

# Node.js의 Async모듈을 이용한 비동기적 웹툰 추천 시스템

이건호\*, 박진수\*\*, 박두순\*

\*순천향대학교 컴퓨터소프트웨어공학과

\*\*순천향대학교 웰니스코칭 서비스연구센터

cjdtjcjdtj@likelion.org

## Asynchronous Webtoon Recommend System Using 'Async' Module in Node.js

Keon-Ho Lee\*, Jin-Soo Park\*\*, Doo-Soon Park\*

\*Dept of Computer Software Engineering, SoonChunHyang University

\*\*Wellness Coaching Service Research Center(C-ITRC)

### 요 약

웹툰은 플랫폼 별 웹툰 데이터가 매우 다양하기 때문에, 데이터 수집이 용이하여 추천 시스템에 적합하다. 반대로 사용자들은 너무 많은 웹툰이 있어 자신에게 맞고 즐길 수 있는 웹툰을 일일이 찾아보기에는 어려움이 있다. 본 논문에서는 이러한 사용자들의 불편함을 해소하고자 웹툰 데이터를 기반으로 Node.js Async 모듈을 사용하여 비동기적 웹툰 추천 시스템을 개발하고자 한다.

### 1. 서론

하루가 다르게 새로이 등장하는 각종 웹 기술과 함께 데스크톱에서 모바일까지 아우르는 시스템 등, IT 기술은 빠르게 발전하고 있다. IT 기술의 핵심적 기술인 소프트웨어 개발에서도 많은 변화가 계속되고 있다. 최근 소프트웨어 개발은 하나의 언어로 개발하기 보다는 여러 가지 라이브러리와 프레임워크 등 오픈소스를 활용하여 개발하는 추세이다. 하지만 이 프레임워크와 라이브러리도 결국 기반 언어를 가지고 있기 때문에, 프로그래밍 언어 역시 소프트웨어 개발에 있어 많은 중요요소 중 하나이다. 현재 전 세계적으로 가장 많이 쓰이는 언어는 자바스크립트이다. (그림 1)은 세계적인 개발자 커뮤니티 Github과 StackOverFlow에서 쓰이는 언어들을 조사하여 순위를 집계한 RedMonk의 프로그래밍 선호도 순위이다.



(그림 1) RedMonk 프로그래밍 언어 순위[1].

본 논문에서는 (그림 1)을 참고하여, 가장 많이 사용되는

자바스크립트언어 기반인 Node.js 라이브러리를 활용한 추천 시스템을 개발하고자 한다.

웹툰은 영화와 드라마와 달리 제작비가 약소하고, 많은 작가들이 플랫폼에 연계하는 방식이다. 그래서 데이터가 지속적으로 쌓이게 되고, 플랫폼 별 웹툰이 매우 다양하기 때문에, 데이터 수집이 용이하여 추천 시스템에 적합하다. 반대로 사용자들은 너무 많은 웹툰이 있어 자신에게 맞고 즐길 수 있는 웹툰을 일일이 찾아보기에는 어려움이 있다. 본 논문에서는 이러한 사용자들의 불편함을 해소하고자 웹툰 데이터를 기반으로 추천 시스템을 개발하려 한다.

웹툰을 기반으로 개인화 추천시스템을 개발하기 위해서는 많은 웹툰 데이터를 저장해야 하고, 이렇게 저장된 데이터를 빠르게 처리 할 수 있어야 한다. 따라서 기존 RDBMS가 아닌 많은 양의 데이터를 처리하기 적합한 NOSQL(Not-Only SQL) 기반이고 자바스크립트와 호환성이 좋은 몽고DB를 사용하여 시스템을 구성하였다.

자바스크립트는 싱글 스레드 기반으로 프로세스를 처리하게 되는데 기본적으로 동기 방식으로 진행된다. 동기 방식이란, 작업이 들어온 순서에 맞게 하나씩 순서대로 진행되는 것이다. 순서에 맞게 진행되는 장점이 있지만 여러 가지의 일을 동시에 처리할 수 없다. 하지만 비동기 방식은 여러 가지 일을 한꺼번에 처리할 수 있어, 많은 양의 데이터를 저장하고, 처리할 때 동기 방식보다 빠르다는 장점이 있다. 이러한 장점을 바탕으로 본 논문에서는 Node.js에서 가장 많이 쓰이는 비동기 모듈 Async를 이용하여 사용자들에게 가장 알맞은 웹툰을 빠르게 추천한다.

※ 본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 대학ICT연구센터육성지원사업의 연구결과로 수행되었음 (IITP-2017-2014-0-00720-002)

## 2. 웹툰 추천 시스템의 구성

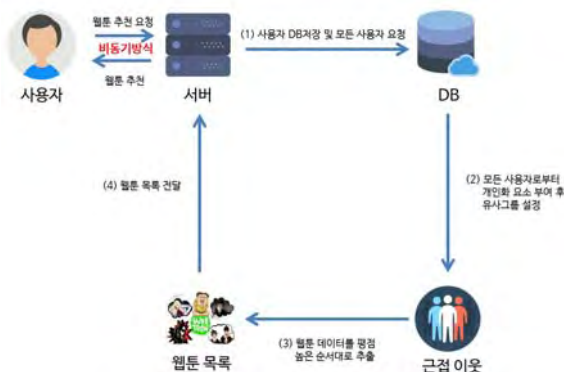
일반적으로 추천 서비스에서는 데이터를 추천하기 위해 사용자가 선호하는 데이터와 데이터의 정보를 제공한다. 추천시스템에 사용되는 대표적인 방법으로는 협업필터링 방법과 내용 기반 필터링 방법이 있다. 본 논문에서는 협업필터링 방법을 적용하고 사용자의 개인화 요소와 웹툰의 데이터에 대한 선호도를 분석하여 웹툰을 추천하는 방법을 선택했다[2].

이 방법을 구현하기 위해서는 많은 웹툰 데이터가 필요하다. 본 논문에서는 웹툰 데이터를 구성하기 위하여 Node.js의 cherrio 모듈을 사용해 웹툰 플랫폼에 있는 데이터를 가져와 DB에 저장하였다. (그림 2)는 웹툰 데이터 추출 시나리오이다.



(그림 2) 웹툰 데이터 추출 시나리오

(그림 2)의 시나리오에서는 (1) 서버에서 웹툰 데이터를 가지고 있는 플랫폼 홈페이지로 url을 요청하게 된다. (2) Node.js의 cherrio 모듈은 요청된 url에 있는 모든 HTML 및 JSON 객체 데이터들을 서버로 가져오게 된다. (3) 가져온 데이터들을 바탕으로 추천 시스템에 필요한 웹툰 데이터(웹툰 제목, 작가, 웹툰 이미지, 평점 등)를 파싱하여 데이터베이스에 저장한다. 이렇게 저장된 웹툰 데이터베이스를 바탕으로 추천 시스템이 구성되었는데 (그림 3)은 본 논문에서 구현하게 될 웹툰 추천 시나리오이다.



(그림 3) 웹툰 추천 시나리오

(그림 3)의 추천 시나리오에서 사용자가 추천을 요청하기 전에 회원가입을 해야 한다. 사용자가 회원가입 시 입력하게 되는 나이, 직업, 선호 카테고리, 선호 플랫폼 등은 추천 시 개인화 요소로 사용된다. 먼저 (1) 사용자가 회원가입을 하게 되면 사용자의 데이터들이 DB로 저장되는데 추천을 진행하기 위해서는 다른 사용자들의 데이터를 가져와야 한다. 따라서 데이터베이스에 모든 사용자들의 데이터베이스를 요청하게 된다. (2) 모든 사용자의 개인화 요소 데이터와 현재 추천을 원하는 사용자의 개인화 요소 데이터들을 바탕으로 협업필터링을 이용, 최근접 이웃을 구성한다. 여기서 협업 필터링이란, 사용자들의 선호도와 관심 표현을 바탕으로 선호도, 관심도가 비슷한 사용자들을 식별해 내는 방법이다. 과거에 이용한 콘텐츠가 비슷하다면 사용자간에 유사한 성향을 가지고 있다고 판단하며, 그 근거를 토대로 추천하는 방식이다.[3]. (3) 구성된 최근접 이웃의 웹툰 데이터를 추출한 뒤 사용자가 이미 본 웹툰은 필터링하여 사용자가 보지 않은 웹툰 중 포털 사이트에서 가져온 평점 데이터가 높은 목록을 전달하여 사용자에게 웹툰 목록을 추천한다.

본 논문에서는 사용되는 개인화 요소는 사람의 생활환경을 가지고 있는 정보인 나이, 직업, 성별을 우선적으로 선택하였다. 나이는 10세 단위별로 구분하였는데 그 이유는 각 세대 간의 문화 습관이나 생활에서 차이를 보이기 때문이다. 마지막으로 웹툰에 관련된 두 가지 개인화요소인 카테고리 및 선호 웹툰 플랫폼을 추가하였는데 사용자마다 선호하는 장르가 너무 다르기 때문에 개인화 요소로 사용하기 적합하였고, 각 플랫폼 별(네이버, 다음, 카카오톡 등) 웹툰 분위기가 달라 이것 역시 추가하였다. 또한 직업은 통계청의 자료를 기준으로 대분류 항목을 구성하였고 웹툰 카테고리는 가장 큰 웹툰 플랫폼인 네이버에서 참고하였다. 개인화 요소의 분류 결과는 (표 1)이다.

(표 1) 사용자 개인화 요소의 분류[3].

Index	Age	Job	Sex	Category	Platform
1	10대	학생	남성	에피소드	네이버
2	20대	관리자	여성	온니버스	다음
3	30대	전문가		스토리	카카오페이지
4	40대	사무직		일상	레전코믹스
5	50대	서비스업		개그	탐툰
6	기타	판매직		판타지	
7		농량어업 종사자		액션	
8		기능직		드라마	
9		기계조작 및 조립 종사자		순정	
10		노무직		감성	
11		군인		스릴러	
12				시대극	
13				스포츠	

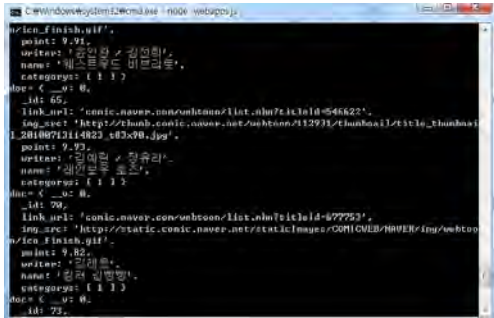
## 3. 웹툰 추천 시스템의 구현

웹툰 추천 시스템을 구현하기 위해서는 기본적인 서버와 데이터베이스의 개발환경이 갖춰져 있어야 한다. (표 2)는 웹툰 추천 시스템의 개발 환경이다.

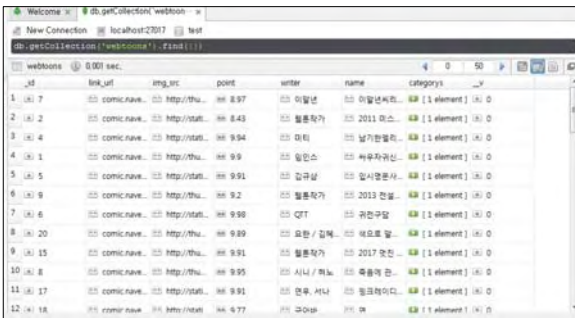
(표 2) 웹툰 추천 시스템 개발 환경

언어	Javascript
라이브러리	Node.js 6.4
프레임워크	Express 3.0
DBMS	MongoDB
서버	Local(PM2)

우선적으로 추천 시스템을 구성하기 위해서는 웹툰 데이터베이스를 가지고 있어야 한다. (그림 4)는 (그림 2)에 있는 웹툰 데이터 추출 시나리오를 바탕으로 구현한 웹툰 추출 기능으로 웹툰 데이터들이 저장되고 있는 모습이다. 또한 (그림 5)는 몽고DB에 저장된 스키마이다.

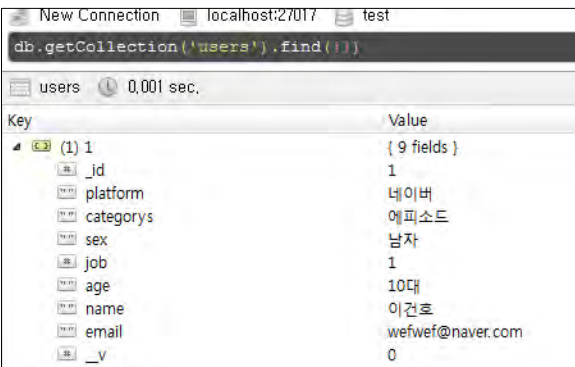


(그림 4) 웹툰 데이터 추출 로그



(그림 5) 몽고DB에 저장된 웹툰 DB

(그림 6)은 개인화 요소를 회원가입 시 사용자에게 받아 서버에 전송하여 몽고DB에 저장한 사용자 데이터이다.



(그림 6) 몽고DB에 저장된 사용자 DB

(그림 6)의 사용자 데이터베이스를 이용하여 근접 이웃을 구성할 수 있는데 개인화 요소, 가중치에 따라 모두 근접 이웃의 구성이 달라진다. 이 때 사용자의 유사도를 나이, 직업, 성별, 카테고리, 선호 플랫폼 다섯 가지로 Jaccard Similarity를 이용해 근접 이웃을 구성하게 되는데, 모든 사용자의 데이터와 비교하게 된다. 사용자의 데이터가 많

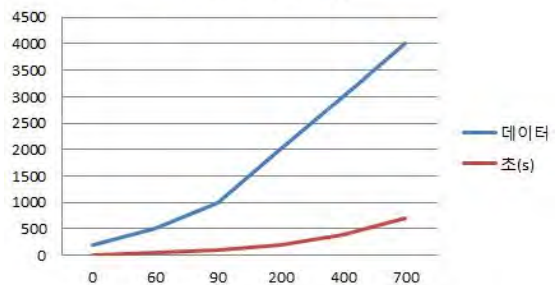
으면 많을수록 속도가 느려지게 되는데, 본 논문에서는 이를 해결하기 위해 Node.js의 Async 모듈을 사용하여 구성한다. (그림 7)은 Async모듈의 ForEach 함수를 이용한 비동기적 추천 시스템의 핵심 코드이다.

```
router.post('/recommend', function(req, res, next) {
  User.find({_id: req.params.id}).function(err, myuser) {
    User.find({}, function(err, alluser) {
      allUser.forEach(function(err, user) {
        if(myuser == user)
          callback();
        var myJaccRelationship = new JaccadRelationship();
        // 근접이웃 스키마 넣을 곳 처리
      });
    });
  });
});
```

(그림 7) 비동기 추천 시스템 핵심 코드

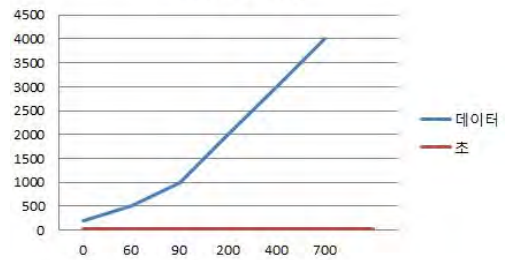
(그림 7)의 코드를 보게 되면, 사용자의 id값을 받아와 현재 요청한 사용자의 객체를 myuser 변수에 가져온다. 그 후 전체 사용자의 객체 배열을 alluser 변수에 가져와 현재 사용자와 전체 사용자 각각 비교하는 forEach 함수가 동작하게 된다. 반복문과 다르게 순차적으로 비교하지 않고, 한꺼번에 함수를 실행시켜 함수가 콜백되었을 때만 알려준다. (그림 8)은 일반적인 방법으로 전체 사용자와의 유사도를 비교하여 근접이웃을 구성한 시간 그래프이고, (그림 9)는 비동기적 웹툰 추천시스템을 사용하여 비교한 시간 그래프이다.

일반적인 경우



(그림 8) 일반적인 반복문 시스템

비동기 경우



(그림 9) Node.js Async 비동기적 시스템

(그림 8)과 (그림 9)의 그래프를 보게 되면 반복문보다 데이터가 많아질수록 비동기적 시스템이 빠르다는 것을 알 수 있다. 또한 비동기로 시스템을 구성하면 한꺼번에 데이터를 처리하여 시간 변화가 거의 없는 지만 이 경우에 단점은 저장될 때 데이터가 순차적으로 저장되지 않는 점이다. (그림 10)은 이와 같은 방법으로 user\_1이 추천을 요청하였을 때 user\_20과 비교한 데이터를 생성한 스키마이다.

(2) 2	{ 8 fields }	Object
id	2	Int32
recommend	{ 8 fields }	Object
_v	0	Int32
name	씨유자귀신아	String
writer	임인스	String
point	9.9	Double
img_src	http://thumb.comic.naver.net/webtoon/350217/thu...	String
link_url	comic.naver.com/webtoon/list.nhn?titleId=23182	String
id	1	Int32
categorys	[ 1 element ]	Array
similarity	0.6667	Double
intersection	4	Int32
union	6	Int32
user_two_id	29	Int32
user_one_id	1	Int32
_v	0	Int32

(그림 10) user\_1과 user\_29의 유사도 비교 스키마

(그림 10)의 스키마를 보게 되면 recommend 값에는 웹툰 데이터가, categorys 값에는 웹툰 카테고리 값까지 같이 저장된 것을 볼 수 있다. 이렇게 모든 사용자와의 비교가 끝나게 되면 user\_1에 대한 유사도와 근접이웃이 구성되어 (표 3)과 같이 결과가 나오게 된다. 본 논문에서의 개인화 요소인 나이, 성별, 직업, 카테고리, 플랫폼에 각각 나이(20%), 성별(10%), 직업(10%), 카테고리(20%), 플랫폼(40%)으로 가중치를 두었는데 그 이유는 개인화 요소를 선택한 이유와 유사하다. 사용자들이 살아온 성장환경 및 주변 환경을 반영한 나이, 성별, 직업에 비교적 적게 가중치를 두었으며 웹툰의 장르와 플랫폼에 따라 성향이 많이 다르기 때문에 비교적 높게 가중치를 주었다. (표 3)은 이와 같은 가중치로 구성된 근접이웃이다.

(표 3) user\_1에 대한 유사도와 근접이웃 구성

ID	Union	Intersection	Similarity	Recommand
user_1	x	x	x	x
user_29	1.1	0.9	0.8182	신의탑
user_127	1.3	0.8	0.6667	노블레스
user_430	1.3	0.8	0.6667	기기괴괴
user_52	1.3	0.7	0.5385	조우
user_76	1.4	0.7	0.5385	체이서
user_10	1.4	0.6	0.4286	버스트
user_7	1.4	0.6	0.4286	연애혁명
user_34	1.4	0.6	0.4286	여중생A

(표 3)과 같이 구성된 근접이웃을 바탕으로 최근접 사용자를 추출하게 되는데 (그림 11)는 추출된 근접 사용자와 그 사용자들이 추천한 웹툰이다.

(1) 7	{ 8 fields }	
id	7	Object
link_url	comic.naver.com/webtoon/list.nhn?titleId=10375	String
img_src	http://thumb.comic.naver.net/webtoon/557672/t	String
point	8.97	Double
writer	이달년	String
name	이달년씨리즈	String
categorys	[ 1 element ]	Array
_v	0	Int32
(2) 2	{ 8 fields }	
id	2	Object
link_url	comic.naver.com/webtoon/list.nhn?titleId=35021	String
img_src	http://static.comic.naver.net/staticimages/COMIC	String
point	8.43	Double
writer	필론작가	String
name	2011 마스터리 단편	String
categorys	[ 1 element ]	Array
_v	0	Int32

(그림 11) 추출된 사용자와 웹툰 목록 스키마

마지막으로 작성된 근접 그룹이 추천한 웹툰을 추천한다. (그림 12)은 10대, 남자, 학생, 에피소드, 네이버라는 개인화 요소를 두었을 때 추천된 웹툰 결과 화면이고 (그림 13)은 20대, 여자, 사무직, 유니버스, 네이버라는 개인화 요소를 두었을 때 추천된 웹툰 결과 화면이다.



(그림 12) 10대, 남자, 학생, 에피소드, 네이버라는 개인화 요소를 두었을 때 추천된 웹툰 결과



(그림 13) 20대, 여자, 사무직, 유니버스, 네이버라는 개인화 요소를 두었을 때 추천된 웹툰 결과

(그림 12), (그림 13)과 같이 결과 화면에서는 추천된 웹툰 제목, 웹툰 배너 이미지, 웹툰 제목, 웹툰 작가, 평점 등 보여지고 있다. 또한 사용자 편의를 위하여 웹툰 카드를 클릭하게 되면, 웹툰의 상세정보 링크로 이동하게 된다.

#### 4. 결론

본 논문에서는 수많은 웹툰 정보와 사용자 정보를 어떻게 하면 효과적이고 빠르게 처리할 수 있을까 해서 Node.js와 Express환경, Async모듈을 사용하여 비동기 시스템을 구현하였다.

다음으로는 Async Parallel이라는 기능의 병렬 데이터 처리방식을 이용한 시스템을 구현해보고, 비동기 처리 방식과 병렬처리 방식일 때의 차이점이 무엇인지를 알 수 있을 것이고, 이에 따른 F1 measure, Precision, Recall 관점에서 성능평가가 이루어져야 할 것이다.

#### 참고문헌

[1] RedMonk, "Popularity Rank on Github", <http://redmonk.com/sograzy/category/programming-languages>, 2016

[2] 이진호, 박두순, "클라우드 컴퓨팅에서 구축한 협업필터링 기반 웹툰 추천 시스템", 한국정보처리학회 춘계 학술대회 발표 논문집, 동국대학교, pp. 451-454, 2016.05

[3] "한국 표준 직업분류 표-대분류", 통계청, 2015.

[4] 심대수, 박진수, 박두순, "오피니언 마이닝과 협업필터링을 이용한 웹툰 추천시스템", 한국정보처리학회 추계학술발표대회 논문집, 한라대학교, pp. 522-524, 2017.05