

Sampeu: Servicing Web Map Tile Service over Web Map Service to Increase Computation Performance

R M Awangga

Applied Bachelor Program of Informatics Engineering, Politeknik Pos Indonesia,
Bandung 40151, Indonesia

awangga@poltekpos.ac.id

Abstract. Open Geospatial Consortium (OGC) has a standard for cartographic over web service; there are Web Map Service (WMS) used by MapServer, Web Feature Service (WFS), and Web Map Tile Service. There is research on comparing two cartography information between WMS and WFS, so this research extends to proposed solution by comparing WMS and WMTS computation cost and created a product called Sampeu. Sampeu has proposed a solution by creating WMTS protocol over WMS. WMTS work on the tile-ing system where the data transform from big picture to part of the Tile is. A lot of Tile making a pyramid of cartography and serving to client one by one, so the computation cost can be decreasing.

1. Introduction

The need to visualize maps in digital form is a necessity in the current era of Internet use where all data is distributed globally [1]. OGC has created standards for the exchange of geospatial data for use in the Internet network. Data processing speed is required to do the job. One factor of the rate of geospatial processing data is the cost of computing from the server. So this research was conducted to perform study between WMTS computing with WMS input with MapServer and MapProxy usage. So with the minimal cost of computing, the speed of data processing and geospatial analysis will be faster [2].

2. Related Works

Research on the performance of the WMS conducted in the field of mobile computer [3] as well as monitoring framework [1], and framework for integration with Discrete Global Grid Systems (DGGS) [4]. Models for integration of the Web Enablement Sensor module involve interconnection using XML [5] as well as its development such as MGeoSciML [6]. The model in Web Ontology Language (OWL) is also performed to illustrate the web service of OGC [7]. The creation of cache server side from WMS [8] as a form of computational efficiency. As a supporter of its infrastructure, the OGC Interoperability Program was created to communicate between users of the OGC standard [9]. OGC has also implemented using Service-Oriented Architecture (SOA) [10].

Research on the use of WMS regarding database performance has performed using LeafletJS for mobile [11]. There is a study to compare performance-servicing WMS with two databases between PostgreSQL and MongoDB. Simulating the same data set with a different database server to get the computational measurement. Different data sets accessed and tried in parallel with a 100-user simulation [12]. Research on the combined use of Navy Coastal Ocean Model (NCOM) with WMS has been done



to produce a more variable layer [13]. Thus, in this study, efficiency was performed with a combination of MapServer and MapProxy usage to reduce the cost of computation.

3. Proposed Framework

The model proposed in this research is the provision of WMTS taken from the WMS. MapServer is chosen as a WMS provider because it is one of the popular open source applications. Then from the WMS service MapServer will be forwarded into input from Mapproxy. Mapproxy is one application to create map cache with output option that can be different from the input. In this research, the selected MapProxy output is OGC WMTS protocol.

3.1. Directory Structure

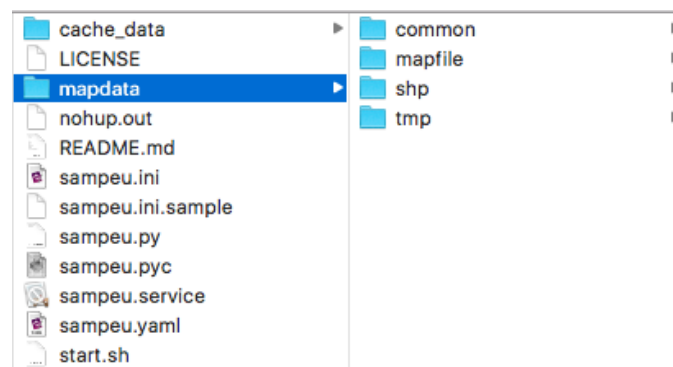


Figure 1. Directory structure of Sampeu Framework

According to Figure 1, there is the main file sampeu.py, sampeu.yaml, sampeu.ini, start.sh. Sampeu.py is a python script to sync MapProxy module and function to call WSGI module from MapProxy by taking configure from sampeu.yaml. Sampeu.yaml is a file that contains OGC format configuration as input data and output data. Sampeu.ini is a server configuration based on uWSGI. Start.sh is a script to run sampeu as a server serving web service WMTS OGC. In the folder, there is a common folder, mapfile, shp, and tmp.

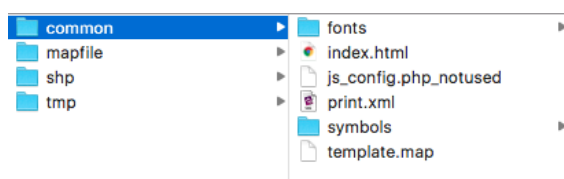


Figure 2. Common directory structure

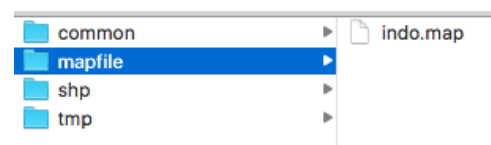


Figure 3. Mapfile directory structure

In the common folders (Figure 2) contain map templates such as fonts, javascript, HTML. The mapfile folder (Figure 3) contains a .map file which is a standard file for querying MapServer as WMS-based input. The shp folder (Figure 4) contains a vapor shapefile file consisting of shp, dbf, shx, prj, sbn, shx, cpq, xml files. While the tmp folder (Figure 5) as a temporary directory location that can be used either by MapServer or MapProxy.



Figure 4. SHP directory structure



Figure 5. tmp directory structure

3.2. The Script

The program code used in the Sampeu framework consists of YAML code, python code, this code, bash code and Mapfile format. Starting from a YAML configuration containing WMS input and WMTS output. The structure of the YAML configuration is as follows.

```

services:
  demo:
    tms:
      use_grid_names: true
      # origin for /tiles service
      origin: 'nw'
    kml:
      use_grid_names: true
wmts:
  # use restful access to WMTS
  restful: true
  # this is the default template for MapProxy
  restful_template:
    '/{Layer}/{TileMatrixSet}/{TileMatrix}/{TileCol}/{TileRow}.{Format}'
  # and also allow KVP requests
  kvp: true
  md:
    # metadata used in capabilities documents for WMTS
    # if the md option is not set, the metadata of the WMS will be
used
  title: Sampeu Base
  abstract: Base data provider for peuyeum geospatial framework
  online_resource: http://sampeu.peuyeum.com/
  contact:
    person: Rolly Maulana Awangga
    position: Software Engineer
    organization: Peuyeum
    address: Cibeunying
    city: Bandung
    postcode: 40191
    country: Indonesia
    phone: +62(0)813-12000-300
    fax: +62(0)813-12000-300

```

```

    email: rolly@awang.ga
    # multiline strings are possible with the right indention
    access_constraints:
        This service is intended for Peuyeum Geospatial Framework.
        The data is under development on peuyeum.com.
        (http://www.peuyeum.com/)
    fees: 'None'
wms:
md:
    title: Peuyeum WMS
    abstract: Peuyeum WMS Web Service.
layers:
    - name: sampeu
      title: Sampeu Base Geo Map - www.awangga.net
      sources: [sampeu_cache]
caches:
    sampeu_cache:
        grids: [ragi]
        sources: [sampeu_source]
sources:
    sampeu_source:
        type: mapserver
        req:
            layers: petaindo
            map: ./mapdata/mapfile/indo.map
            transparent: true
        coverage:
            bbox: [94.5011475, -11.007385, 141.01947, 6.076721]
            srs: 'EPSG:4326'
        mapserver:
            # for ubuntu
            # binary: /usr/lib/cgi-bin/mapserv
            # for mac os install via homebrew
            binary: /usr/local/opt/mapserver/bin/mapserv
            working_dir: ./mapdata/tmp
            supported_srs: ['EPSG:4326']
grids:
    ragi:
        base: GLOBAL_WEBMERCATOR
globals:

```

In the tag sources: it is a WMS input from the MapServer. Inside there is also the location of the Mapfile, where the MapServer file and working directory. Service in code is the output of web service in the form of WMTS with the restful format. Python script calls this configuration.

```

from mapproxy.wsgiapp import make_wsgi_app
application = make_wsgi_app(r'./sampeu.yaml')

```

This program code calls the wsgiapp submodule of the MapProxy module in python. Then declare the application variable result from the YAML configuration read. Sampeu.ini consist of IP configuration and port to serve the web service. This file contains the module variable that calls the form variable from the sampeu.py file.

```
[uwsgi]
pcre = True
http = 0.0.0.0:5473
module = sampeu:application
threads = 2
master = true
processes = 5
vacuum = true
die-on-term = true
```

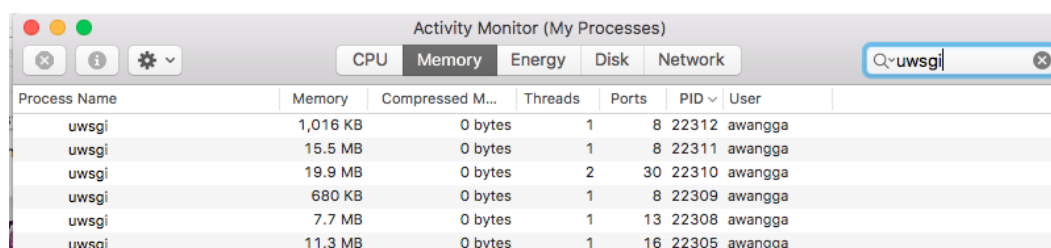
HTTP is the access address of the web service. 0.0.0.0 means we can access any IP on the server (without IP binding). 5473 is an access port of a web service. That is usually written behind IP in a browser with a colon separator (:). To run it then created a shell script with the name start.sh which contains the calling of this file by using the uWSGI module from Python already installed on the operating system.

```
#!/bin/bash
nohup uwsgi sampeu.ini &
```

Nohup is a feature of the operating system so that the service is still alive. This service is running on background with ampersand sign (&).

4. Experiment

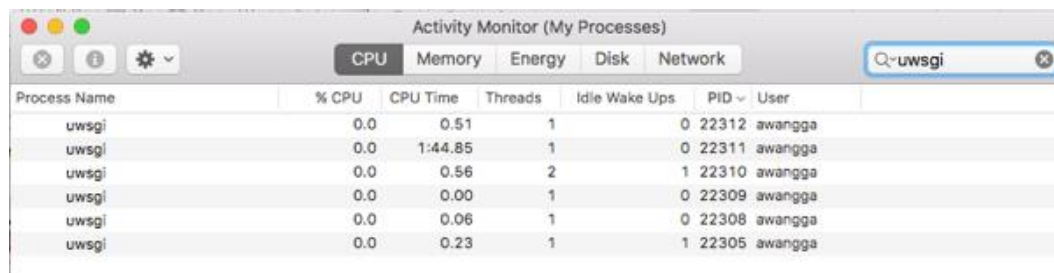
In this experiment, we will use shapefile districts city of Indonesia to form map Indonesia. Tinting and determination of provincial and district or city limits set in the indo.map Mapfile. Provincial boundaries are displayed when the default map of Indonesia. District or city limits will appear after reaching max. 5000000 on Mapfile.



The screenshot shows the Activity Monitor window with the 'Memory' tab selected. A search bar at the top right contains 'Q~uwsgi'. The table below lists several uwsgi processes with their memory usage, threads, ports, PIDs, and users.

Process Name	Memory	Compressed M...	Threads	Ports	PID	User
uwsgi	1,016 KB	0 bytes	1	8	22312	awangga
uwsgi	15.5 MB	0 bytes	1	8	22311	awangga
uwsgi	19.9 MB	0 bytes	2	30	22310	awangga
uwsgi	680 KB	0 bytes	1	8	22309	awangga
uwsgi	7.7 MB	0 bytes	1	13	22308	awangga
uwsgi	11.3 MB	0 bytes	1	16	22305	awangga

Figure 6. RAM monitoring of sampeu activity



The screenshot shows the macOS Activity Monitor window with the 'CPU' tab selected. A search bar at the top right contains the text 'uwsgi'. The table below lists several 'uwsgi' processes with their respective CPU usage, CPU time, threads, idle wake ups, PID, and user.

Process Name	% CPU	CPU Time	Threads	Idle Wake Ups	PID	User
uwsgi	0.0	0.51	1	0	22312	awangga
uwsgi	0.0	1:44.85	1	0	22311	awangga
uwsgi	0.0	0.56	2	1	22310	awangga
uwsgi	0.0	0.00	1	0	22309	awangga
uwsgi	0.0	0.06	1	0	22308	awangga
uwsgi	0.0	0.23	1	1	22305	awangga

Figure 7. CPU activity monitoring of Sampeu

The measurement aimed at monitoring the sampeu process on the activity monitor in the operating system (Figure 6). Websites that use web service until executed in a web browser with the action of using zooming map on the Peuyeum framework. The highest usage of CPU and RAM recorded with Process ID number (PID) of the sampeu process is 22310,22311,22312. These three PID are then summed to get the total value of RAM and CPU usage (Figure 7). Highest RAM usage value of 51.8 MB. Highest CPU usage on one thread is 80.2% of 2.3GHz. So, the CPU recommendation for sampeu is 1.8446 GHz.

5. Conclusions

This research tries to answer the need for OGC web service by using application based on an open source with best-selected performance. By using Sampeu framework, little RAM value was presenting. WMTS used in Sampeu Framework is one of a solution in accommodate computing efficiency of web service based on OGC. With the sampeu as a basic framework which is a merger between MapServer and MapProxy, it is expected to make research development of OGC web service will be more and more varied.

References

- [1] Sun Z, Di L, Fang H, Zhang C, Yu E, Lin L, Tan X and Yue P 2016 *Agro-Geoinformatics (Agro-Geoinformatics), 2016 Fifth International Conference on (IEEE)* pp 1–4
- [2] Alarabi L, Eldawy A, Alghamdi R and Mokbel MF 2014 *Proceedings of the 2014 ACM SIGMOD international conference on Management of data (ACM)* pp 897–900
- [3] Davis C A, Kimo Y J and Duarte-Figueiredo F L 2009 *Geoinformatics, 2009 17th International Conference on (IEEE)* pp 1–6
- [4] Purss M B, Gibb R, Samavati F, Peterson P and Ben J 2016 *Geoscience and Remote Sensing Symposium (IGARSS), 2016 IEEE International (IEEE)* pp 3610–3613
- [5] Jo J and Jang I 2016 *Information and Communication Technology Convergence (ICTC), 2016 International Conference on (IEEE)* pp 852–854
- [6] Chafiq T, Azmi R, Oulidi H J and Fekri A 2017 *Wireless Technologies, Embedded and Intelligent Systems (WITS), 2017 International Conference on (IEEE)* pp 1–5
- [7] Miao L, Guo J, Cheng W and Zhou Y 2016 *Geoinformatics, 2016 24th International Conference on (IEEE)* pp 1–4
- [8] Zhang Y, Li D and Zhu Z 2008 *Embedded Software and Systems Symposia, 2008. ICESS Symposia '08. International Conference on (IEEE)* pp 32–37
- [9] Trakas A and McKee L 2011 *Space Technology (ICST), 2011 2nd International Conference on (IEEE)* pp 1–5
- [10] Qomariah and Suhardi 2014 *2014 International Conference on ICT For Smart Society (ICISS) vol 2013 (IEEE) pp 7–13 ISBN 978-1-4799-6322-5 URL <http://ieeexplore.ieee.org/document/7013145/>*
- [11] Teslya N 2014 *Open Innovations Association FRUCT, Proceedings of 15th Conference of (IEEE)* pp 135–143

- [12] Schmid S, Galicz E and Reinhardt W 2015 *Military Technologies (ICMT), 2015 International Conference on* (IEEE) pp 1–6
- [13] Schoenhardt N C, Ioup E Z and McCreedy F P 2010 *OCEANS 2010* (IEEE) pp 1–7