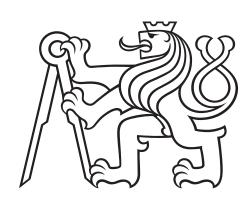
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE FAKULTA STAVEBNÍ PROGRAM GEODÉZIE A KARTOGRAFIE OBOR GEODÉZIE, KARTOGRAFIE A GEOINFORMATIKA



BAKALÁŘSKÁ PRÁCE MOŽNOSTI INTEGRACE DATABÁZOVÉHO ÚLOŽIŠTĚ V RÁMCI FRAMEWORKU PYWPS

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

leden 2018 Jan PIŠL

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE



I. OSOBNÍ A STUDIJNÍ ÚDAJE

Fakulta stavební Thákurova 7, 166 29 Praha 6

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Příjmení: Pišl	Jméno: Jan	Osobní číslo: 441
Zadávající katedra: Katedra geon	natiky	
Studijní program: Geodézie a ka	rtografie	
Studijní obor: Geodézie, kartogra	ifie a geoinformatika	
II. ÚDAJE K BAKALÁŘSKÉ PI	RÁCI	
Název bakalářské práce: Možnos	ti integrace databázového úložiště v rám	ci frameworku PyWPS
Název bakalářské práce anglicky:	Database Output Storage Support in Py	yWPS Framework
Web Processing Service (WPS) v p databázového úložiště pro výstup je souborových formátů uložených př vzdáleného databázového úložiště l	rozšíření frameworku PyWPS. PyWPS j programovacím jazyku Python. Hlavní ná ednotlivých procesů. V současnosti proce ímo na výpočetním serveru, ze kterého s by přesun výstupních dat směrem ke klie em implementace zvoleného scénáře.	áplní práce je ověření možnosti esy ukládají výstupní data do si je poté klient může stáhnout. I
Seznam doporučené literatury: Pilgrim, M.: Dive Into Python, Cre OGC® WPS 2.0 Interface Standard Obe, R.: PostGIS in Action, ISBN:		9781441413024
Jméno vedoucího bakalářské práce	: Ing. Martin Landa, Ph.D.	
Datum zadání bakalářské práce:		bakalářské práce: 14.1.2018 us datem v časovém plánu příslušného
Podpis vedoucího prá		Podpis vedoucího katedry
III. PŘEVZETÍ ZADÁNÍ		
poskytnutých konzultací. Seznan v bakalářské práci a při citování	en vypracovat bakalářskou práci samo n použité literatury, jiných pramenů postupovat v souladu s metodickou přír pokynem ČVUT "O dodržování etických	a jmen konzultantů je nutno ručkou ČVUT "Jak psát vysoko
Datum převzetí zadá	 iní	Podpis studenta(ky)

ABSTRAKT

Bude doplněno

KLÍČOVÁ SLOVA

PyWPS, databáze, Python, GDAL

ABSTRACT

To be implemented.

KEYWORDS

 $PyWPS,\ databases,\ Python,\ GDAL$

PROHLÁŠENÍ		
Prohlašuji, že bakalářskou práci na téma "Možnosti integrace databázového úložiště v rámci frameworku PyWPS" jsem vypracovala samostatně. Použitou literaturu a podkladové materiály uvádím v seznamu zdrojů.		
V Praze dne		
	(podpis autora)	

PODĚKOVÁNÍ

Bude doplněno.

1 Úvod

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.