A Microservice Architecture for the Processing of Large Geospatial Data in the Cloud

Michel Krämer
Urban planning use case

Goal:
Identify trees in 3D point clouds acquired by LMMS

Challenge:
Process data at least as fast as it is acquired, in order to be able to continuously monitor tree growth

Paparoditis et al. (2012)
Land monitoring use case

Goal:
Analyse topography and orographic precipitation to prepare against hazardous events (floods and landslides)

Challenge:
Being able to continuously monitor evolution of terrain for the first time
Problem statement

• Similar challenges in other use cases
  • change detection in urban areas
  • traffic pattern analysis
  • monitoring of seabed and coastal changes
  • etc.

• These use cases require “Big Geo Data”

• Goals cannot be reached with state-of-the-art approaches
Processing of large geospatial data

Users

Use platform

Cloud-based data processing

Contribute processing algorithms

Developers/researchers
Problem statement – Users

Requirements:
A platform to process Big Geo Data

My approach:
Microservice architecture for Cloud-based processing
Problem statement – Users

Requirements:

- A platform to process Big Geo Data
- Automate recurring processing tasks

My approach:

- Microservice architecture for Cloud-based processing
- Workflow management
Problem statement – Users

Requirements:

- A platform to process Big Geo Data
- Automate recurring processing tasks
- High-level interface for process automation

My approach:

- Microservice architecture for Cloud-based processing
- Workflow management
- Domain-Specific Language
Problem statement – Developers/researchers

Requirements:

Execute existing algorithms in the Cloud without modifications

My approach:

Service integration based on lightweight metadata
Problem statement – Developers/researchers

Requirements:
- Execute existing algorithms in the Cloud without modifications
- Focus on algorithms and not on the details of Cloud

My approach:
- Service integration based on lightweight metadata
- Workflow management
Problem statement – Developers/researchers

Requirements:

- Execute existing algorithms in the Cloud without modifications
- Focus on algorithms and not on the details of Cloud
- Orchestrate algorithms to workflows

My approach:

- Service integration based on lightweight metadata
- Workflow management
- Workflow management
A microservice architecture and Domain-Specific Languages can be used to orchestrate existing geospatial processing algorithms, and to compose and execute geospatial workflows in a Cloud environment for efficient application development and enhanced stakeholder experience."
Hypothesis

- microservice architecture
- Domain-Specific Languages
- orchestrate existing
- geospatial processing algorithms
- compose
- execute geospatial workflows
- Cloud environment
- efficient application development
- enhanced stakeholder experience
Microservice architecture  
Domain-Specific Languages  
Cloud

- Orchestrate existing geospatial processing algorithms
- Compose and execute geospatial workflows
- Efficient application development
- Enhanced stakeholder experience
Related work – Cloud for geospatial applications

Related work:
Mostly for Smart Cities
e.g. Khan, Anjum, & Kiani (2013), Krylovskiy, Jahn, & Patti (2015)

My approach:
Comprehensive platform for user-defined workflows
Related work – Cloud for geospatial applications

Related work:

Mostly for Smart Cities
  e.g. Khan, Anjum, & Kiani (2013),
  Krylovskiy, Jahn, & Patti (2015)

Single experiments
  e.g. Qazi, Smyth, & McCarthy (2013),
  Warren et al. (2015), Li et al. (2010)

My approach:

  Comprehensive platform for
  user-defined workflows

  Supports various use cases
Related work – Cloud for geospatial applications

Related work:

Mostly for Smart Cities
- e.g. Khan, Anjum, & Kiani (2013), Krylovskiy, Jahn, & Patti (2015)

Single experiments
- e.g. Qazi, Smyth, & McCarthy (2013), Warren et al. (2015), Li et al. (2010)

Specific platform/processing model
- e.g. Ajiy et al. (2013), Eldawy & Mokbel (2013)

My approach:

Comprehensive platform for user-defined workflows

Supports various use cases

Supports mix of multiple programming paradigms
Related work – Cloud for geospatial applications

**Related work:**

Mostly for Smart Cities
- e.g. Khan, Anjum, & Kiani (2013), Krylovskiy, Jahn, & Patti (2015)

Single experiments
- e.g. Qazi, Smyth, & McCarthy (2013), Warren et al. (2015), Li et al. (2010)

Specific platform/processing model
- e.g. Ajiy et al. (2013), Eldawy & Mokbel (2013)

No workflow management

**My approach:**

- Comprehensive platform for user-defined workflows
- Supports various use cases
- Supports mix of multiple programming paradigms
- Workflow management
System design
Architecture overview

Krämer & Senner (2015)
Architecture overview

Krämer & Senner (2015)
Architecture overview

Krämer & Senner (2015)
Architecture overview

Krämer & Senner (2015)
Architecture overview

Krämer & Senner (2015)
Processing services

Krämer & Senner (2015)
Benefits of microservice architecture

• Focus

• Independency

• Development distributability

• Composability

see also Krämer & Senner (2015)
for each [PointCloudCollection] do
  # service 35
  apply Resampling
    using resolution: 0.3

  # 66 C++ (IGN)
  apply PointCloudDimensionality
    using minSizeNeighbourhood: 16
    and maxSizeNeighbourhood: 128
    and windowSize: 0

  # 59 C++ (IGN)
  apply MultiObjectClassification

store

# 119 C++ (TUDelft)
apply PointCloudAttributeToTreeClass

//# 26 C++ (TUDelft)
apply IndividualTreeDetection

store
end
Architecture overview

Krämer & Senner (2015)
Dynamic workflow execution

Existing work:

A priori design-time knowledge (one DAG for the whole workflow)

My approach:

A priori runtime-time knowledge (iterative)
JobManager architecture
JobManager architecture
JobManager architecture
JobManager architecture
JobManager architecture
Controller

Execute workflow

- load metadata
  - start reasoning
  - register process chains
  - lookup process chain results
  - register workflow results

Process Chain Management

New process chains

No new process chains

Error

All process chains succeeded

Controller

Rule System
Controller

Execute workflow

load metadata

start reasoning

fire rules

register process chains

lookup process chain results

register workflow results

new process chains

no new process chains

all process chains succeeded

error

Controller

Rule System

Controller
Execute workflow

load metadata

start reasoning

fire rules

register process chains

lookup process chain results

all process chains succeeded

error

register workflow results

new process chains

no new process chains
Controller

Execute workflow

load metadata

start reasoning

register process chains

lookup process chain results

all process chains succeeded  error

register workflow results

fire rules

new process chains

no new process chains
Controller
Process Chain Manager

lookup next process chain

new

running

select node

request process chain status

execute process chain

finished

no node available

still running

register process chain results
Process Chain Manager

lookup next process chain

new

select node

running

request process chain status

no node available

still running

execute process chain

finished

register process chain results
Process Chain Manager

lookup next process chain

new

select node

running

request process chain status

no node available

still running

execute process chain

finished

register process chain results
Process Chain Manager

1. **lookup next process chain**
2. **new**
   - **select node**
3. **running**
   - **request process chain status**
4. **still running**
   - **execute process chain**
   - **finished**
5. **no node available**
   - **register process chain results**
Process Chain Manager

lookup next process chain

new

running

select node

request process chain status

no node available

still running

finished

execute process chain

register process chain results
Process Chain Manager

- lookup next process chain
  - new
  - running
  - select node
  - request process chain status
  - execute process chain
    - no node available
    - still running
      - finished
        - register process chain results
Controller

Execute workflow

load metadata

start reasoning

register process chains

lookup process chain results

register workflow results

error

all process chains succeeded

new process chains

no new process chains

fire rules

Controller

Rule System

Process Chain Manager

HTTP Server

Metadata Service

Data Storage

Workflow

Processing Chain 1

Processing Chain 2

Processing Chain n
Results
Urban planning use case

Dataset:
City of Toulouse, 120.63 GiB,
1.58 billion points, 529 tiles
Acquisition time: 1h 53m

Evaluation results:
Processing time: 1h 51m
18 compute nodes on Fraunhofer IGD Cloud
Almost linear scalability

Result: Point cloud with labels for individual trees
Urban planning use case

Dataset:
City of Toulouse, 120.63 GiB, 1.58 billion points, 529 tiles
Acquisition time: 1h 53m

Evaluation results:
Processing time: 1h 51m
18 compute nodes on Fraunhofer IGD Cloud
Almost linear scalability

Result: Point cloud with labels for individual trees
Land monitoring use case

Dataset:
3D point cloud of Liguria Region
451.16 GiB, 17.35 billion points, 684 strips

Evaluation results:
Processing time: 35m 49s
Previously (without my work): several days (*)

Result: Point cloud allowing for fast extraction of basins in a certain level of detail

* according to GIS users from the Liguria region (personal communication)
Scientific contributions

Microservice Architecture
- Scalability
- Modifiability
- Development distributability
- Availability

Processing
- Service integration
- Service orchestration
- Dynamic workflow management
- Rule-based workflow execution

Workflow Modelling
- Domain-Specific Language (DSL)
- Method for DSL modelling
Scientific contributions

Microservice Architecture
- Scalability
- Modifiability
- Development distributability
- Availability

Processing
- Service integration
- Service orchestration
- Dynamic workflow management
- Rule-based workflow execution

Workflow Modelling
- Domain-Specific Language (DSL)
- Method for DSL modelling
Conclusions

Research hypothesis is supported by

• Positive results from detailed evaluation
• Successfully satisfied stakeholder requirements
Conclusions

Research hypothesis is supported by
• Positive results from detailed evaluation
• Successfully satisfied stakeholder requirements

My thesis documents a major step in the paradigm shift from desktop GIS to the Cloud within the geospatial community and geoinformatics
(Future) work

JobManager has been put into production at Deutsche Telekom AG

Several improvements:

• Create VMs on demand (OpenStack)
• Capability-based scheduling
• Improved security
General features

- Geospatial feature store
- Schema agnostic
- Format preserving
- Cloud-based
- Event-driven
- Easy to use/integrate
Michel Krämer
Fraunhofer-Institut für
Graphische Datenverarbeitung IGD
Fraunhoferstraße 5
64283 Darmstadt
michel.kraemer@igd.fraunhofer.de