

CZECH TECHNICAL UNIVERSITY IN PRAGUE
FACULTY OF CIVIL ENGINEERING
STUDY PROGRAM: GEODÉZIE A KARTOGRAFIE
BRANCH OF STUDY: GEODÉZIE, KARTOGRAFIE
A GEOINFORMATIKA



BACHELOR'S THESIS
DATABASE OUTPUT STORAGE SUPPORT IN PYWPS
FRAMEWORK

Vedoucí práce: Ing. Martin Landa, Ph.D.
Katedra geomatiky



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: Pišl

Jméno: Jan

Osobní číslo: 441

Zadávající katedra: Katedra geomatiky

Studijní program: Geodézie a kartografie

Studijní obor: Geodézie, kartografie a geoinformatika

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce: Možnosti integrace databázového úložiště v rámci frameworku PyWPS

Název bakalářské práce anglicky: Database Output Storage Support in PyWPS Framework

Pokyny pro vypracování:

Bakalářská práce se věnuje návrhu rozšíření frameworku PyWPS. PyWPS je jedna z implementací standardu Web Processing Service (WPS) v programovacím jazyku Python. Hlavní náplní práce je ověření možnosti databázového úložiště pro výstup jednotlivých procesů. V současnosti procesy ukládají výstupní data do souborových formátů uložených přímo na výpočetním serveru, ze kterého si je poté klient může stáhnout. Vzdáleného databázového úložiště by přesun výstupních dat směrem ke klientovi mohla zefektivnit. V rámci bakalářské práce se počítá s návrhem implementace zvoleného scénáře.

Seznam doporučené literatury:

Pilgrim, M.: Dive Into Python, Createspace Independent Pub 2009, ISBN: 9781441413024

OGC® WPS 2.0 Interface Standard

Obe, R.: PostGIS in Action, ISBN: 978-1935182269

Jméno vedoucího bakalářské práce: Ing. Martin Landa, Ph.D.

Datum zadání bakalářské práce: 9.10.2017

Termín odevzdání bakalářské práce: 14.1.2018

Údaj uveďte v souladu s datem v časovém plánu příslušného

Podpis vedoucího práce

Podpis vedoucího katedry

III. PŘEVZETÍ ZADÁNÍ

Beru na vědomí, že jsem povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s využitím poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je nutno uvést v bakalářské práci a při citování postupovat v souladu s metodickou příručkou ČVUT „Jak psát vysokoškolské závěrečné práce“ a metodickým pokynem ČVUT „O dodržování etických principů při přípravě vysokoškolských závěrečných prací“.

Datum převzetí zadání

Podpis studenta(ky)

ABSTRACT

To be implemented.

KEYWORDS

PyWPS, databases, Python, GDAL

ABSTRAKT

Bude doplněno

KLÍČOVÁ SLOVA

PyWPS, databáze, Python, GDAL

STATUTORY DECLARATION

I hereby declare that this bachelor thesis is completely my own work and that I used only the cited sources in accordance with the Methodical instruction about observance of ethical principles of preparation of university final projects.

V Praze dne

.....

(podpis autora)

Bude doplněno.

1 Introduction

The amount of data that is being collected and that people work with is increasing rapidly. That is due to the new technologies and instruments for collecting data that are being developed and used. While it provides data for more thorough and precise analysis than ever before, it also makes the data organization a much more difficult and complex task for which new methods had to be developed.

Relational databases are such a method. They are designed and organized especially for rapid search and retrieval in what can be a large amount of data. The database-management systems (DBMS) are a tool for managing and interacting with databases. DBMS have several major advantages over the traditional system where data is stored in files. Unlike the file management system, there can be more users accessing the same data concurrently without corrupting the data. Indexing speeds up the data retrieval operations. There is a standardized database language to use for queries. There are mechanisms such as data normalization that can be used to avoid duplicity of data and save storage space.

Being aware of these advantages, the main goal of this thesis is to explore the options of integrating an extension to the PyWPS framework that would allow the output data to be stored in a database. PyWPS is an implementation of the Web Processing Service standard from the Open Geospatial Consortium. The Web Processing Service (WPS) Interface Standard provides rules for standardizing inputs and outputs (requests and responses) for geospatial processing services. Written in Python, PyWPS enables integration, publishing and execution of Python processes via the WPS standard.

As of now, PyWPS is saving output data as files that are uploaded on the server. The client is then given a URL link from which they can download the file. If there was an option for the data to be stored in a database it could significantly improve the efficiency of the data transfer.

Also, the issue of actually implementing this extension to the PyWPS framework will be addressed in this thesis.

2 Theoretical part

2.1 Web Processing Services

2.1.1 OGC WPS Implementations

- PyWPS
- 52°North WPS
- ZOO Project

2.1.2 ESRI Web Processing

2.2 Geodatabases

- PostGIS
- Oracle Spatial and Graph
- SpatiaLite
- ArcSDE Geodatabase (ESRI)

3 Technologies

3.1 Python



Figure 3.1: Python logo (zdroj: Python.org)

3.2 GitHub

3.3 PyCharm

4 Practical part

4.1 Development

4.2 Functionality

5 Conclusion and future work