

Node.js를 리용한 WebGIS봉사기의 설계와 실현

김유성, 강영호

WebGIS는 최신Web기술에 의하여 계속 개선되면서 리용자들에게 쓸모있고 정확한 공간정보를 제공하고있다. 최근에 WebGIS봉사기에서는 의뢰기의 요청이 신속히 처리되고 공간정보에 대한 봉사능력이 개선되고있다.[1, 3]

Web상에서 공간자료를 처리하고 말단에 전송하는것은 전용WebGIS봉사기에 의하여 진행되는데 만약 WebGIS봉사기의 성능이 떨어지면 체계전체의 성능도 떨어지게 된다. 그러므로 WebGIS개발에서 봉사기의 성능을 높이는것이 중요한 문제로 제기된다.

지금까지 대부분의 봉사기들이 C++ 또는 Java언어에 기초하여 개발되었고 WebGIS봉사기구축에서도 이러한 프로그램작성언어들을 리용하였다. 최근에 끊임없이 출현하는 최신 기술들에 의거하여 급변하는 Web환경에 신속히 적응되고 종전보다 향상된 성능과 기술을 가진 WebGIS봉사기를 설계하고 실현하는 문제도 제기된다.

론문에서는 봉사기개발에서 JavaScript를 리용할 때 사건기반의 비동기식입출력문제처리가 가능한 프레임워크기술인 Node.js를 활용하여 WebGIS봉사기를 설계하고 실현하기 위한 문제에 대하여 서술하였다.

1. Node.js의 기술적특징

Node.js의 가장 중요한 특징은 사건기반비동기입출력방식을 제공한다는것이다.

이전의 봉사기들은 동시에 다량적으로 발생하는 요청들을 처리하기 위하여 스레드를 생성하였다. 봉사기의 입출력명령처리에서 스레드처리는 Web응용프로그램을 관리하는 가장 좋은 방법으로 간주되어왔다. 특히 동기방식의 스레드를 동시에 다중으로 생성하여 명령을 처리하는 다중스레드를 주로 리용하고있다. 그렇지만 스레드를 개발하는것은 품이 많이 들고 쉽지 않다.[2] 그리고 봉사기들에서 요청이 계속 증가할수록 새로운 스레드를 추가하기 위하여 지속적으로 기억기를 할당해야 하므로 기억기소비량이 늘어나고 봉사기의 성능이 저하되는 결함이 생겨나게 된다. 이에 비해볼 때 사건기반의 비동기방식은 망입출력사건을 callback함수를 통하여 비동기입출력방식으로 처리하기때문에 요청이 증가되는 경우에도 봉사기의 성능저하를 방지할수 있다.

Node.js의 특징은 또한 다양한 모듈들과 객체들을 제공한다는것이다.

Node.js는 기본적으로 내장모듈들을 제공하며 외부모듈들을 설치하여 리용하는것도 가능하게 한다. 다양한 모듈들을 제공한다는것은 프로그램개발에서 효과성과 확장성을 제공한다는것이며 이것은 Node.js가 확장성있는 망환경을 편리하게 구축할수 있는 매우 유용한 개발환경을 제공한다는것을 말한다.

이전의 WebGIS봉사기구축에 비해볼 때 Node.js의 우점은 다음과 같다.

① Node.js는 비동기입출력방식으로 동작한다.

Node.js에서는 요청과 응답사이에 대기시간이 존재하지 않는다. 요청이 발생하면 Node.js는 응답이 도착할 때까지 대기하지 않고 다른 요청들을 계속 실행하는데 해당 요청에 대한 응답이 도착하면 곧바로 사건을 발생시켜 정보를 처리하기때문에 처리속도와 기억기호를 보장하는데서 매우 유리하다.

② Java Script를 리용한 개발이 가능하다.

Java Script는 주로 의뢰기에서 개발언어로 리용되어왔다. Node.js를 리용한 봉사기개발에 Java Script를 리용하면 개발자들이 의뢰기와 봉사기의 두 측면을 프로그램작성언어의 제약없이 익숙한 언어와 기술로 편리하게 조종할수 있다. 이것은 WebGIS봉사기개발실천에서 중요한 의의를 가진다.

2. 봉사기의 설계와 구성

우리는 Node봉사기를 GIS자료처리, 화상처리, 의뢰기요청에 대한 응답처리의 세 부분으로 나누어 설계하고 구성하였다. 사용자에게 GIS자료를 보여줄 때 Node봉사기는 DBMS에 저장된 공간자료를 추출한 후 모듈을 통하여 화상처리를 진행한다. 또한 의뢰기의 요청에 응답하여 최종봉사를 제공해주는 단계로 봉사기를 구성하였다.

① GIS자료처리

먼저 GIS자료를 DBMS에 저장하기 위한 자료변환모듈을 구현하여 다양한 DBMS를 지원할수 있도록 설계하였다. 이를 위하여 Open Source기반의 DBMS인 Postgres의 PostGIS를 선택하였다. 그리고 Node.js의 pg모듈을 설치하고 Node봉사기에 삽입하여 리용하였다. 한편 Node봉사기는 pg모듈을 통하여 Node.js와 PostGIS사이의 Query를 받아서 자료를 가져올수 있도록 하였다. 이때 사용된 명령은 다음과 같다.

```
> npm install pg
```

모듈을 설치한 다음에는 필요한 자료를 입력하기 위하여 shape자료입력모듈을 개발하였으며 자료층에 자료를 입력하고 GeoJSON형태로 자료층에서 자료를 봉사받도록 하였다.

② 화상처리

GIS봉사기를 구축하기 위해서는 호출된 GIS자료를 시각화해야 한다. 이를 위하여 Node-Canvas모듈을 사용하였다. Node-Canvas모듈은 HTML 5의 canvas를 리용하며 Java Script로 작성된 명령을 인식하여 화상객체를 시각화하는 역할을 수행한다.

③ Web응답처리

Npm레지스트리에 등록된 모듈들가운데서 널리 알려진 Web봉사모듈 expressJS를 리용하였다. expressJS는 Web요청을 받고 응답하는 역할을 수행한다.

④ 봉사기구성

의뢰기는 지도경계를 찾아서 Node봉사기에 넘겨주며 Node봉사기는 이 요청을 다음과 같은 순서로 처리한다.

지도경계에 의하여 계산된 값에 따라 각 행렬의 원소들을 추출한다.

요청에 해당되는 공간질문을 DBMS(PostGIS)에 요청하고 질문에 대한 결과값은 GeoJSON 형태의 자료로 받는다.

GeoJSON형태로 받은 자료를 각 행 또는 feature별로 분석하고 추출된 값과 자리표값을 리용하여 시각화한다.

시각화가 끝나면 Node-Canvas의 canvas객체에서 PNG stream을 추출한다.

열람기를 통하여 GIS자료가 시각화된것을 확인한다.

우리는 봉사기의 성능검사를 위하여 Java로 실현한 봉사기와 Node봉사기의 지도자료 요청에 대한 응답시간을 비교하였다. 이때 20, 40, 60대의 의뢰기에서 동시에 동일한 공간 자료조건에 대한 WMS GetMap요청을 보내고 응답시간을 측정하는 방법으로 성능검사를 진행하였는데 그 결과는 표와 같다.

표. 봉사기의 평균응답시간

표에서 보는바와 같이 Node.js로 구축된 봉사기에서 동시에 발생하는 요청의 개수 즉 의뢰기의 대수가 증가될수록 요청-응답시간은 Java로 실현한 봉사기보다 줄어든다는것을 알수 있다.

봉사기형태	평균응답시간/s		
	20대	40대	60대
Node봉사기	0.194	0.285	0.369
Java봉사기	0.359	0.403	0.890

맺 는 말

Node.js기술을 리용한 Web봉사기는 Node.js의 기술적특성을 살려 비동기식으로 입출력을 처리하므로 대규모요청처리에 적합하다.

참 고 문 헌

- [1] U. Paudyal; Scalable Web Application using Node.js and CouchDB, Uppsala University, 21~154, 2011.
- [2] I. S. Yoon; Node.js Programming for Modern Web, Hanbit Media, 14~150, 2012.
- [3] D. M. Clements; Node Cookbook, Packt Publishing, 7~121, 2014.

주체106(2017)년 5월 5일 원고접수

Design and Realization of WebGIS Server using Node.js

Kim Yu Song, Kang Yong Ho

We analyzed the technical character of Node.js.

We designed and implemented WebGIS Server by Node.js Framework technology that is possibility of asynchronous I/O processing.

Through the quality test of server, we confirmed that it reduced far the time of request-answer by applying Node.js in developing of WebGIS server.

Key words: WebGIS Server, Node.js, I/Omodel