

# Node.js 기반의 고가용성 웹 어플리케이션 서버 시스템 구조 설계 및 구현

정주영<sup>(\*)</sup>, 김민규<sup>(\*)</sup>, 류병제<sup>(\*)</sup>, 이익행<sup>(\*)</sup>, 지호준<sup>(\*\*)</sup>, 정중화<sup>(\*)</sup>, 고석주<sup>(\*)</sup>,

(\*) 경북대학교 컴퓨터학부, op@jjo.kr, snow\_chris@naver.com, fbqudwp8@naver.com, ddrgd@naver.com,  
sjkoh@knu.ac.kr, godopu16@gmail.com

(\*\*) (주) 범일정보, hojun.ji@bumil.co.kr

## Design and Implementation of High Availability Web Application Server System Architecture Based on Node.js

Jeong Ju-Yeong<sup>(\*)</sup>, Kim Min-Kyu<sup>(\*)</sup>, Ryu Byeong-Je<sup>(\*)</sup>, Lee Ic-Haeng<sup>(\*)</sup>,

Ji Ho-Jun<sup>(\*\*)</sup>, Jung Jung-Hwa<sup>(\*)</sup>, Koh Seok-Joo<sup>(\*)</sup>

(\*) *Kyungpook University, School of Computer Science and Engineering*

(\*\*) *Bumil Information(Ltd)*

### 요약

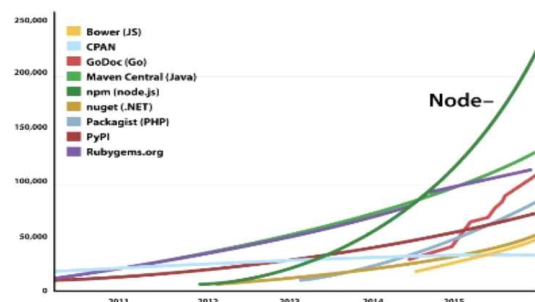
웹을 기반으로 하는 서비스 및 플랫폼을 개발하기 위한 다양한 언어와 도구가 존재하지만 그 중 Frontend의 개발에 주로 사용되었던 자바스크립트를 Backend에 사용하기 위해 개발된 Node.js는 최근 매우 각광받는 언어가 되었으며, 타 언어에 비해 높은 점유율과 성장률을 보이고 있다. Node.js는 단일 스레드 사용 및 비동기 방식을 기본 작업 처리방식으로 가진다. 여기에는 여러 장점들이 존재하지만 예외나 에러에 대한 프로세스 종료로 인해 어플리케이션 전체에 영향을 끼칠 수 있다는 문제점을 가지고 있다. 본 논문에서는 이러한 문제를 해결하기 위해, WAS Master를 사용하는 새로운 서버 시스템 구조를 제안하며 이 구조의 안정성을 검증하기 위해 테스트베드를 구축하여 WAS가 불가피하게 종료되는 상황에서 지속적인 서비스를 제공할 수 있음을 확인하였다.

## 1. 서론

Node라고도 불리는 Node.js는 최근 매우 각광받는 언어이다[1]. (그림 1) 에서 확인할 수 있듯이 타 언어에 비해 높은 점유율과 최근에도 높은 성장률을 가지고 있다. Node.js는 이벤트 기반, 논 블로킹 I/O 모델, 단일 스레드 사용 등의 여러 특징들을 가진다. 여기에는 다양한 장점이 있지만 단일 스레드 사용으로 인해 어느 한 곳에 예외 상황 및 에러가 발생하면 프로세스 자체가 종료되어 어플리케이션 전체에 영향을 끼칠 수 있으며, 하나의 작업이 지연된다면 전체 성능이 저하되는 단점이 존재한다[2]. 특히 프로세스 자체가 예기치 못하게 종료되는 상황이 발생하면 서비스를 제공하지 못하는 상황이 발생하는데, 이는 큰 손해로 이어질 수

있다.

(그림 1) 웹 언어 점유율



(Figure 1) The market share of web language

## 2. 시스템 구조

우리는 프로세스의 예외 및 에러로 인한 예기치 못한 종료로 발생하는 문제를 효율적으로 처리할 수 있는 서버 시스템 구조를 제안한다. (그림 2) 는 본 논문에서 제안한 시스템 구조를 나타낸다. 각각의 Web Server는 클라이언트들의 요청을 독립적으로 처리한다. 이때 Web Application을 필요로 하는 요청을 수신한다면 Web Server는 WAS Master에 해당 요청에 대한 처리를 요구하고 WAS Master는 적절한 WAS에게 작업을 전달한다. WAS의 작업이 완료되면 Web Server에 완료된 작업의 결과를 전달함으로써 WAS Master의 임무가 완료된다.

이러한 구조에서 Web Server들은 WAS Master를 1개의 WAS로 인식하며, WAS Master에 연결된 WAS들을 서로 공유한다. Web Server들은 WAS를 공유하는 형태이므로 한 Web Server에 WAS를 필요로 하는 요청이 몰릴 경우에도 부하 균형 효과를 가질 수 있다. 또한 Web Server는 1개의 WAS에 문제가 생겨도 클라이언트의 요청에 응답을 할 수 있기 때문에 예외적인 상황으로 인해 한 개의 WAS가 종료되는 상황에도 효율적으로 대처할 수 있다.

(그림 2) 시스템 제안 구조

(Figure 2) System Suggested Architecture

## 3. 구현 및 안정성 분석

우리는 먼저 주기적으로 WAS를 필요로 하는 요청을 시스템에 전달하는 상황에서, 고의적으로 시스템에 에러를 발생시켜 한 개의 WAS 프로세스를 종료시켰을 때 시스템이 요청을 처리하는 과정을 확인한다.

(그림 3) 안정성 테스트 결과



(Figure 3) Stability test result

(그림 3) 은 각 WAS에 대해 요청 수(REQ), CPU 사용량, RAM 사용량을 차례대로 보여준다. 그래프의 x축은 현재 시간으로부터 상대적인 시간을 뜻한다. 그래프에서 y축 방향의 일자로 뺀 직선은 프로세스가 에러로 인해 종료된 시점을 뜻한다. 그래프의 시간 -20 일 때를 보면, 하나의 WAS가 종료되어도 다른 WAS는 정상적으로 작업을 수행해 전체적인 시스템은 안정적인 서비스 제공이 가능함을 확인할 수 있다.

## 4. 결론

본 논문에서는 Node.js 기반의 고가용성 웹 어플리케이션 서버 시스템 구조를 제안하였다. 또한 테스트베드 구현 및 분석을 통해 CPU, RAM, 요청 수를 분석함으로써 이 구조의 안전성을 살펴보았다. 제안된 구조는 단일 스레드 기반 작업 처리 방식으로 인해 발생하는 문제에 효율적으로 대처할 수 있으며, 이와 함께 시스템의 안전성을 증가시키며, 부하 균형 효과를 얻을 수 있음을 확인하였다.

## 참고 문헌

- [1] <https://blog.risingstack.com/what-is-node-js-used-for-2017-survey/>
- [2] "Node.js", <http://nodejs.org>