

다양한 맵 플랫폼을 이용한 대용량 동적정보와 공간정보의 매쉬업 성능 비교 연구

A Comparative Study on Mashup Performance of Large Amounts of Spatial Data and Real-time Data using Various Map Platforms

강진원* · 김민수**
Kang, Jin-Won · Kim, Min-Soo

Abstract

Recently, the use of mashup that integrates real-time data with spatial data such as tiled map and satellite imagery has been increased significantly. As the use of mashup has been extended to various fields of O2O, LBS, Smart City, and Autonomous Driving, the performance of mashup has become more important. Therefore, this study aims to compare and analyze the performance of various map platforms, when large amounts of real-time data are integrated with spatial data. Specifically, we compare the performance of most popular map platforms available in Korea, such as Google Maps, OpenStreetMap, Daum Map, Naver Map, olleh Map, and VWorld. We also compare the performance using most common web browsers of Chrome, Firefox and Internet Explorer. In the performance analysis, we measured and compared the initialization time of basic map and the mashup time of real-time data for the above map platforms. From analysis results, we could find that Google Maps, OpenStreetMap, VWorld, and olleh Map platforms showed a better performance than the others.

Keywords: Map Platform, Open API, Mashup, Performance Analysis, Real-time data

1. 서론

최근 2차원 및 3차원 공간정보, 위성영상, 항공사진, 도로사진, 실내공간정보 등과 같은 다양한 형태의 공간정보 서비스가 일상생활에 미치는 영향은 날이 갈

수록 커지고 있다. 특히, 2005년 Google Map/Earth의 공간정보 서비스 이후, 웹 기반 공간정보 활용이 급격히 증가하고 있다. 공간정보의 활용방식에 있어서도 웹상에서 사용자가 원하는 공간정보를 단순히 검색·조회·표출하는 방식에서 최근에는 사용자가 보유한

* 대전대학교 컴퓨터공학과 석사과정 Department of Computer Engineering, Daejeon University (First author: kjwonmail@gmail.com)

** 대전대학교 컴퓨터공학과 부교수 Department of Computer Engineering, Daejeon University (Corresponding author: minsoo@dju.ac.kr)

다양한 속성 및 동적정보들과 공간정보를 융합(매쉬업)하는 방식으로 변모하고 있다 (이인수 외 2014). 실제로 이러한 공간정보 매쉬업 서비스는 다양한 분야에서 활용되고 있다. 사용자가 보유한 버스정류장, 음식점, 관광지, 주유소, 학교, 병원 등과 같은 정적 POI 정보들과의 매쉬업 서비스뿐만 아니라, 실시간 변화하는 동적정보들과의 매쉬업 서비스 수요도 발생하고 있다. 예를 들어, 시간에 따라 동적으로 변화하는 교통정보, 날씨정보, 버스노선정보, 대기오염정보, 시설물 센서정보, 재난재해 정보 등이 맵 상에 효율적으로 융합되어 표출되고 있다. 특히, 이러한 동적정보 매쉬업은 O2O, LBS, Smart City, 자율주행 등과 같은 최신의 ICT 서비스 분야에서 활발히 이용되고 있으며, 이에 따라 맵 플랫폼이 제공하는 매쉬업의 성능도 크게 중요해지고 있다.

이에 본 연구에서는 대용량 동적정보의 매쉬업 수행 시 현재 활용 가능한 맵 플랫폼들의 매쉬업 성능을 측정하여 비교해보고자 한다. 다시 말하여 어떠한 맵 플랫폼이 어느 정도 수준의 대용량 동적 데이터 매쉬업을 효율적으로 지원할 수 있는지를 파악하여 보고자 한다. 이를 통하여 사용자가 대용량 데이터의 매쉬업을 수행하고자 할 때, 가능한 데이터 용량을 파악하고 맵 플랫폼을 결정하는데 도움을 주고자 한다. 구체적으로 맵 플랫폼에 대한 성능 분석에서는 플랫폼의 서버 성능과 웹 브라우저(클라이언트)의 Open API 성능을 측정하여 분석하였다. 서버 성능 분석에서는 사용자가 Open API를 통하여 요청한 전체 배경지도의 타일을 로드하는데 걸리는 초기화 시간을 측정하였으며, 클라이언트의 매쉬업 성능 분석에서는 Open API를 통하여 웹 브라우저에서 수행되는 배경지도와 동적정보의 매쉬업 소요시간을 측정하였다. 아울러, 다양한 맵 플랫폼들의 성능을 비교 분석하기 위하여 현재 흔히 이용되고 있는 Google Maps, Daum Map, OpenStreetMap, Naver Map, olleh Map, VWorld의 맵 플랫폼을 실험에 이용하였다. 또한, 다양한 웹 브

라우저 환경에서의 성능을 분석하기 위하여 Chrome, Firefox, Internet Explorer 11(IE)의 브라우저를 이용하였다. 성능분석 결과를 요약하면 Google Maps 플랫폼이 가장 우수한 성능을 보여주었으며 OpenLayers의 오픈소스를 기반으로 Open API가 구현된 OpenStreetMap, VWorld, olleh Map 플랫폼도 비교적 우수한 성능을 보여 주었다. 이에 비하여 Daum Map과 Naver Map은 매쉬업 객체 수가 증가함에 따라 성능이 급격하게 저하됨을 알 수 있었다. 또한, 매쉬업을 이용하는 웹 브라우저에서는 Chrome 또는 Firefox가 IE에 비하여 우수한 성능을 보여줌을 알 수 있었다.

본 논문의 구성은 다음과 같다. 2장에서는 맵 플랫폼이 보유한 Open API의 기능, 성능, 활용성과 관련된 최신 연구동향에 대하여 제시하고자 한다. 3장에서는 Open API의 성능을 측정하기 위하여 본 연구에서 이용되는 성능지표를 제안하고 성능 측정을 위해 구현되는 시스템 아키텍처를 제안한다. 4장에서는 다양한 맵 플랫폼과 웹 브라우저에 대하여 수행된 매쉬업 성능분석 결과를 제시하고, 5장에서는 본 연구의 결론과 향후 연구방향에 대하여 제시하고자 한다.

2. 관련 연구

현재 Google Maps, Here Maps, Bing Maps, Daum Map, OpenStreetMap, Naver Map, olleh Map, VWorld 등과 같이 국내외의 다양한 맵 플랫폼들이 널리 이용되고 있다. 이러한 맵 플랫폼들은 공간정보 매쉬업을 통하여 지도검색, 길찾기, 부동산, 대중교통 검색, 교통정보 검색, 내비게이션 등과 같은 사용자의 수준 높은 요구사항을 만족시키는 서비스를 제공하고 있다.

최근에는 맵 플랫폼 Open API를 이용하여 사용자가 보유한 다양한 정보와 공간정보를 융합하여 활용하는 매쉬업(Mashup) 서비스가 매우 활발히 이용되고 있다 (이인수 외 2014). 또한, 이러한 매쉬업 서비스의 활성화와 더불어 Open API에 대한 기능·성능 개

선을 위한 연구개발과 Open API 기반 솔루션 개발 등에 대한 관심이 크게 증가하고 있다. (김문기 외 2015; 김병선 외 2014; 최원근 외 2014)는 Open API의 기능 및 성능이 공간정보 품질과 더불어 사용자가 맵 플랫폼을 선택하는데 있어서 중요한 기준임을 제시하고 있다.

먼저 맵 플랫폼의 Open API 기반의 신규 솔루션 개발과 관련하여 공간·위치정보, 센서정보, 소셜미디어 정보, 콘텐츠 등을 융합하여 LBS, O2O, Smart City, 자율주행, VR/AR 등의 분야에 적용하기 위한 다양한 연구개발이 진행되어 왔다. 구체적으로 (Lunsford 2016)는 Geo-Fence 영역을 설정하고 사용자의 위치 정보 기반 보안 솔루션을 제안하였으며, (Balandina et al 2015)는 문화·관광·재난재해 등의 특수 목적 서비스를 위한 LBS 플랫폼 솔루션을 제안하고 있다. (Lv et al 2015; Lunsford 2016; 정용희 외 2012)는 무선 인터넷, 실시간 측위, 스마트폰, 공간정보 기술 등을 융합하여 포켓몬고와 같은 AR 솔루션을 제안하고 있으며, (Andreini 2011; Solanki 2017; 김진희 외 2010)는 센서정보와 공간정보를 융합하여 대기오염, 시설물, 수자원, 교통, 재난재해 등에 대하여 공공 모니터링 서비스를 수행하는 Smart City 솔루션을 제안하고 있다. (문은배 외 2016)는 자율주행 차량의 경로 계산을 위하여 맵 플랫폼과 사용자의 경로정보 및 무인 차량의 GPS 정보를 연동하는 솔루션을 제안하고 있다.

맵 플랫폼이 제공하는 Open API의 기능 개선과 관련해서도 다양한 연구개발이 진행되어 왔다. (김민수·장인성 2015; Auer et al 2012; Christen et al 2012)는 Active-X 방식의 맵 플랫폼을 HTML5/WebGL의 웹 표준 방식으로 개선하는 방안을 제안하고 있으며, (Ribeiro et al 2015; Gray 2015)는 맵 플랫폼에 동적 센서정보를 효율적으로 융합할 수 있는 새로운 기능의 추가를 제안하고 있으며, (강보람 외 2010)는 맵 플랫폼에 대한 사용자 경험과 사용성 평가를 통한 인터페이스의 기능 개선 방법을 제안하고 있다.

끝으로, 맵 플랫폼 Open API의 사용자가 증가하고

매쉬업되는 사용자의 데이터가 급증함에 따라, 최근에는 맵 플랫폼의 성능 개선과 관련된 연구개발이 활발히 진행되고 있다. (장한솔 외 2015a; 장한솔 외 2015b)는 맵 플랫폼의 Web Server, WAS, DB 등의 서버 자원을 재조정하여 서버 성능을 최적화시키기 위한 방안을 제안하기 위하여 웹 로그를 이용한 2D/3D/Portal 서비스별 시스템 사용현황 분석, 시뮬레이션을 통한 CPU, Memory의 하드웨어에 대한 성능 분석, 웹 서버 응답 시간 분석 등을 수행하였다. (고준희 외 2015)는 대용량 맵 플랫폼 서비스에 적합한 최적의 아키텍처 설계 방안을 제안하기 위하여 CDN, 가상화, 클러스터링, 스토리지와 같은 다양한 신기술들의 특징 및 성능을 분석하였다. (김정민·정욱락 2016)은 효율적인 대용량 벡터 공간정보 웹 서비스 방안을 마련하기 위하여 새로운 벡터 타일 서비스 방법을 제안하고 OGC WFS와 성능을 비교 분석하였다.

이와 같이, 맵 플랫폼과 관련하여 다양한 솔루션 개발, 새로운 기능 추가, 아키텍처 성능 개선 등과 관련하여 다양한 연구가 수행되어 왔다. 그러나 기존 연구로는 최근의 매쉬업 데이터가 급증하고 선택 가능한 맵 플랫폼이 늘어난 상황을 고려할 때, 어떠한 맵 플랫폼을 선택하는 것이 효율적인지 판단하기에는 어려움이 여전히 존재한다. 다시 말하면, 맵 플랫폼을 선택하는데 있어서의 어떠한 기준도 없는 상황이다. 이에 본 연구에서는 맵 플랫폼 선택을 위한 성능적인 기준을 제시하기 위하여 다양한 맵 플랫폼 Open API에 대하여 대용량 동적정보의 매쉬업 성능을 비교하고 분석해보고자 한다.

3. 대용량 동적정보의 매쉬업 성능지표 및 성능분석 시스템 구성

본 장에서는 다양한 맵 플랫폼들의 매쉬업 성능을 분석하기 위하여 본 연구에서 이용되는 성능지표를 제안하고, 제안된 성능지표를 이용하여 성능을 분석

하기 위하여 개발되는 매쉬업 시스템의 아키텍처를 제안한다.

3.1. 대용량 동적정보의 매쉬업 성능지표

본 연구에서는 우선 사용자의 Open API 요청에 대한 맵 플랫폼 서버의 성능을 동일한 기준으로 비교하기 위하여 배경지도 초기화 시간을 성능지표로 설정하였다. 여기서 배경지도 초기화 시간은 사용자가 Open API를 통하여 요청한 영역에 해당되는 전체 배경지도 타일을 서버로부터 클라이언트에 로드하는데 소요되는 시간으로 정의한다. 이러한 배경지도 초기화는 배경지도와 사용자 정보와의 매쉬업을 수행하기 전에 반드시 수행되어야 하는 작업으로 맵 플랫폼 서버의 성능을 비교하기 위하여 배경지도 초기화 시간을 성능지표로 설정하였다. 배경지도 초기화 시간 측정에서는 클라이언트의 다양한 사용 환경을 고려하여 웹 브라우저 캐시를 적용한 경우와 적용하지 않은 경우를 구분하여 시간을 측정하였다.

둘째 사용자의 Open API 호출에 의하여 클라이언트의 웹 브라우저에서 수행되는 배경지도와 사용자 정보의 매쉬업 시간을 성능지표로 설정하였다. 본 연구에서는 동일한 기준의 성능지표를 설정하기 위하여 매쉬업 시간을 배경지도와 사용자 정보가 클라이언트에 로드된 이후의 사용자 정보에 대한 벡터 레이어 생성 시간과 생성된 벡터 레이어와 배경지도와의 매쉬업 맵 생성 시간의 합으로 정의하고 있다. 다시 말하면, 매쉬업 시간은 사용자 정보의 로드 시간과 매쉬업 맵의 화면 렌더링 시간 등을 제외하고 순수하게 매쉬업에 소요된 시간만을 포함한다. 또한, 본 연구에서는 사용자 정보에 대하여 벡터 레이어 생성을 통하여 매쉬업 시간을 측정하였다. 이는 벡터 레이어가 사용자에 의하여 가장 흔하게 이용되고 있으며, 매쉬업을 통한 벡터 레이어 생성은 상당히 많은 시간을 소모하기 때문에 맵 플랫폼들 간의 성능차이가 크게 나타날 수

있기 때문이다.

끝으로, Table 1은 본 절에서 주어진 성능지표를 이용하여 분석하고자 하는 맵 플랫폼들과 Open API 버전을 보여준다.

Table 1. Spatial open platforms and open API versions used for performance analysis

Open Platform (Vendor)	Open API Version
Google Maps (Google)	3.30
VWorld (SpaceN)	2.0 (OpenLayers 3)
OpenStreetMap (OSM)	3.0 (OpenLayers 3)
Daum Map (Kakao)	3.0
Naver Map (Naver)	3.1.0
olleh Map (KT)	3.0

3.2. 성능분석을 위한 매쉬업 시스템 구성

본 연구에서는 맵 플랫폼들의 매쉬업 성능을 측정하기 위하여 배경지도 로드, 동적정보 로드, 벡터 레이어 생성, 배경지도와 벡터 레이어 매쉬업 등의 기능을 수행할 수 있는 간단한 시스템을 구현하였다 (Figure 1).

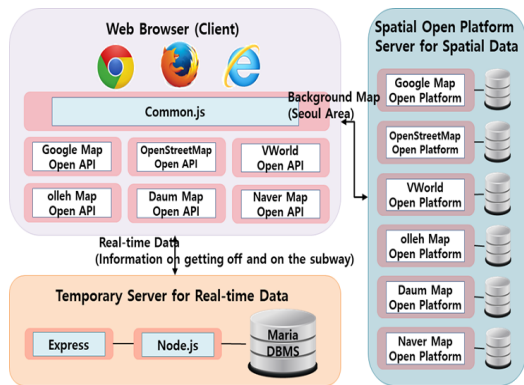


Figure 1. Mashup system architecture for performance analysis

Figure 1에서 보듯이, 성능분석을 위한 전체 매쉬업 시스템은 웹 브라우저 클라이언트, 맵 플랫폼 서버, 실시간 동적정보 제공을 위한 임시 서버로 구성되어 있다.

임시 서버는 Node.js의 Express 프레임워크 기반 WAS(Web Application Server)와 MariaDB의 DBMS로 구성되어 있다. 여기서, MariaDB는 클라이언트에게 제공하기 위한 동적정보로서 공공데이터 센터에서 수집된 약 17만 건의 수도권 지하철 승하차 정보를 저장관리하고 있으며, WAS는 클라이언트 요청에 대응하여 지하철 승하차 정보를 제공할 수 있도록 구현되어 있다.

맵 플랫폼 서버는 Open API 서비스를 제공하고 있으며, 서울 지역의 배경지도를 제공할 수 있도록 연결되어 있다.

클라이언트는 임시 서버로부터 동적정보를 로드하고 맵 플랫폼의 Open API를 활용하여 배경지도 로드, 벡터 레이어 생성, 매쉬업 등을 수행할 수 있도록 구현되어 있다. 특히, 클라이언트는 각 맵 플랫폼들의 자체 Open API가 호출되는 부분 이외의 임시 서버 연계 모듈, 기타 기능 모듈 등을 플랫폼들 간에 불필요한 성능 차이가 발생하지 않도록 공통 모듈로 구현하고 있다. 다만, 앞 절에서 설명한 순수한 매쉬업 시간만을 정확히 측정하기 위하여 클라이언트는 일부 맵 플랫폼들의 Open API 코드를 분석하여 내부에 사용자 로그 정보를 추가함으로써 실제로 매쉬업 작업에만 소요된 시간을 측정할 수 있도록 API 일부를 수정하였다. 구체적으로 Table 2와 같이 배경지도 로드 완료 이벤트(tilesload event)와 매쉬업 완료 이벤트(mapshupend event)를 발생시키지 않는 맵 플랫폼들에 대하여 웹 브라우저에서 제공하는 MutationObserver 기능을 이용하여 DOM 트리의 변경사항을 실시간 감시하여 배경지도 완료와 매쉬업 완료 시점을 찾아 사용자 로그 정보를 추가하도록 시스템을 구현하고 있다.

예를 들어, Table 2에서 Google Maps, Daum Map, Naver Map은 매쉬업 완료 이벤트를, olleh Map은 배경지도 로드 완료 및 매쉬업 완료 이벤트를 발생시키지 않고 있다.

Table 2. Event handling support of spatial open platforms

Open Platform	tilesload Event	mashupend Event
Google Maps	Supported	Not Supported
VWorld	Partially	Partially
OSM	Partially	Partially
Daum Map	Supported	Not Supported
Naver Map	Supported	Not Supported
olleh Map	Not Supported	Not Supported

끝으로, 클라이언트에서 IE 웹 브라우저는 ECMAScript 6(ES6)의 Javascript 코드를 지원하지 않기 때문에, Babel을 이용하여 ES6 버전으로 구현된 클라이언트를 ES5 버전으로 변환하여 실험을 수행하였다.

4. 배경지도 초기화 및 매쉬업 성능분석

본 장에서는 Table 3과 같은 HW, 배경지도, 실시간 동적정보에 대하여 수행된 배경지도 초기화와 매쉬업의 성능분석 결과를 보여준다.

Table 3. Experimental environment for mashup performance analysis

H/W	<ul style="list-style-type: none"> · CPU: Intel i5 3.5GHz dual core · GPU: Intel HD graphics 630 · Memory: DDR4 8GB · OS: Windows 10 pro 64bit
Basic Map	<ul style="list-style-type: none"> · Background map: Seoul area · Map size: 1024 x 768
Real-time Data	<ul style="list-style-type: none"> · Data of getting on and off the subway by time frame · Number of total dataset: 170,000 · Number of experimental data: 2,000 ~ 20,000 · Creation of a vector layer for mashup

Table 3에서 보듯이, 일반적인 컴퓨팅 환경에서 대용량 동적정보의 매쉬업 가능성을 분석하기 위하여 실험에 사용된 클라이언트로는 Windows 플랫폼의

저 사양 컴퓨터를 이용하였다. 배경지도로는 각 맵 플랫폼들이 서울 전역에 대한 타일 이미지들을 포함할 수 있도록 동일한 좌표, 동일한 줌 레벨과 1024 x 768의 맵 크기를 적용하였다. 매쉬업 되는 실시간 동적정보로는 각 시간대별로 생성된 수도권의 지하철 승하차 정보 17만 건을 이용하였으며, 실제 실험에서는 배경지도 영역에 포함된 각 역에 대하여 시간대별로 2,000 ~ 20,000 건의 데이터를 추출하여 Circle 형태의 벡터 레이어를 생성하여 배경지도와의 매쉬업에 이용하였다. Figure 2는 Google Maps와 VWorld 플랫폼에서 1024 x 768 크기의 서울 지역 배경지도와 지하철 승하차 정보의 벡터 레이어가 매쉬업된 예제를 보여준다.

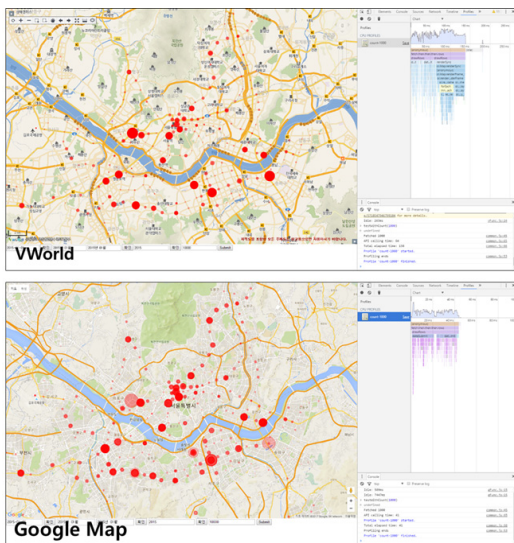


Figure 2. Mashup example of a background map and a real-time vector layer

끝으로, 본 실험에서는 정확한 성능 측정 결과를 얻기 위하여 Google Maps, VWorld, OSM, Daum Map, Naver Map, olleh Map의 맵 플랫폼에 대하여 Chrome, Firefox, IE의 웹 브라우저를 이용하여 배경지도 초기화 시간 및 동적정보 매쉬업 시간을 반복적으로 5회 측정하여 평균값을 추출하였다.

4.1. 배경지도 초기화 성능분석

본 절에서는 첫 번째 성능지표인 맵 플랫폼들의 배경지도 초기화 시간을 측정하고 비교하였다. 성능분석은 Table 1에서 언급된 전체 맵 플랫폼들에 대하여 Chrome, Firefox, IE를 이용하여 수행되었으며, 웹 브라우저 캐시를 적용한 경우와 적용하지 않은 경우에 대하여 수행되었다. Figure 3은 Google Maps, VWorld, OSM, Daum Map, Naver Map, olleh Map의 배경지도 초기화 성능을 보여준다.

먼저 브라우저 캐시를 이용하지 않은 경우, 배경지도 초기화 성능은 일반적으로 서버의 처리 용량, 서버의 위치, 그리고 접속된 사용자 수와 관련이 깊다고 할 수 있다. 예를 들어, Figure 3에서 서버가 해외에 위치한 Google Maps와 OSM은 국내의 다른 플랫폼들에 비하여 배경지도 초기화에 많은 시간을 소모하고 있음을 알 수 있다. 더구나 Google Maps와 OSM은 부가적으로 TLS(Transport Layer Security)의 암호화도 지원하고 있어서 배경지도 로드에도 더욱 많은 시간을 소모하고 있다. 실제로 OSM은 3,000ms 이상의 상당한 시간을 소모하고 있으며, Google Maps는 600ms의 시간을 소모하고 있다. 이에 비하여, 국내의 VWorld, Daum Map, Naver Map, olleh Map은 단지 227 ~ 470ms의 시간을 소모하고 있다. 여기서, Google Maps는 서버가 해외에 위치하고 TLS를 지원함에도 불구하고, 뛰어난 서버 처리 용량과 HTTP/2의 멀티플렉싱 기능을 통하여 600ms의 비교적 우수한 성능을 보여주고 있다. Daum Map과 Naver Map도 수많은 사용자에도 불구하고, 효율적인 다중 서버 활용을 가능하게 하는 DNS(Domain Name System) 등을 이용한 로드 밸런싱 기능을 통하여 우수한 성능을 보여주고 있다. 또한, 웹 브라우저별 배경지도 초기화의 시간차를 살펴보면 배경지도 초기화 시간이 상당히 많이 걸리는 OSM을 제외한 다른 맵 플랫폼들에서는 큰 시간 차이가 발생하지 않음을 알 수 있다.

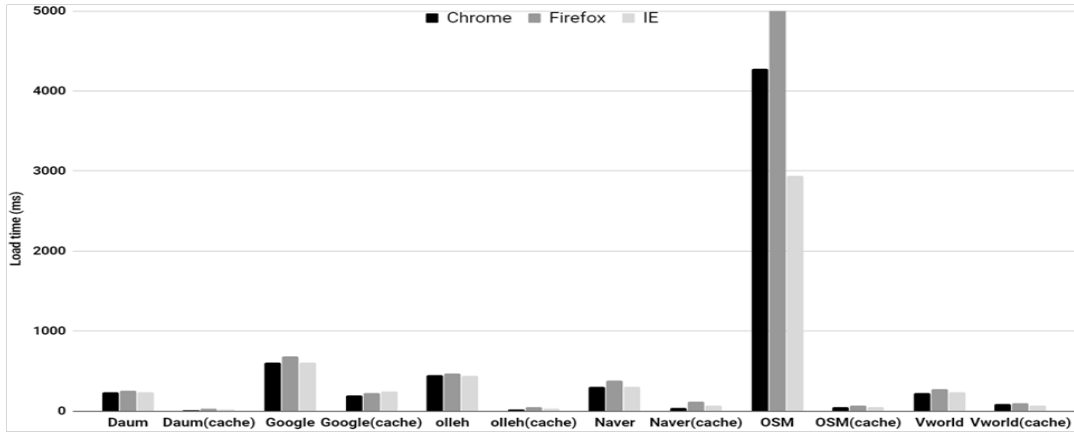


Figure 3. Load time of 1024 x 768 sized background map for six kinds of spatial open platforms, where three web browsers of Chrome, Firefox, and IE are used in this experiment

브라우저 캐시를 이용한 경우는 서버로부터 배경지도를 다운로드 받을 필요가 없기 때문에 모든 맵 플랫폼들이 200ms 가량의 빠른 시간 내에 배경지도 초기화를 완료하고 있음을 알 수 있다.

본 실험결과를 통하여 배경지도 초기화의 성능은 맵 서버의 처리 용량, 서버의 위치, 접속 사용자의 수의 기본적인 요인뿐만 아니라, TLS의 암호화 적용여부, DNS 기반 부하 분산 지원여부, HTTP/2 지원여부 등의 요인과의 밀접한 관련이 있음을 알 수 있었다. Table 4는 본 연구에서 실제로 이용된 맵 플랫폼들의 TLS, DNS 부하 분산, HTTP/2 지원여부 상황을 보여준다. Table 4에서 보듯이 향후 국내의 맵 플랫폼들은 Google과 마찬가지로 TLS의 암호화 적용을 통하여 성능이 다소 떨어질 위험성이 있으며, 성능 향상을 위하여 HTTP/2의 멀티플렉싱 기능도 도입할 필요가 있음을 의미한다.

배경지도 초기화의 결과를 요약해보면 단순 배경지도를 이용하는 경우, 현재 OSM 보다는 맵 서버가 국내에 위치한 Daum, Naver, olleh Map, VWorld와 해외 서버임에도 불구하고 우수한 성능을 보여주고 있는 Google Maps을 이용하는 것이 바람직하다고 할 수 있다.

Table 4. TLS, DNS load balancing, and HTTP/2 support of spatial open platforms

Open Platform	TLS	DNS Load Balancing	HTTP/2
Google Maps	O	X	O
VWorld	X	X	X
OSM	O	O	X
Daum Map	X	O	X
Naver Map	X	O	X
olleh Map	X	X	X

4.2. 대용량 동적정보 매쉬업 성능분석

본 절에서는 두 번째 성능지표인 배경지도와 동적정보의 매쉬업 시간을 측정하고 비교하였다. Figure 4는 Chrome, Firefox, IE 브라우저를 이용하여 수행된 Google Maps, OSM, VWorld, olleh Map, Daum Map, Naver Map의 매쉬업 성능을 보여준다.

첫째 Figure 4에서 동적정보의 수가 2,000 ~ 20,000 개로 증가할 때 매쉬업 시간의 전반적인 추세를 분석해 보면, Google Maps가 최대 500ms 이하의 매쉬업 시간으로 가장 우수한 성능을 보이고 있음을 알 수 있다. 이후, 그 뒤를 OSM(600ms 이하), VWorld(800ms

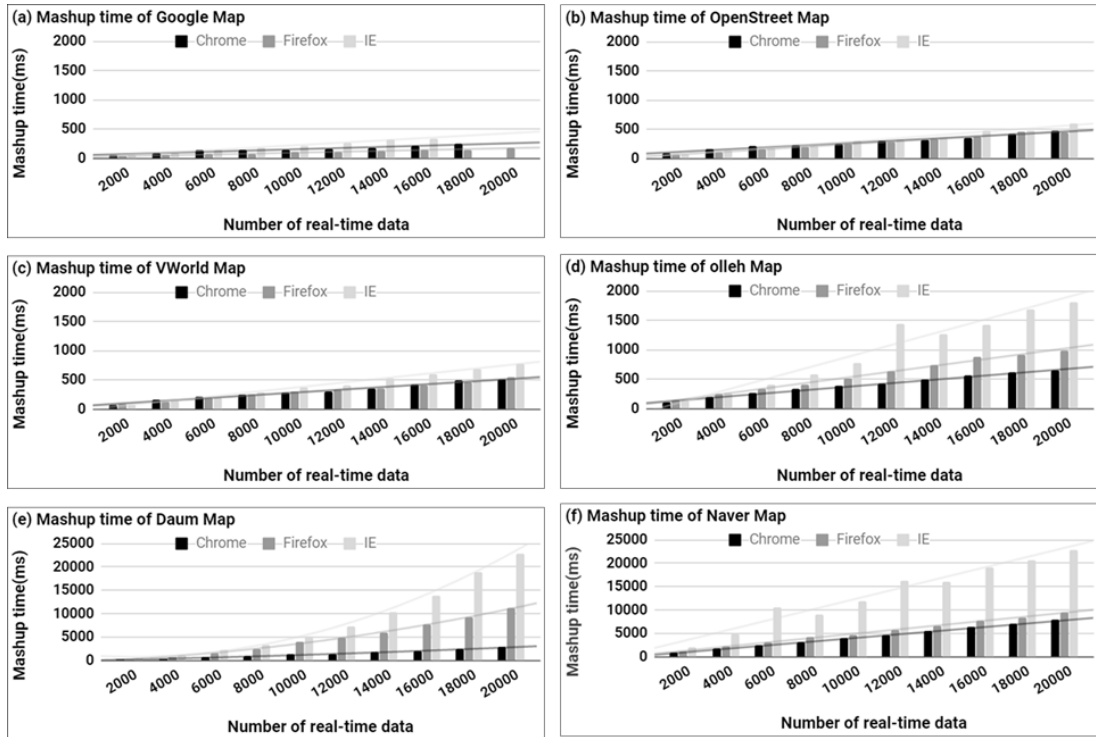


Figure 4. Mashup time of spatial data and real-time data for six kinds of spatial open platforms (a)~(f), where the number of real-time data varies from 2,000 to 20,000

이하), olleh Map(1,800ms 이하)가 따르고 있다. 이에 비하여, Daum Map과 Naver Map은 약 23,000ms 이하의 매쉬업 시간으로 성능이 크게 떨어짐을 알 수 있다. 이는 Google Maps가 병렬 및 비동기 처리 기술 기반의 가장 뛰어난 성능의 Open API를 제공하고 있으며, OSM, VWorld, olleh Map은 성능이 우수한 OpenLayers 2 또는 3 버전으로 구현된 Open API를 제공하고 있기 때문으로 추측된다. 이에 비해 Daum Map과 Naver Map은 벡터 레이어의 매쉬업 성능 향상을 위하여 Open API에 대한 개선 여지가 있을 것으로 판단된다.

둘째 Figure 4에서 Chrome, Firefox, IE의 적용된 웹 브라우저를 고려하여 각 맵 플랫폼별로 매쉬업 성능을 분석한 결과는 다음과 같다. 추세선 분석에서 가장 우수한 성능을 보여주었던 Google Maps는

Firefox, Chrome, IE 순으로 성능 차이가 있으나, 브라우저 간의 최대 매쉬업 시간차가 단지 0.18초로 사용자가 이용하는데 큰 차이가 없음을 알 수 있다. 다만, Google Maps는 동적정보의 객체수가 18,000개 이상으로 증가할 때, 브라우저 내에서 메모리 부족 등의 이유로 매쉬업이 중단되는 단점이 있었다. OSM과 VWorld에서는 Chrome과 Firefox과 비슷한 수준의 우수한 매쉬업 성능을 보여주며 IE의 성능이 다소 떨어지나, 브라우저 간의 최대 매쉬업 시간차가 OSM에서 0.16초, VWorld에서 0.22초로 큰 차이는 없음을 알 수 있다. olleh Map은 Chrome, Firefox, IE 순의 매쉬업 성능으로 Chrome과 Firefox는 최대 0.31초, Chrome과 IE는 최대 1.17초의 비교적 큰 시간차를 보이고 있으며, 특히 객체 수가 증가할수록 시간차가 더욱 커지고 있음을 볼 수 있다. Daum Map은 동적정보

의 수가 증가할 때 Chrome은 비교적 안정적인 성능 저하를 보여주고 있으나, Firefox와 IE는 성능이 급격히 저하되고 있음을 볼 수 있다. 구체적으로 동적정보의 수가 20,000개 일 때, Chrome은 2.9초, Firefox는 11.2초, IE는 22.8초의 매쉬업 시간을 소모하고 있다. Naver Map은 동적정보의 수가 증가할 때 모든 웹 브라우저들에서의 매쉬업 성능이 전반적으로 크게 저하되고 있으며, 특히 IE는 성능이 급격히 저하되고 있음을 알 수 있다. 구체적으로 동적정보의 수가 20,000개 일 때, Chrome은 7.8초, Firefox는 9.5초, IE는 22.9초의 매쉬업 시간을 소모하고 있다.

본 매쉬업 성능분석 결과를 종합해보면 다음과 같은 결론을 얻을 수 있다. 첫째, 실험에서 적용된 2,000 ~ 20,000개의 동적정보 매쉬업에 대하여 Google Maps가 가장 우수한 성능을 보였으며 OSM, VWorld, olleh Map도 사용자가 이용하는데 무리가 없는 수준의 비교적 우수한 성능을 보여 주었다. 이에 비하여, Daum Map과 Naver Map은 동적정보 객체 수가 증가함에 따라 매쉬업 성능이 크게 저하됨을 알 수 있었다. 둘째, 웹 브라우저별 매쉬업 성능은 맵 플랫폼별로 일부 차이는 존재하나, 전반적으로 동적정보의 수가 증가할수록 Chrome과 Firefox가 IE에 비하여 우수한 매쉬업 성능을 제공함을 알 수 있었다. 셋째, 매쉬업을 위한 동적정보의 객체 수가 급격히 증가하는 경우 벡터 레이어를 이용한 매쉬업 성능이 급격히 떨어지거나 매쉬업이 수행되지 않는 경우가 다수 발생함을 알 수 있었다. 실제로 실험에서 Google Maps는 Chrome에서 20,000개 이상, IE는 18,000 이상일 때 매쉬업을 수행할 수 없었다. 이에 현재 수준의 맵 플랫폼들에 대하여 대용량의 정보를 매쉬업하는 경우, 벡터 레이어 방식 이외의 히트맵과 같은 방식을 이용한 매쉬업이 바람직할 것으로 예측된다. 또한, 객체 수가 급격히 증가한 동적정보의 빠른 매쉬업을 위해서는 웹 클라이언트에서의 매쉬업보다는 맵 플랫폼이 위치한 서버의 클라우드 인프라에 동적 객체를 미리 업로드하고 서

버에서 매쉬업을 수행하는 것이 더욱 바람직할 것으로 예측된다.

5. 결론

본 연구에서는 현재 이용 중인 다양한 맵 플랫폼들과 다양한 웹 브라우저들을 이용하여 대용량 동적정보의 매쉬업 성능을 비교 분석하였다. 성능분석 과정에서 맵 플랫폼의 서버 성능을 분석하기 위하여 배경지도의 초기화 시간을 측정하였으며, Open API의 매쉬업 성능을 분석하기 위하여 웹 브라우저에서 배경지도와 동적정보의 매쉬업 시간을 측정하였다. 성능분석 결과를 통하여 배경지도 초기화 성능은 서버 처리 용량, 서버 위치, 사용자 수의 요인 이외에도 서버의 TLS, 부하 분산, HTTP/2의 적용 여부와도 밀접한 관련이 있음을 알 수 있었다. 또한, 매쉬업 성능은 동적정보의 수가 증가할 때 Google Maps, OSM, VWorld, olleh Map이 좋은 성능을 제공하였으며, Chrome과 Firefox가 IE에 비하여 전반적으로 좋은 성능을 제공하였음을 알 수 있었다.

이와 같이 본 연구에서는 배경지도 초기화와 매쉬업 시간 측정을 통하여 맵 플랫폼들의 매쉬업 성능을 부분적으로나마 분석할 수 있었다. 이를 통하여 최근 이슈가 되고 있는 동적 데이터를 다양한 맵 플랫폼과 융합하고자 할 때, 매쉬업 성능에 의하여 최소한의 기본적인 선택 기준을 제시할 수 있었다. 그러나 본 연구에서는 히트맵을 이용하는 등의 매쉬업 방법이 있어서의 다양성과 매쉬업 시의 메모리 소모량 측정 등과 같은 성능지표에 있어서의 다양성이 일부 부족하였다. 이에 향후 연구에서는 더욱 구체적이고 정확한 매쉬업 성능을 분석하기 위하여 매쉬업 이후 렌더링 소요 시간, 히트맵을 이용한 매쉬업 소요 시간, 그리고 메모리 소모량을 추가적으로 분석할 계획이다.

감사의 글

This research was supported by the Daejeon University Research Grants (2016).

참고문헌

References

- 고준희, 임용화, 김민수, 장인성. 2015. 차세대 브이월드 시스템 아키텍처 구성에 관한 연구: 최적의 아키텍처 설계를 위한 신기술 분석. 한국공간정보학회지. 23(4):13-22.
- Go JH, Lim YW, KIM MS, Jang IS. 2015. A Study on the Next VWorld System Architecture: New Technology Analysis for the Optimal Architecture Design. *Journal of Korea Spatial Information Society*. 23(4):13-22.
- 국토교통부. 2017. 브이월드 [인터넷]. [http://map.vworld.kr]. 2017년 10월 4일 검색.
- (MOLIT) Ministry of Land, Infrastructure and Transport. 2017. VWorld [Internet]. [http://map.vworld.kr] Last accessed 4 Oct 2017.
- 김문기, 윤동현, 고준환. 2015. 미래 공간정보 오픈플랫폼의 개발전략에 관한 연구. 한국공간정보학회지. 23(2):59-68.
- Kim MG, Yoon DH, Koh JH. 2015. A Study on the Development Strategy for Future GeoSpatial Open Platform. *Journal of Korea Spatial Information Society*. 23(2):59-68.
- 김민수, 장인성. 2015. HTML5/WebGL 기반 3D 공간정보 오픈플랫폼 소프트웨어 설계 및 구현. 한국공간정보학회지. 23(6):57-66.
- Kim MS, Jang IS. 2015. Design and Implementation of 3D Geospatial Open Platform Based on HTML5/WebGL Technology. *Journal of Korea Spatial Information Society*. 23(6):57-66.
- 김병선, 안종욱, 신동빈. 2014. 서비스 중심의 국가 공간정보 플랫폼 연계 방안에 관한 연구. 한국공간정보학회지. 22(2):11-18.
- Kim BS, Anh JW, Shin DB. 2014. A Study on the Construction of Service-oriented Connection Model among National GeoSpatial Information Platforms. *Journal of Korea Spatial Information Society*. 22(2):11-18.
- 김정민, 정웅락. 2016. 공간정보 오픈플랫폼 지도 서비스 성능 향상을 위한 벡터 파일 포맷 소개. 한국지형공간정보학회 학술대회. 75-78.
- Kim Jm, Jung WR. 2016. *Introduction of the Vector Tile Format of the Spatial Information Open Platform for improving map service performance*. Conference of The Korean Society for Geospatial Information Science. 75-78.
- 네이버. 2017. 네이버 지도 [인터넷]. [http://map.naver.com]. 2017년 10월 4일 검색.
- Naver. 2017. Naver Map [Internet]. [http://map.naver.com]. Last accessed 4 Oct 2017.
- 이인수, 이준석. 2014. 공간정보 포털 동향 분석에 관한 연구. 지적과국토정보. 44(2):125-138.
- Lee IS, Lee JS. 2014. A Study on Analysis on Geospatial Information Portal Trends. *Journal of Cadastre*. 44(2):125-138.
- 장한술, 고준희, 김민수, 장인성. 2015a. 공간정보 오픈플랫폼 서비스의 성능 분석 및 자원 재조정 방안에 관한 연구. 한국공간정보학회지. 23(4):1-11.
- Jang HS, GO JH, Kim MS, Jang IS. 2015a. A Study on Performance Analysis and Resource Re-distribution Method of the Spatial Information Open Platform Service. *Journal of Korea Spatial Information Society*. 23(4):1-11.
- 장한술, 홍성훈, 김민수, 장인성. 2015b. 웹 및 로그 분석 기반 공간정보 오픈플랫폼 서비스 사용 현황

- 분석. 한국공간정보학회지. 23(3):45-54.
- Jang HS, Hong SH, Kim MS, Jang IS. 2015b. Service Status Analysis About the Spatial Information Open Platform based on the Analysis of Web Server Log and System Log. *Journal of Korea Spatial Information Society*. 23(3):45-54.
- 정용희, 양평우, 이창우, 남광우. 2012. WFS와 GeoSensor 서버를 지원하는 모바일 AR 시스템. 정보과학회논문지: 컴퓨팅의 실제 및 레터. 18(12):848-858.
- Jung YH, Yang PW, Lee CW, Nam KW. 2012. Mobile AR System supporting WFS and GeoSensor Server. *Journal of KIISE*. 18(12):848-858.
- 최원근, 김민수, 장인성, 장윤섭. 2014. 공간정보 오픈 플랫폼 설계를 위한 2D Web Mapping Open API 비교 연구. 한국공간정보학회지. 22(5):87-98.
- Choi WG, Kim MS, Jang IS, Chang YS. 2014. The Comparative Research On 2D Web Mapping Open API for Designing Geo-Spatial Open Platform. *Journal of Korea Spatial Information Society*. 22(5):87-98.
- 카카오. 2017. 다음 지도 [인터넷]. [http://map.daum.net]. 2017년 10월 4일 검색.
- Kakao. 2017. Daum Map [Internet]. [http://map.daum.net]. Last accessed 4 Oct 2017.
- 케이티. 2017. 올레 맵 [인터넷]. [http://map.ollehmap.com]. 2017년 10월 4일 검색.
- KT. 2017. olleh Map [Internet]. [http://map.ollehmap.com]. Last accessed 4 Oct 2017.
- Andreini F, Crisciani F, Cicconetti C. 2011. A scalable architecture for geo-localized service access in smart cities. *Future Network & Mobile Summit*.1-8.
- Auer. M. 2012. Real-Time Web GIS Analysis Using WebGL, *International Journal of 3-D Information Modeling*. 1(3):49-61.
- Balandina E, Balandin S, Koucheryavy Y, Mourmtsev D. 2015. *Innovative e-Tourism Services on Top of Geo2Tag LBS Platform*. International Conference on Signal-Image Technology Internet-Based Systems. 752-759.
- Choi WG, Kim MS, Jang IS, Chang YS. 2014. The Comparative Research On 2D Web Mapping Open API for Designing Geo-Spatial Open Platform. *Journal of Korea Spatial Information Society*. 22(5):87-98.
- Christen. M, Nebiker. S, Loesch. B. 2012. Web-Based Large-Scale 3D-Geovisualisation Using WebGL: The OpenWebGlobe Project. *International Journal of 3D Information Modeling*. 1(3):16-25.
- Google. 2017. Google Maps [Internet]. [https://maps.google.com]. Last accessed 4 Oct 2017.
- Gray MA. 2015. Spheres: *A Web Services Framework for Smartphone Sensing as a Service*. International Conference on Next Generation Mobile Applications, Services and Technologies. 19-28.
- Lunsford P. 2016. *Geo-fencing Technologies and Security*. East Carolina University, College of Technology and Computer Science, Department of Technology Systems, William Jason Haddock. ICTN 4040.
- Lv Z, Yin T, Zhang X, Song H, Chen G. 2016. Virtual Reality Smart City Based on WebVRGIS. *IEEE Internet of things journal* 3(6):1015-1024.
- Microsoft. 2017. Bing Maps [Internet]. [http://www.bing.com/maps]. Last accessed 4 Oct 2017.
- OpenStreetMap Foundation. 2017. OpenStreetMap

- [Internet]. [<https://www.openstreetmap.org>]. Last accessed 4 Oct 2017.
- Ribeiro A, Vieira J, Sousa V, Cardoso A. 2015. *Demonstration of GIS web-based platform for experimentation supported by geosensors in a WSN*. Experiment International Conference. 137-138.
- Solanki VK, Venkaesan S, Katiyar S. 2017. Conceptual Model for Smart Cities: Irrigation and Highway Lamps using IoT. International *Journal of Interactive Multimedia and Artificial Intelligence*. 4(3): 28-33.
- VW Group, BMW, Daimler. 2016. Here Maps [Internet]. [<https://here.com>]. Last accessed 4 Oct 2017.
-
- 2017년 10월 9일 원고접수(Received)
2017년 11월 21일 1차심사(1st Reviewed)
2017년 12월 7일 2차심사(2nd Reviewed)
2017년 12월 8일 게재확정(Accepted)

초 록

최근 2/3차원 공간정보, 항공/위성영상, 실내공간정보 등의 다양한 공간정보 활용방법에 있어서 사용자가 보유한 실시간 동적정보와 공간정보를 융합하는 매쉬업 활용이 급격히 증가하고 있다. 특히, O2O, LBS, Smart City, 자율주행 등의 분야에서 공간정보와 대용량 동적정보의 매쉬업 수요가 급증함에 따라, 최근 맵 플랫폼이 제공하는 매쉬업 성능에 대한 관심도 크게 증가하고 있다. 이에 본 연구에서는 현재 국내에서 활용 가능한 다양한 맵 플랫폼들의 Open API 기반 대용량 동적정보의 매쉬업 성능을 비교 및 분석해보고자 한다. 구체적으로 Google Maps, OpenStreetMap, Daum Map, Naver Map, olleh Map, VWorld에 대하여 Chrome, Firefox, Internet Explorer의 브라우저를 이용하여 매쉬업 성능을 비교 및 분석하였다. 성능분석 과정에서 맵 플랫폼의 서버 성능을 분석하기 위하여 배경지도 초기화 시간을 측정하였으며, Open API 기반 매쉬업 성능을 분석하기 위하여 배경지도와 동적정보의 매쉬업 시간을 측정하였다. 성능분석 결과, Google Maps, OpenStreetMap, VWorld, olleh Map이 비교적 우수한 성능을 보여주었으며, Chrome과 Firefox가 IE에 비하여 좋은 성능을 제공함을 알 수 있었다.

주요어 : 맵 플랫폼, 공개 API, 매쉬업, 성능분석, 실시간 정보