가능",
  "tcode": "13",
  "tcode": "31200"
```

그림 2. 충전기/충전소 데이터 추출
Fig. 2. Charger/Station Data Extraction

이렇게 가공한 데이터를 MongoDB에 저장해야 하는데, 일반적인 저장 방식을 사용하게 되면 수십만 건의 신규 데이터를 처리할 때 오랜 시간이 걸리게 된다는 문제가 있다. 하지만 후술할 방법으로

이 시간을 단축할 수 있다.

2.3 전기차 충전소 사용 로그 생성 및 갱신

데이터를 수집하며 얻어 낸 데이터 중 사용 중인 충전기 데이터들을 따로 모아서 이력을 남기게 되면 사용자에게 유용한 데이터를 제공할 수 있을 것으로 기대했다.

단순하게 수집한 데이터를 모두 그대로 두고, 새로 생성된 데이터를 저장하는 단순한 방식으로 관리하는 것이 굉장히 편리하겠지만, 수집 사이클 당 생성되는 도큐먼트의 개수가 최소 30만 건에 달한다는 문제가 있어, 실시간 서비스를 하는 경우 저장 공간이 금방 부족해질 것임을 예상할 수 있다. 따라서 적은 용량으로 사용 로그를 기록할 필요가 있었다.

기본적으로 KECO API는 충전기 데이터만을 제공하고 있기에 충전기를 기준으로 로그를 생성하는 것이 시스템 관리 측면에서 유리하다고 생각했고, 하나의 충전소 안에 여러 형태의 충전기가 있다는 것을 고려할 때도 충전기 별로 로그를 저장해야 한다고 판단했다. 매주 특정 요일, 충전기의 시간대/요일별 사용 여부를 기록할 수 있는 로그가 생성된다. 주차 별로 생성된 사용 로그를 필요한 만큼 호출하여 누적시키면 특정 기간에 어떤 충전소가 얼마나 혼잡했는지를 알 수 있게 될 것이다.

이 방법에서도 한 번에 수십만 건의 데이터를 신규 등록 및 갱신해야 하는 문제가 있어 후술할 방법으로 시간을 단축했다.

```
{
  "id": "202236ME00010201",
  "chgerId": "01",
  "logs": {
    "mon": {
      "tue": {
        "wed": {
          "thu": {
            "fri": {
              "sat": {
                "sun": {
                  "region": "충청남도",
                  "statId": "ME000102",
                  "week": "202236"
                }
              }
            }
          }
        }
      }
    }
  }
}
```

그림 3. 충전기 사용 로그
Fig. 3. Log of Charger Usage

2.4 대량의 데이터를 벌크로 저장하기

이 프로그램에서 수집 사이클 당 처리해야 하는 데이터가 수십만 개에 달하므로 데이터를 저장하고 갱신하는 과정에서 너무 오랜 시간을 소요하면 실시간 서비스의 의미가 퇴색될 수 있다.

수집 및 가공을 통해 나온 데이터들은 모두 mongoDB로 신규 저장하거나, 갱신한다. 이 데이터들을 일반적인 방식으로 처리하면 수십만 건의 데이터를 DBMS로 일일이 보내야 하므로 속도가 떨어진다. 따라서 mongoDB의 벌크 기능을 활용하게 되면 데이터 처리를 개별로 하는 것이 아닌, 벌크 단위로 처리를 요청할 수 있어 처리 속도가 대폭 개선된다.[3]

2.5 비동기적인 데이터 처리

처음 데이터 수집기를 설계할 때는 Java로 작성하였고, KECO API로부터 전국 각지의 데이터를 순차적으로 수집하는 방식을 택하였다. 이전 지역의 데이터 수집·정제·저장이 끝나기 전까지 다음 지역을 처리할 수 없었다. 프로그램 설계가 간단하였지만, 속도가 너무 느려지는 문제가 있었다. 스레드 기능을 활용하면 처리 순서를 일부 개선할 수 있었으나, 비동기 처리를 손쉽게 지원하는 Node.js로 작성하여 수집을 진행했다. 데이터 처리 과정을 구현해두고, 해당 과정을 비동기 처리에 맡기게 되면 CPU가 해당 과정을 지역별로 동시에 수집 및 가공을 처리해주게 된다. Node.js의 기본이 되는 JavaScript는 싱글 스레드로 동작하지만, 비동기 처리가 필요한 순간에는 작업을 libuv에 맡겨 멀티 스레딩이 가능해지게 된다. 해당 작업이 특정 지역의 데이터 수집·정제·저장이고, 여러 지역을 동시에 명령을 내린다면 거의 동시에 작업이 진행되게 된다는 것이다.

다만, 지역마다 충전기 수가 다르므로 반드시 처리 속도가 차이 난다. 따라서 무한히 데이터를 처리하도록 명령하되, JavaScript에서 지원하는 Promise, ES7에서 지원하는 async/await을 사용하여 모든 지역의 처리가 끝날 때만 다음 루프로 동작하도록 하여 데이터 수집의 불균형을 제어할 수 있었다.

III. 실험

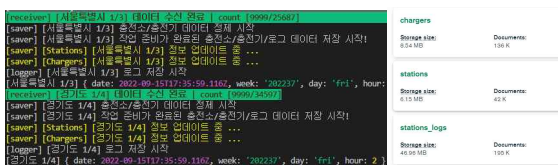


그림 4. 데이터 관리 시스템

Figure 4. Data Management System

데이터 관리는 자동화가 되어있다. 약 30만 건의 전국 데이터를 처리하는데 걸리는 시간은 약 300초가 걸린다. 이 방식은 앞선 설명대로 비동기적인 수집 방식과 벌크 기능을 모두 적용한 결과이다.

동기적인 처리와 일반 저장 방식을 적용했던 초기 모델의 경우에는 약 1200초의 많은 시간이 소요되었던 것에 비하면 굉장히 안정적인 속도로 데이터 관리가 되고 있음을 알 수 있었다.

IV. 결론 및 제언

본 연구에서 전기차 충전소가 아닌 충전기 데이터를 기준으로 사용량 통계를 수집하면서 충전소와 충전기 데이터도 실시간으로 생성해주는 시스템을 개발했다. KECO API를 통해 다루는 데이터는 작업당 수십만 개에 달하지만, 벌크쓰기와 비동기 처리를 통해 순차적인 방식보다 빠르게 갱신할 수 있게 되었다. 실시간으로 수집되는 데이터와 사용 누적 통계 데이터로 요일 및 시간대별 이용률을 분석하여 여유로운 충전 시간대를 파악할 수 있게 될 것으로 기대한다.

감사의 글

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학지원사업의 연구결과로 수행되었음(2021-0-01393).

References

- [1] 한국환경공단, "전기자동차 충전소 정보 조회 서비스", 공공데이터 포털, <https://www.data.go.kr/tcs/dss/selectApiDataDetailView.do?publicDataPk=15076352>, 2022.08.
- [2] 남궁주홍, 손시운, 문양세 "공공 데이터 오픈 API를 사용한 실시간 데이터 수집 시스템", 한국정보과학회 2017 한국소프트웨어종합학술대회 논문집, 212 - 214 (3page), 2017.12.
- [3] 가태한, 이석훈, "라이프로그 수집 및 관리를 위한 MSSQL과 MongoDB의 성능 비교 평가", 한국정보기술학회 2018년도 공동학술대회 및 대학생논문경진대회, 376 - 377 (2page), 2018.11.