A framework for processing large scale geospatial and remote sensing data in MapReduce environment\[1\]

Presented by - Yiqing Liu
Yuanhao Chang
1. Introduction and related work
2. Background knowledge
3. The AEGIS framework
4. Distributed geospatial data processing
5. Conclusion
1. Introduction and related work

• Geospatial and remote sensing data is becoming enormously large, big data implementation in this field is needed.

• The industrial standard of big data are MapReduce model and its open-source implementation, the Apache Hadoop framework.

• The original aim of the MapReduce paradigm was to process simple text document, several extensions and toolkits have been introduced that operate over the Hadoop platform enabling the implementation and management of complex algorithms and data structures.
The MapReduce paradigms have already been successfully applied in multiple cases, leading to systems specialized for big spatial data storage and processing.

However, no existing solutions provide complete geospatial and remote sensing image processing functionality in the Hadoop framework.

Thus, the author propose a framework that enables the handling of spatial and remote sensing datasets based on MapReduce and Hadoop, and enable previously implemented algorithms and existing toolkits (on single machine) to be easily adapted to distributed execution without major effort.
2. Background knowledge

- Factory Pattern
- Abstract Factory Pattern
- Dynamic Factory Pattern
• Factory Pattern
• Abstract Factory Pattern
• Dynamic Factory Pattern
Which mouse we get is determined by the parameter we pass to `createMouse` function.
Factory Pattern

```
MouseFactory mf = new DellMouseFactory();
Mouse m = mf.createMouse();
m.sayHi();
```

- **Mouse**
  - + sayHi()

- **DellMouse**
  - + sayHi(){ "DellMouse" }
  - + Mouse createMouse(){ return DellMouse }

- **HpMouse**
  - + sayHi(){ "HpMouse" }
  - + Mouse createMouse(){ return HpMouse }

- **MouseFactory**
  - + Mouse createMouse()

工厂模式：
定义一个用于创建对象的接口，让子类决定实例化哪一个类。使一个类的实例化延迟到其子类。
• Factory Pattern
• Abstract Factory Pattern
• Dynamic Factory Pattern

When we have more than one kind of product...
Abstract Factory Pattern

Which kind of object we get is determined by which factory instance we use.

https://www.runoob.com/design-pattern/abstract-factory-pattern.html
When we need to add a new factory...

增加一个工厂(Asus)，需要增加一个工厂类(AsusFactory)，每个产品需要增加一个工厂-产品类(AsusMouse, AsusKeybo)

https://www.runoob.com/design-pattern/abstract-factory-pattern.html
When we want to add a new product...

增加一个产品(Mic)，需要增加一个产品父类(Mic)，每个工厂需要增加一个工厂-产品类(DellMic, HpMic)，工厂父类及所有工厂子类都需要增加此产品的创建

This is a lot of work!

Manually update all factory.

https://www.runoob.com/design-pattern/abstract-factory-pattern.html
• Factory Pattern
• Abstract Factory Pattern
• Dynamic Factory Pattern
Dynamically produce all product, without manual configuration to factory.

Figure 3 - Dynamic Factory Class Diagram

3. The AEGIS framework (in general)

• The AEGIS framework is based on the mainstream MapReduce paradigm, and the Apache Hadoop library.
• It is a platform independent library, implemented using .NET/Mono Framework.
• The primary goal of this framework is high adaptability and extensibility, which is achieved by the separation of working fields and the interchangeability of data models, methods and algorithms.
3. The AEGIS framework

- Data
- Data management
- Processing
- Extensibility and compatibility
3. The AEGIS framework

• Data
• Data management
• Processing
• Extensibility and compatibility
• OGC Simple Feature Access (SFA) standard: specifies a common storage and access model of mostly two-dimensional geometries (point, line, polygon, multi-point, multi-line, etc.) used by geographic information systems.

• OGC Spatial Referencing by Coordinates (SRC): a coordinate reference system that defines the coordinate space such that the coordinate values are unambiguous.
3. The AEGIS framework

- Data
- Data management
- Processing
- Extensibility and compatibility
Fig. 1. The core data management architecture of the AEGIS framework (UML notation).

Figure 3 - Dynamic Factory Class Diagram

Fig. 2. Integration of raster imagery support (UML notation).

3. The AEGIS framework

- Data
- Data management
- Processing
- Extensibility and compatibility
Fig. 4. Utilization of Orfeo toolbox functionality in AEGIS.
3. The AEGIS framework

- Data
- Data management
- Processing
- Extensibility and compatibility
• Can be changed to any toolbox

4. Distributed geospatial data processing

- Overview
- Data distribution and organization
- Data indexing and maintenance
- Data format
- MapReduce
• Overview
• Data distribution and organization
• Data indexing and maintenance
• Data format

- Multispectral space based partition
- VS
- Spectral space based partition

Before

After

Fig. 8. Histogram equalization performed on a partitioned geometry.
Thank you for your listening!