

# 빅데이터 앱 컨테이너 구조의 설계 및 구현\*

김주민, 임병돈, 장병철, 김예림, 유다금, 정의현  
안양대학교 컴퓨터학과

e-mail:{4qazwsx, qudehs227, bn8090, okyeahlim, manysilk}@naver.com,  
jung@anyang.ac.kr

## A Design and Implementation of Bigdata App Container Structure

Jumin Kim, Byeongdon Lim, Byeongcheol Jang, Yelim Kim, Dakeum Yoo,  
Euihyun Jung

Dept of Computer Science, Anyang University

### 요 약

기존의 빅데이터 분석은 코드주도(code-driven) 방식과 쿼리주도(query-driven) 방식의 두 가지 방식만이 있었다. 코드주도 방식은 분석력을 극대화할 수 있지만 사용성이 떨어지고, 쿼리주도 방식은 그 반대로 사용성이 좋지만 분석력에 한계가 있었다. 본 연구에서는 이 두 가지 방식의 장단점을 혼합하여, 분석력을 극대화하면서도 사용성이 높은 방식인 앱 컨테이너 구조를 제안하고 구현하였다. 제안된 앱 컨테이너는 독립된 형태로 운용이 가능하여 여러 형태의 스파크(Spark) 시스템에서 활용이 가능하다.

### 1. 서론

빅데이터(Bigdata)가 소개된 이래로 정보통신 분야는 물론이고, 의료, 경제, 제조 분야를 포함하는 산업계 전 분야에 걸쳐서 빅데이터를 적극적으로 활용하기 위한 여러 노력이 있어왔다[1]. 국내에서는 교통이나 카드 구매 분석을 통한 상권 분석 등에서 효과를 본 사례가 보고되었고, 이러한 성공사례는 계속해서 늘어날 전망이다[2].

빅데이터의 성공에는 데이터 소스의 증가, 데이터 분석 기법의 성숙 등의 여러 요소들이 있지만, 저렴한 하드웨어에서 대용량 데이터를 분석할 수 있는 하둡(Hadoop) 생태계(ecosystem)의 다양한 기술들의 대두가 가장 큰 동력을 부정할 수 없다. 특히 최근에 각광받고 있는 스파크(Spark)[3]는 메모리 기반의 분석 시스템으로 기존 하둡을 대체하고, 배치 분석과 스트리밍 분석을 모두 지원할 수 있는 기술로 인정받고 있다.

스파크가 여러 면에서 빅데이터를 선도할 수 있는 기술이기는 하나, 스파크를 포함한 기존 빅데이터 분석은 크게 빅데이터 분석 코드를 프로그래밍 언어로 작성하는 코드주도(code-driven) 방식과 쿼리(query) 언어를 이용하는 쿼리주도(query-driven) 방식의 사용으로 한정된다. 코드주도 방식은 스파크의 모든 분석 기능을 100% 활용할 수 있는 장점이 있으나, 프로그래밍을 하지 못하는 다수의 데이터 분석가들에게는 접근이 어려운 단점이 있다. 이에 비해 쿼리주도 방식은 쿼리를 이용하여 분석을 하기 때문에 초보자라도 쉽게 데이터 분석이 가능한 장점을 갖고 있다.

그러나 쿼리주도 방식은 쿼리 언어가 제공하는 기능 내에서만 분석이 가능하기 때문에 스파크의 모든 분석 기능을 활용할 수는 없다.

본 논문에서는 이러한 기존 빅데이터 분석 방식인 코드주도 방식과 쿼리주도 방식을 결합한 새로운 사용방법인 앱 컨테이너(App Container)방식을 제안하였다. 먼저 데이터분석가가 필요한 분석기능이 있으면, 프로그래머에게 분석기능을 갖고 있는 빅데이터앱을 개발해줄 것을 요청한다. 이렇게 개발된 빅데이터앱은 앱 컨테이너에 탑재되고, 그 후에 여러 데이터분석가들이 마치 스마트폰의 앱처럼 쉽게 사용할 수 있게 된다. 또한 제안된 앱 컨테이너는 API 기능을 제공하기 때문에, 서드파티(3rd party) 개발자들이 쉽게 빅데이터앱을 이용한 다양한 응용 개발이 가능할 것으로 기대된다.

### 2. 제안 구조

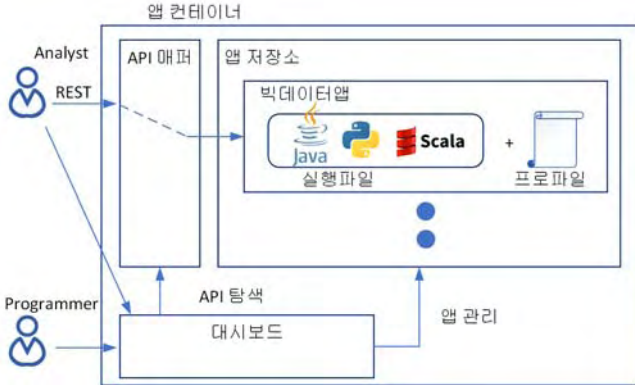
본 시스템은 그림 1과 같이 앱 저장소, API 매핑, 대시보드의 3가지 요소로 구성되어 있다. 먼저, 모든 빅데이터앱들은 실행파일과 API 구성 메타데이터를 담고 있는 프로파일(profile)로 구성된 패키지 형태를 갖는다. 프로파일은 빅데이터앱을 호출하기 위한 API의 이름, 인자(parameter) 등의 정보를 갖고 있어, 앱 컨테이너가 동적으로 API를 구성할 수 있게 한다. 그리고 실행파일은 스파크 API를 이용하는 Java, Python, Scala가 지원된다.

대시보드는 개발자와 일반 데이터 분석가가 모두 사용할 수 있는데, 개발자는 빅데이터앱을 앱 컨테이너에 탑재하는 용도로 활용한다. 데이터 분석가는 이와는 다르게 탑재된 빅데이터앱을 검색하고, 자신이 분석하려는 데이터를

\* 본 연구는 한국정보화진흥원(NIA)의 미래네트워크선도 시험망(KOREN) 사업 지원과제의 연구결과로 수행되었음 (18-951-00-001).

연결한 후, 빅데이터앱의 실행을 할 수 있는 기능을 제공한다.

API 매핑은 REST를 이용한 API 호출이나 대시보드를 통한 요청을 받아서 실제 빅데이터앱의 실행을 처리해주는 역할을 담당한다. API 매핑의 API는 빅데이터앱의 프로파일에서 자동으로 생성된다.



(그림 1) 앱 컨테이너의 구조

앱 프로파일의 구조는 다음 표 1과 같다. 앱 프로파일은 JSON 형식으로 구성되어 있으며, 상기된 바와 같이 빅데이터앱의 실행 파일과 같이 탑재가 된다. 프로파일은 앱에 대한 설명 부분과 앱의 실행에 관한 부분으로 구성이 되는데, 앱의 실행 부분에서 특히 callback 부분은 다른 메타데이터들과 구별되는 요소이다. 빅데이터 분석은 다른 API 호출과 달리 실행시간이 오래 걸리기 때문에 바로 응답을 제공하는 동기적(synchronous) API 호출의 지원이 어렵다. 따라서 본 논문에서는 분석이 종료된 후에 callback으로 지정된 방식에 맞춰서, 결과를 보고하는 방식을 제시하였다. 현재 email과 Slack 채널을 callback으로 지원하고 있다.

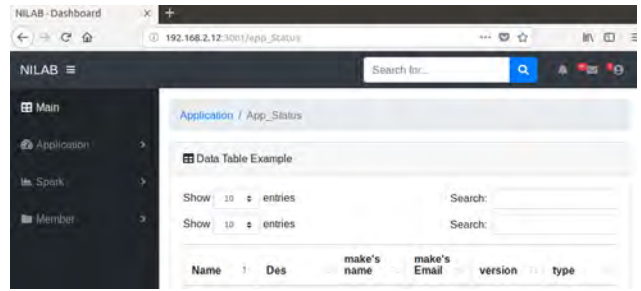
[표 1] 프로파일 메타데이터 구조

요소	설명
uid	앱의 고유 아이디
name	앱의 이름
version	앱의 버전
desc	앱에 대한 설명
lang	앱 개발 언어
signature	URI 템플릿[4]을 이용한 API 호출 템플릿
parameters	앱의 파라미터들의 구성
callback	콜백 방식 처리

### 3. 제안 시스템의 구현 및 평가

제안된 시스템의 구현은 스파크 클러스터를 구축하고, 그 위에 Node.js로 앱 컨테이너를 구성하였다. 기본적으로 제안된 앱 컨테이너는 스파크 시스템 위에서 운용되기 때문에 스파크 운영이 필요하기는 하지만, 앱 컨테이너 자체는 Node.js 실행 프로그램이기 때문에 스파크 클러스터와 단독모드에서 모두 운용이 가능하다. 메타데이터와 실행로그의 저장은 MongoDB를 이용하여 처리하였다. 그림 2는

앱 컨테이너의 대시보드를 보여주고 있다.



(그림 2) 대시보드

본 논문이 제안한 앱 컨테이너 구조의 효용을 파악하기 위해 기존의 코드주도 방식 및 쿼리주도 방식과 여러 측면에서 비교를 하였다. 표 2는 확장성(extensibility), 기능성(capability), 사용성(usability), 습득용이성(learning curve)로 요소를 지정하여 비교한 표이다. 코드주도 방식은 확장성과 기능성에서 가장 우수하지만, 사용성과 습득용이성은 좋은 평가를 받기 어렵다. 쿼리주도 방식은 코드주도 방식의 장점과 약점을 반대로 갖고 있다. 이에 비해서 앱 컨테이너 방식은 두 가지의 장점과 약점을 섞어서, 적당한 수준의 평가를 보여준다. 특히 앱 컨테이너의 실행은 코드가 중심이기 때문에 코드주도 방식과 마찬가지로 스파크의 분석력을 최대한 이용할 수 있다.

[표 2] 기존방식과의 비교

요소	코드주도	쿼리주도	앱 컨테이너
확장성	매우우수	나쁨	우수
기능성	매우우수	보통	우수
사용성	나쁨	매우우수	우수
습득용이성	어려움	쉬움	보통

### 4. 결론

본 연구에서는 기존 빅데이터 분석방식인 코드주도 방식과 쿼리주도 방식을 결합하여 새로운 분석방식인 앱 컨테이너 구조를 제시하였다. 이를 통해 스파크 시스템의 분석력을 최대화하면서도 사용성을 높이는 성과를 얻을 수 있었다.

### 참고문헌

[1] Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. *Mobile networks and applications*, 19(2), 171-209.

[2] 이성훈, & 이동우. (2013). 빅데이터의 국내·외 활용 고찰 및 시사점. *디지털융복합연구*, 11(2), 229-233.

[3] Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., ... & Ghodsi, A. (2016). Apache spark: a unified engine for big data processing. *Communications of the ACM*, 59(11), 56-65.

[4] Gregorio, J., Fielding, R., Hadley, M., Nottingham, M., & Orchard, D. (2012). Uri template (No. RFC 6570).