OMICS International
conferenceseries.com

International Conference and Exhibition on
Satellite

Satellite-2015

Houston, USA August 17-19, 2015

Flexible and Adaptive Processing of Earth Observation Data over HPC Architectures

CGIS
Computer Graphics and Interactive Systems

TECHNICAL UNIVERSITY OF CLUJ-NAPOCA

Dorian Gorgan
Computer Science Department
Technical University of Cluj-Napoca
http://users.utcluj.ro/~gorgan
dorian.gorgan@cs.utcluj.ro

Contents

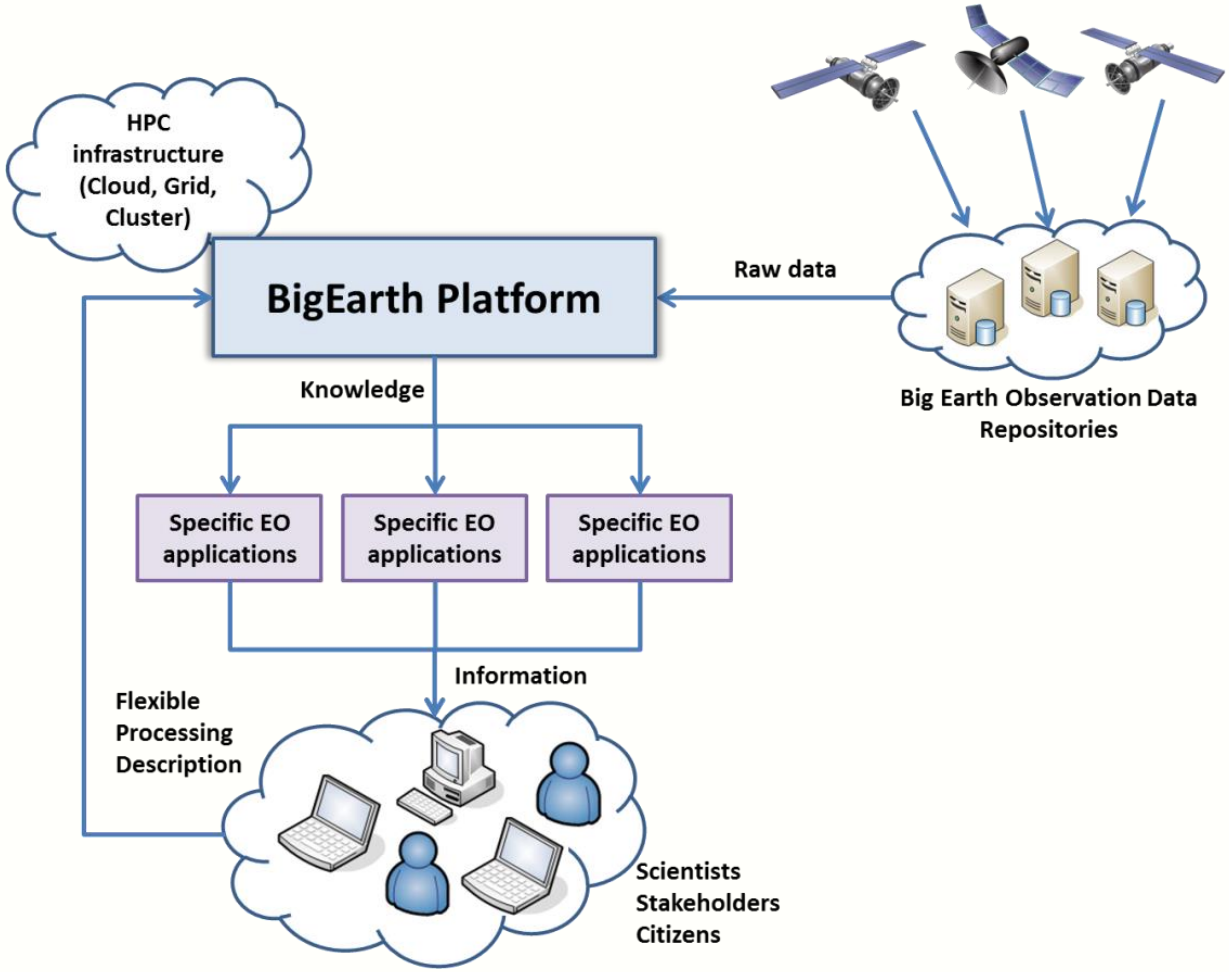
- Big data and Earth Observation data
- Satellite image data processing types
- BigEarth platform
- Process Description Languages
- WorDeL language
- WorDeL vs. MOML
- WorDeL vs. Python Scripting
- Experiments on WorDeL
- KEOPS - Kernel Operators
- BigEarth project

Big Data

- Huge data globally available
  - 2012, 2.8×10^{21} Bytes (2.8 ZB) at global level, 10 x 2007
  - 2020, 40 ZB (~14 x 2012)
- 3% marked/annotated, 0.5% analyzed
- Big Data – volume, variety, velocity, variability, veridity
- Earth Observation Data (EO Data)
- High costs of data management
- Increase data value by using instead of just storing
- Data -> knowledge -> information
- High Performance Computation resources + Analytics

Data, knowledge and information

- Flexible description and adaptive processing

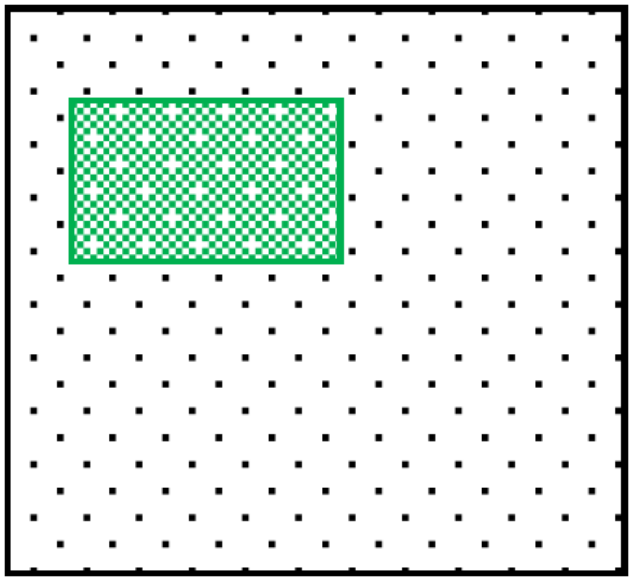


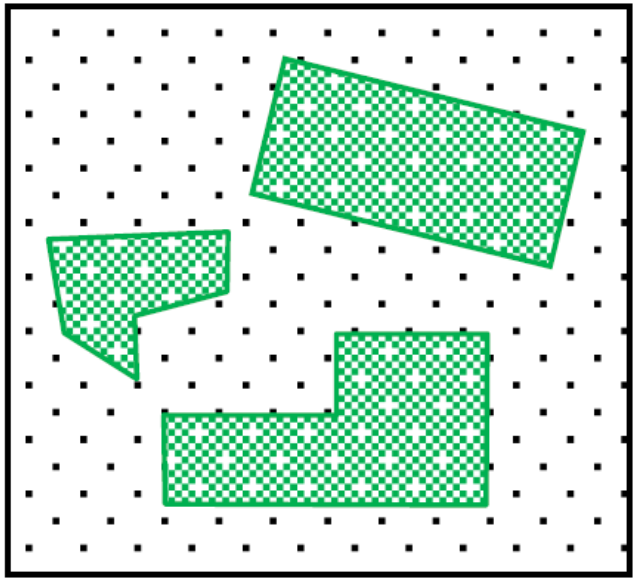
Satellite Image Data Processing Types

- Custom processing within the same input image
  - Earth data processing for retrieving information about specific geographic areas that are enclosed within the same image extent
  - Entire image vs. selected areas (bounding box, shapefiles)
- Indexed items processing
  - Information is defined based on structured models, such as matrices or arrays, where each element has a known position
- Unindexed items processing

Processing within Custom Area of Interest

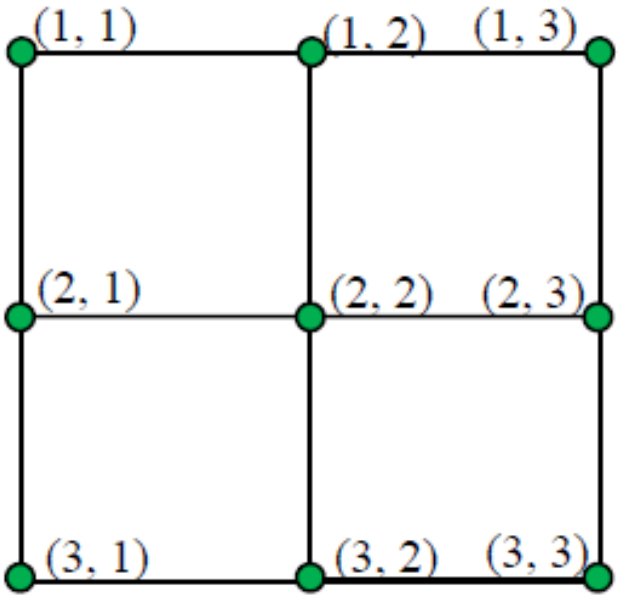
- Specify one area of interest vs. multiple areas of interest
  - E.g. Compare the vegetation growth over years in these regions, by computing the Normalized Difference Vegetation Index (NDVI) for each particular area
  - E.g. Change tracking, Urban evolution over years.

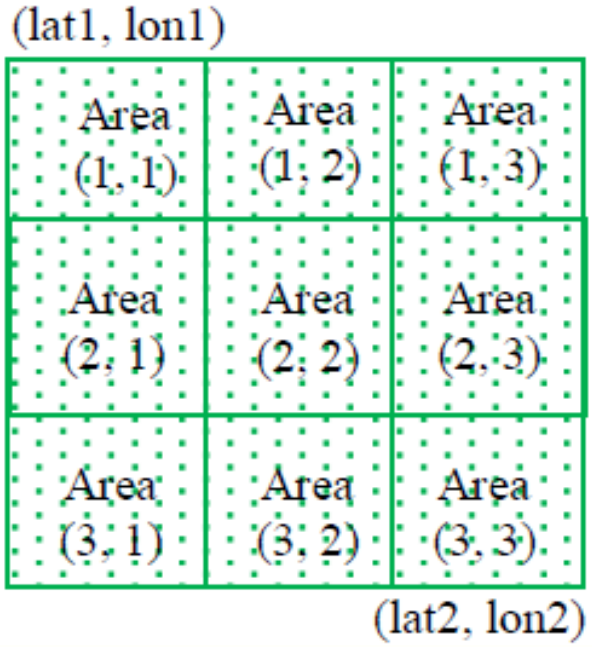




Indexed Items Processing

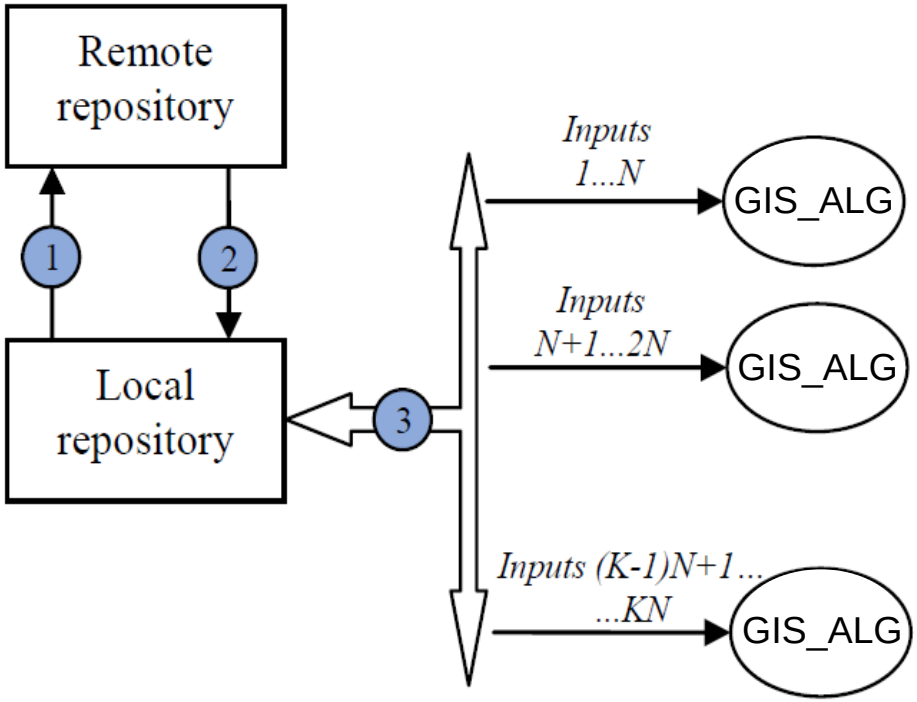
- Order a set of items by indexing them (e.g. list of satellite images, satellite data)
- Decompose the satellite image within equal inner areas
- Sub-areas are processed individually to extract information



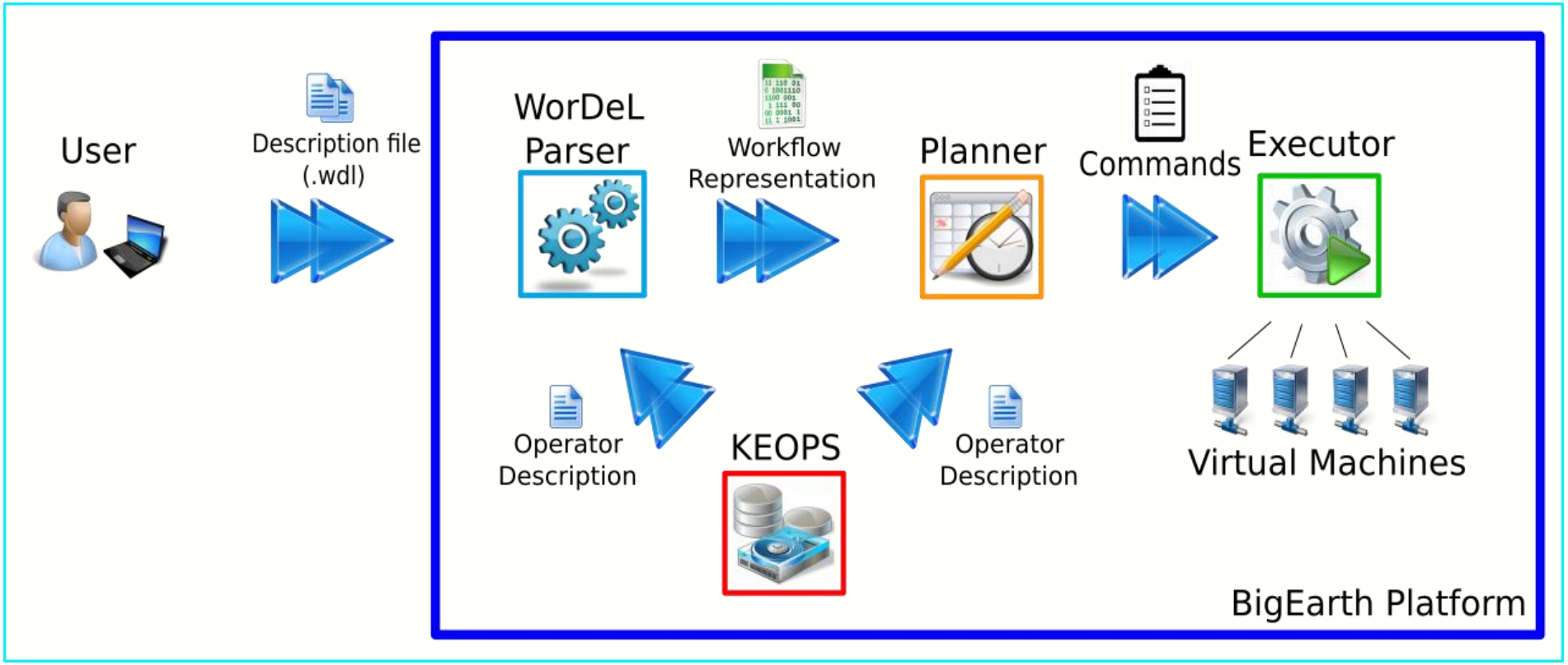


Unindexed Items Processing

- Items with unknown order are computed on the fly
- E.g. Different processing types that collect information from remote data feeders providing access to unindexed resources



BigEarth Platform



BigEarth Architecture

- WorDeL Parser
  - Communication between user and system
  - Interprets the processes description
  - Input and output description
- KEOPS Repository
  - KEOPS (Kernel Operators) is a collection of all operators which can be accessed by the process

BigEarth Architecture

- Planner
  - Interprets data provided by the user
  - Identifies relationships between atomic operators
  - Identifies the parallel and sequential tasks
- Executor
  - Prepares the inputs and launches the tasks into execution
  - Manages pools of execution machines
  - Identifies the parallel and sequential tasks
  - Launches in execution parts of the processing chain given by the user

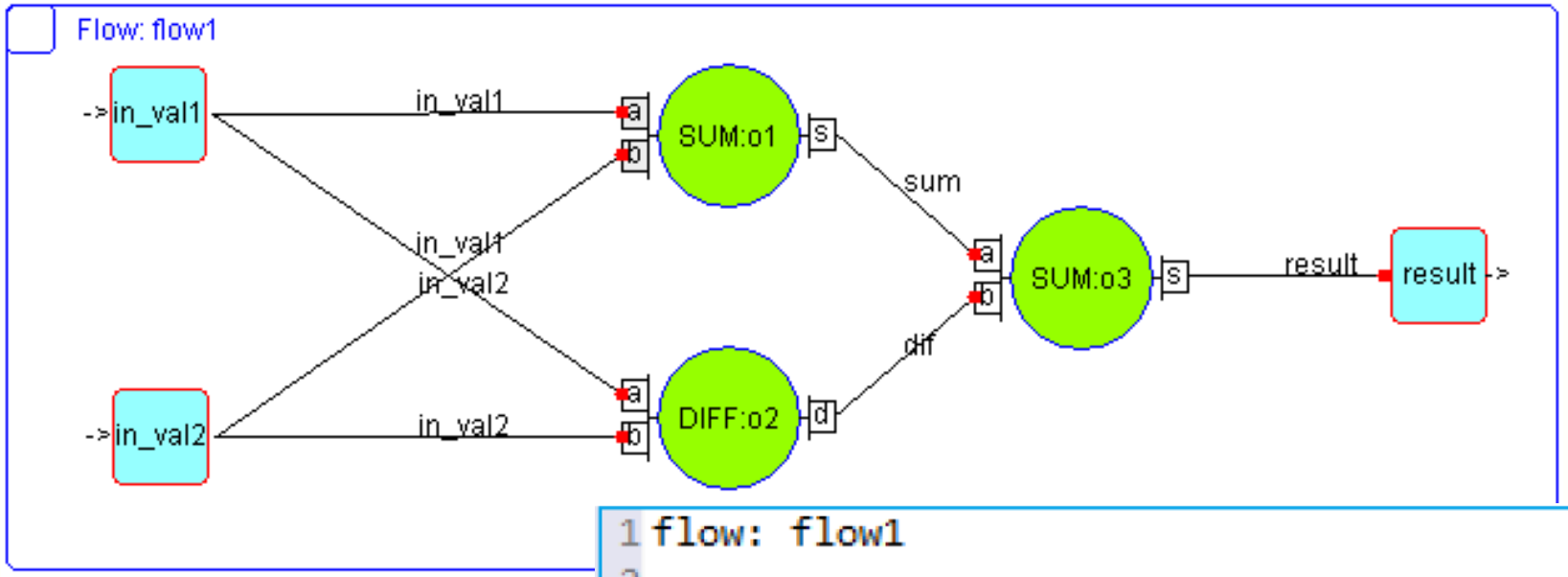
Process Description Languages

- BPEL, WS-BPEL, BPEL4WS (Business Process Execution Language for Web Services)
  - BPEL is OASIS standard (OASIS - Organization for the Advancement of Structured Information Standards)
  - Used in linking up web services=> creating more advanced entities, dubbed business processes
  - XML-based format
- MOML (Modeling Markup Language)
  - XML-based format
  - Possibility of defining relationships between entities without making any assumptions with regard to their meaning and interpretation

WorDeL Language

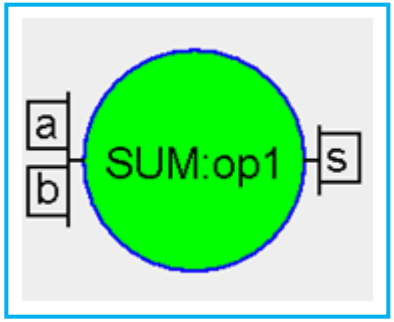
- Workflow Description Language (WorDeL)
- Describes the processing algorithms
  - Compact format
  - Intuitive
  - Allows the identification of parallelizable algorithm sections
  - Flexibility – easy to use, increase the efficiency

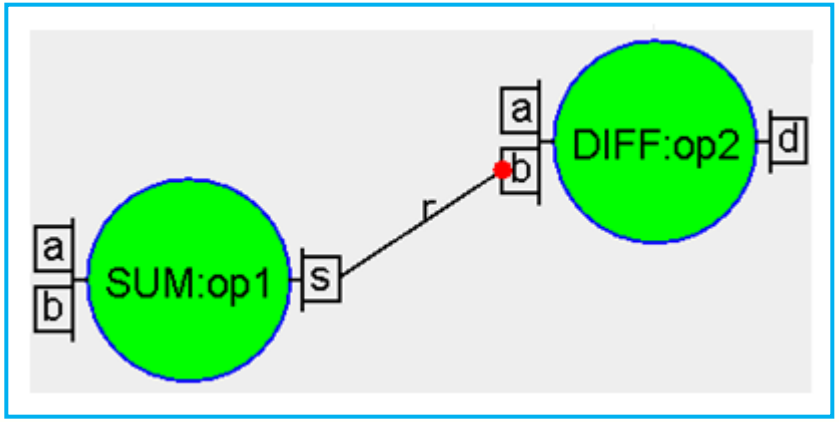
WorDeL Language



Operators and Connections

- Operator
  - Self-contained piece of software developed for a specific functionality
  - Input and output ports (interface)
- Connection
  - Link between two or many operators
  - One to many operators
  - Data type compatibility



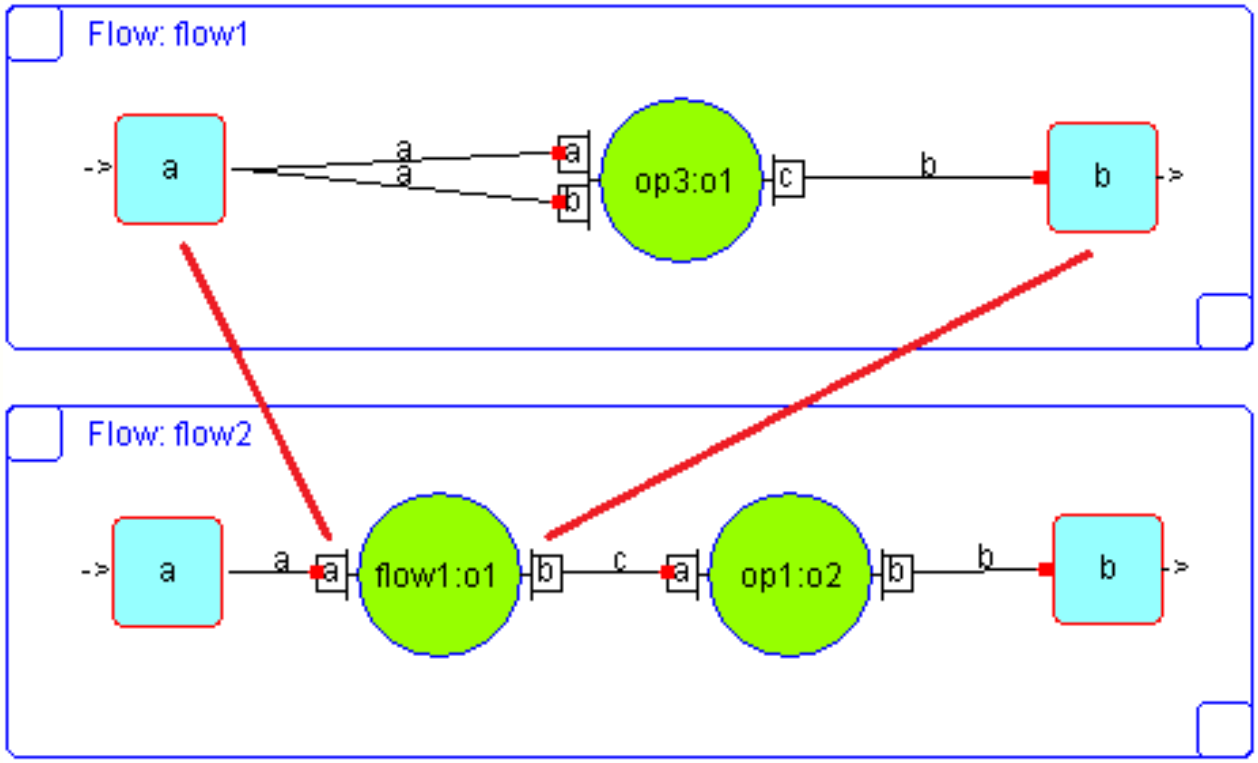


[a, b] SUM:op1 [r]

[a, r] DIFF:op2 [result]

Workflow

- Network of interconnected operators
- Process (algorithm) description
- Hyperstructure



Data Types

- Basic types: String, Integer, Float, Boolean, File
  - File type: checked just at execution (e.g. satellite images)

    E.g. Landsat, MODIS, etc.
- Aggregate types: List, Tuple
  - List: homogeneous collection of data

    E.g. List (<element_type>)
  - Tuple: heterogeneous collection of data

    E.g. Tuple (Integer, Float, String)

    Tuple (String, List (Integer))

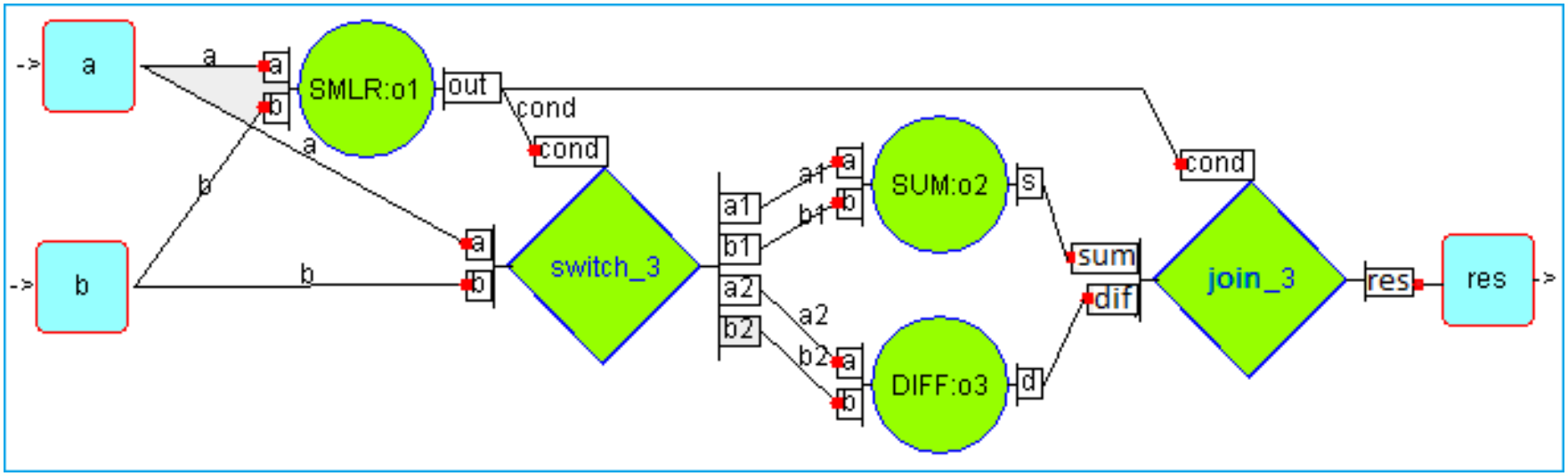
    Tuple (List (Tuple (List (List…..))))

Special Constructs

- Decisional constructs
  - Switch
  - Join
- Example:

  res = a+b, if a<b

  res = a-b, if a>b



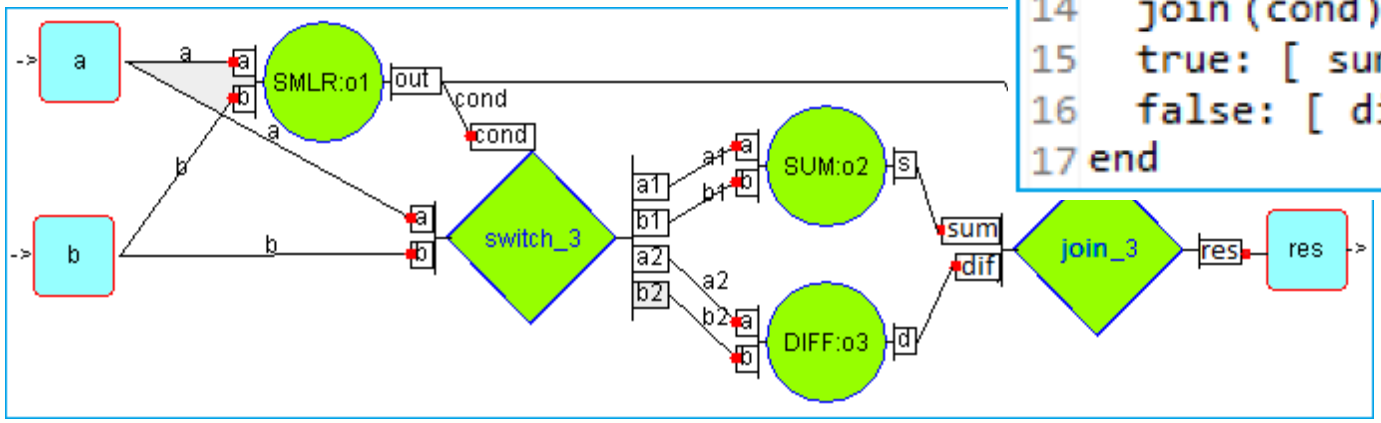
Special Constructs - Example

- Example:

  res = a+b, if a<b

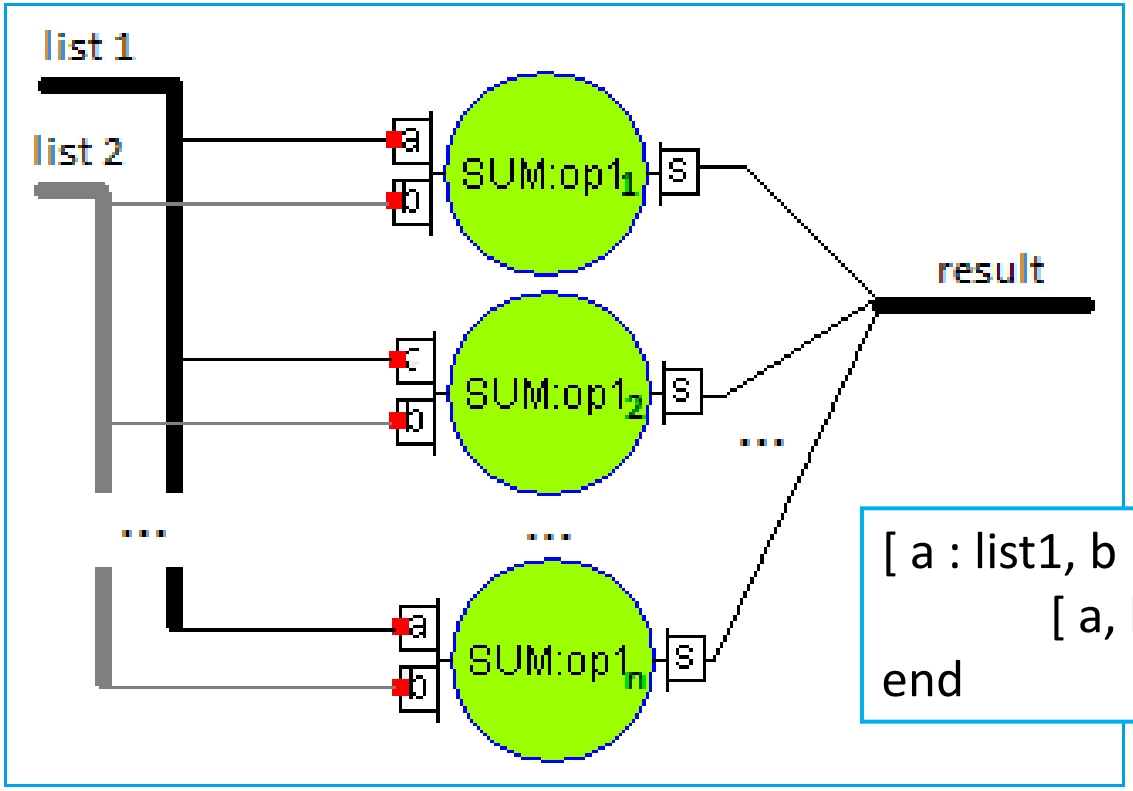
  res = a-b, if a>b

```
1  flow: flow1
2    input: a , b : Integer
3    output: res : Integer
4
5    [ a, b ] SMLR:o1 [ cond ]
6
7    [ a, b ] switch (cond)
8              true: [ a1 , b1 ]
9              false: [ a2 , b2 ]
10
11   [ a1, b1 ] SUM:o2 [ s ]
12   [ a2, b2 ] DIFF:o3 [ d ]
13
14   join (cond) [res]
15   true: [ sum ]
16   false: [ dif ]
17 end
```



Special Constructs

- Repetitive constructs: For-each
  - Repetitive processing on different data




```
[ a : list1, b : list2 ] foreach ( ) [ r : result ]
                [ a, b ] SUM: op1 [ r ]
end
```

Program Structure

- Include section – includes external sections/ files
- Definition section – workflow definition
- Process section – processing tasks

```
1  include flow1.wdl
2
3  flow: flow2
4     Inputs: a:String;
5     Outputs: b:String;
6   [ a ] flow1: o1 [ c ]
7   [ c ] op1: o2 [ b ]
8  end
9
10 process: p1
11   var: inString = "test_string";
12    [ inString ] flow2: instance [ result.dat ]
13 end
```

Include section

Interface definition

Workflow body

Workflow Description section

Processing section

WorDeL vs. MOML

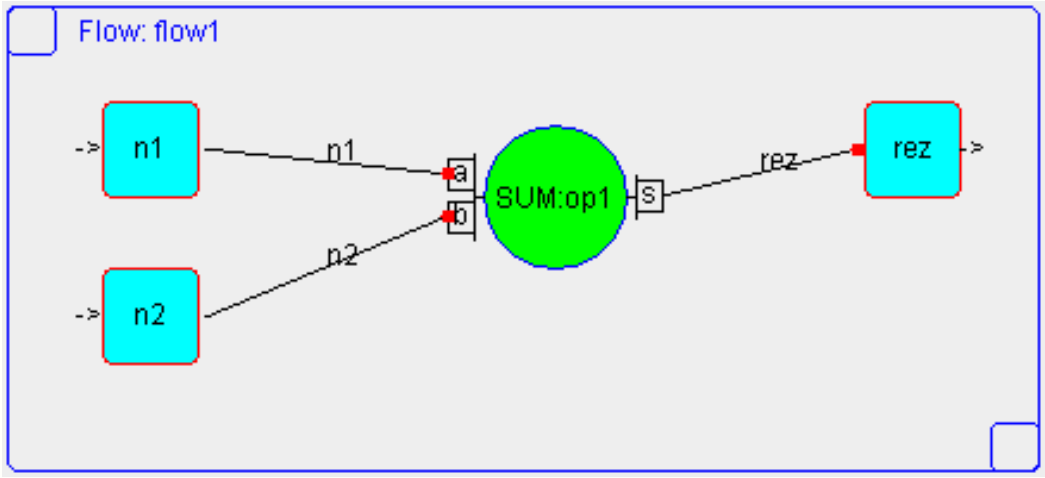
- Example: sum of two integers
- Compact form of WorDeL

```
<entity id="flow1">
    <entity id="op1" class="SUM">
        <port id="a" type="in" data="Integer"/>
        <port id="b" type="in" data="Integer"/>
        <port id="s" type="out" data="Integer"/>
    </entity>

    <port id="n1" type="in"/>
    <port id="n2" type="in"/>
    <port id="rez" type="in"/>

    <relation id="n1a" in="n1" out="a"/>
    <relation id="n2b" in="n2" out="b"/>
    <relation id="sRez" in="s" out="rez"/>
</entity>
```

MOML




```
flow: flow1
  input: n1,n2 :Integer
  output: rez :Integer

        [ n1, n2 ] SUM : op1 [ rez ]

end
```

WorDeL

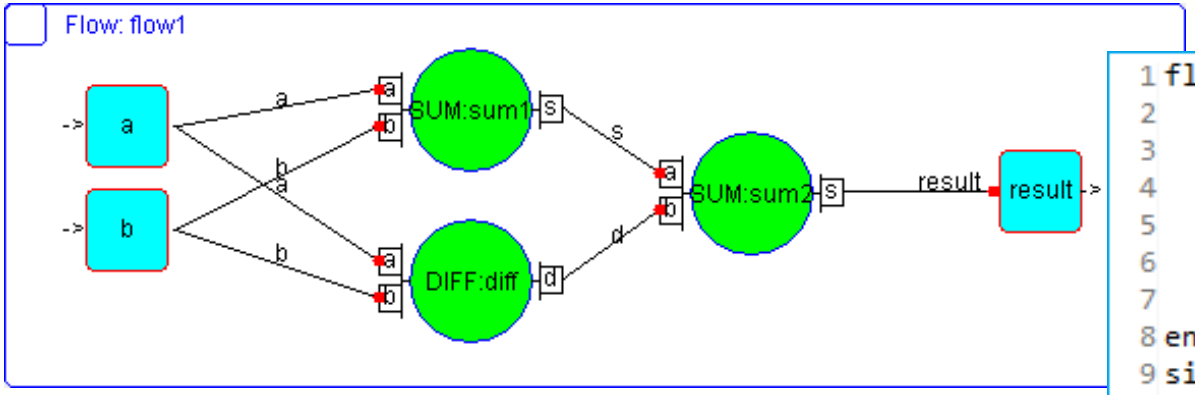
WorDeL vs. Python Scripting

- Compute: E=(a+b)+(a-b)
- Use executable programs for each arithmetic operation; Store the result in a given file – parameter of the command line

```
1 a=25
2 b=10
3 result='result.dat'
4 s='interm1.dat'
5 d='interm2.dat'
6 grass.run_command('pack/SUM.exe', first = a , second=b, output = s);
7 grass.run_command('pack/DIFF.exe', first = a , second=b, output = d);
8 grass.run_command('pack/SUM.exe', first = s , second=d, output = result);
```

Python

WorDeL




```
1 flow: flow1
2     input: a,b:Integer
3     output: result:Integer
4
5     [ a, b ] SUM:sum [ s ]
6     [ a, b ] DIFF:diff [ d ]
7     [ s, d ] SUM:sumFinal [ result ]
8 end
9 simulation:s1
10    [ 25, 10 ] flow1:instance1 [res.dat]
11 end
```

WorDeL vs. Python Scripting

- Python:
  - Python provides a programming-centered approach for defining data processing algorithms
  - Users - full responsibility for the intermediate files management
- WorDeL:
  - Specifically for describing a network of interconnected entities. Has a straight-forward, compact format, based on a “black-box” approach
  - The entities might well be Python scripts themselves
  - WorDeL looks less like a programming language and more like a format for representing workflows
  - Users don’t need to manage files or invoke the operators

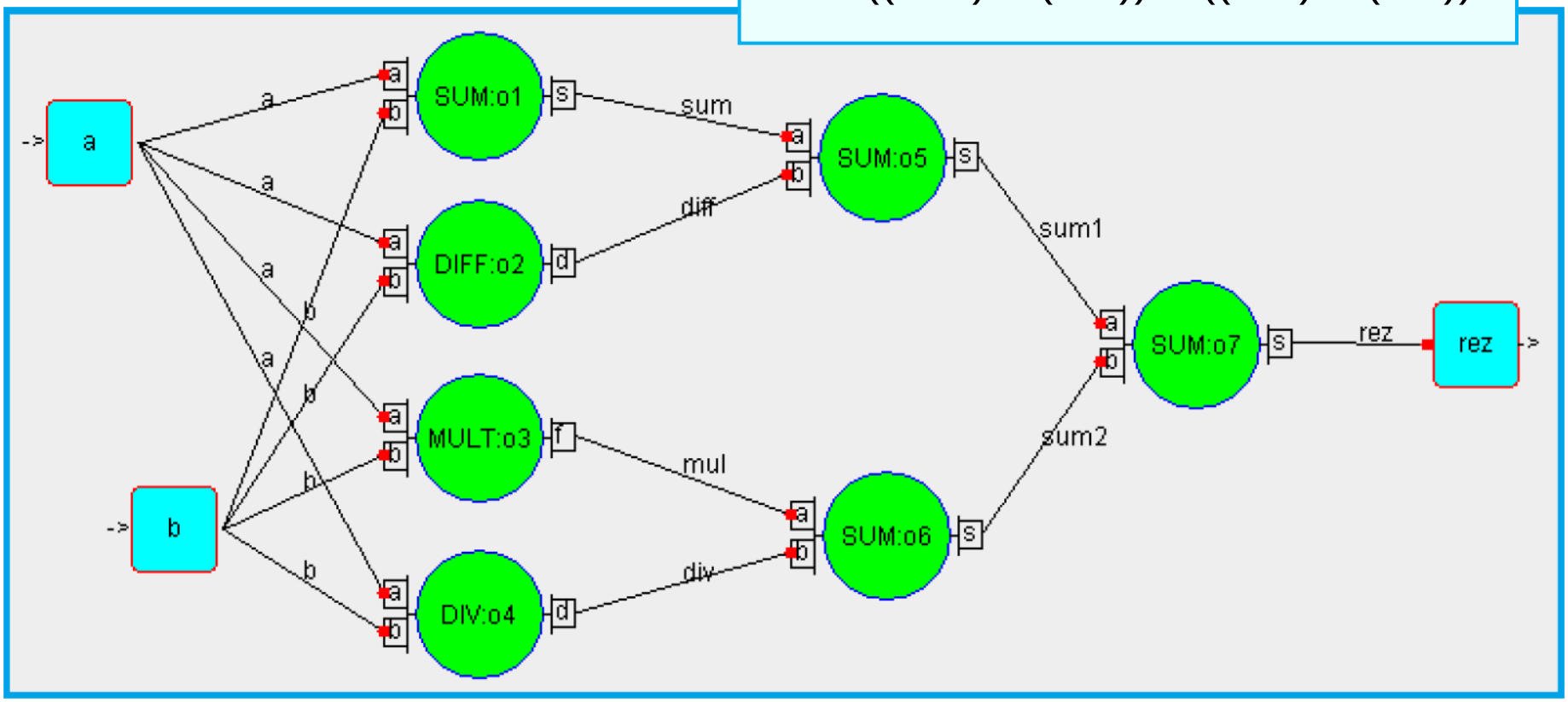
Experiments on WorDeL

- Integrating WorDeL within the BigEarth platform
- Computation of: E = (a+b) + (a-b) + a*b + a/b
- Experimental considerations:
  - a, b, E: integers
  - Computation time estimated as:

    “+” (2 sec), “–” (2 sec), “*” (3 sec), “/” (4 sec)
- Objectives:
  - Functionality
  - Computation time
  - Parallelism

Experiments: Process Workflow

E = ((a+b) + (a-b)) + ((a*b) + (a/b))



Experiments: WorDeL Description

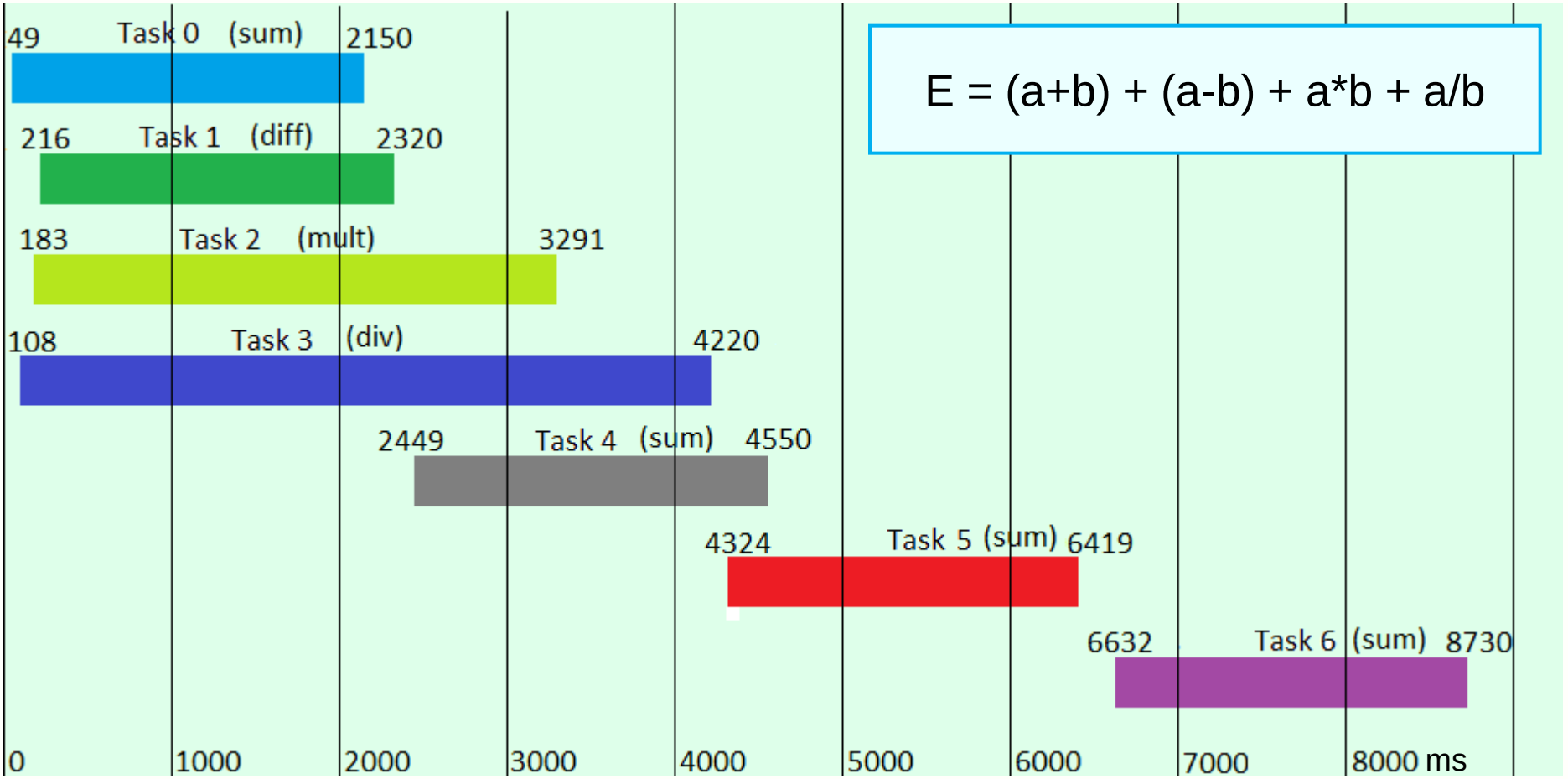
E = (a+b) + (a-b) + a*b + a/b

```
flow: flow1
  input: a,b:Integer
  output: rez :Integer

        [ a, b ] SUM : o1 [ sum ]
        [ a, b ] DIFF : o2 [ diff ]
        [ a, b ] MULT : o3 [ mul ]
        [ a, b ] DIV : o4 [ div ]

        [ sum, diff ] SUM : o5 [ sum1 ]
        [ mul, div ] SUM : o6 [ sum2 ]
        [ sum1, sum2 ] SUM : o7 [ rez ]
end
simulation: s1
    [ valueA, valueB ] flow1 : instance1 [ result ]
end
```

Experiments: Task Execution Diagram



8,730 ms of the parallel execution vs 17,716 ms of the sequential execution.

Conclusions on WorDeL Language

- Compact
- Intuitive
- Flexible
- Integration within the BigEarth platform
- Simple way of defining data processing algorithms
- Distribution and execution within a HPC infrastructure
- Efficient processing big geospatial data

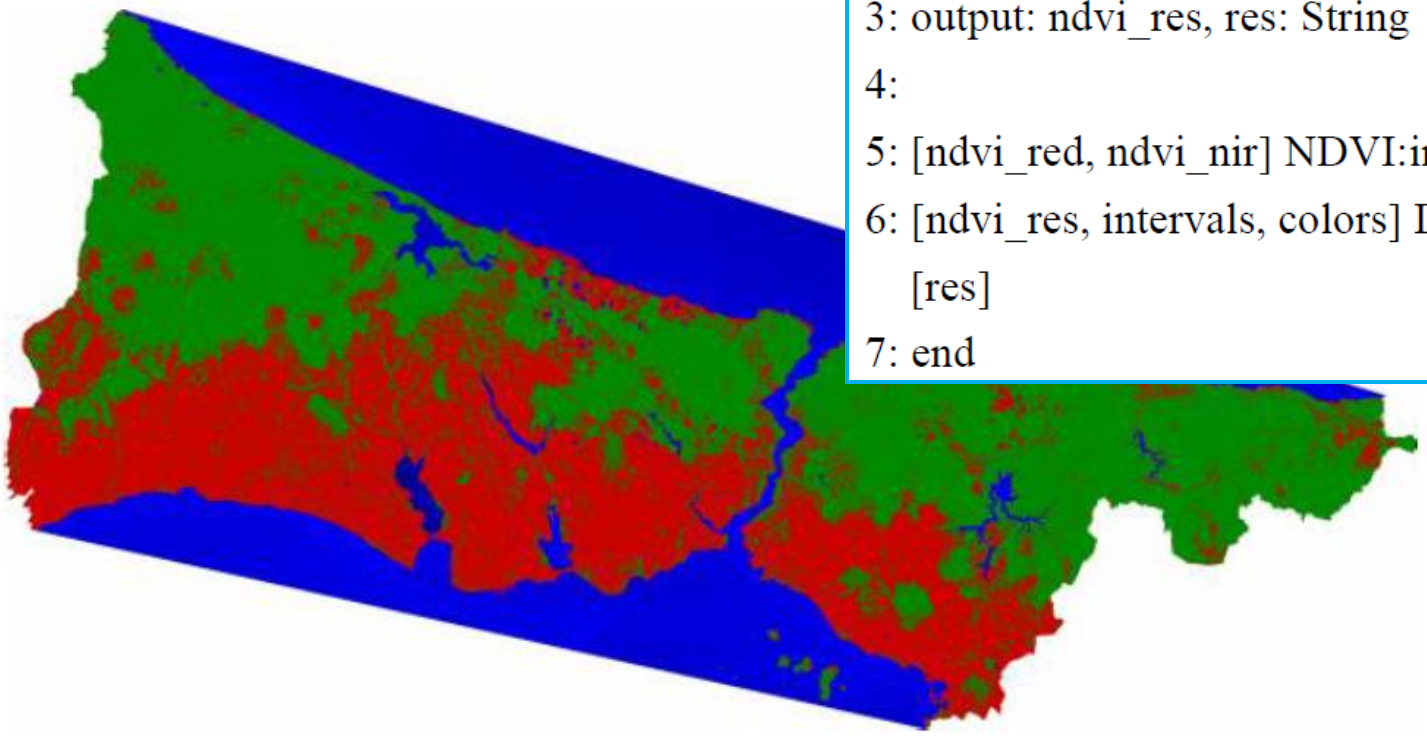
KEOPS – Kernel Operators

- Collection of atomic functionalities used to develop EO oriented complex processing
  - Algorithms for computing the greenness of specific geographic locations, arithmetic operations applied on EO data (e.g. adding a constant to a satellite image), operators for managing satellite bands (e.g. mosaic, crop, etc.)
- KEOPS is based on GRASS (Geographic Resources Analysis Support System)
- KEOPS was developed initially within GreenLand platform for geospatial data management, satellite image processing, interactive visualization, etc.
- Built in system within the BigEarth platform

Experiments on KEOPS and WorDeL

- Combines two operators: NDVI (Normalized Difference Vegetation Index) and Density Slicing

```
1: flow: KEOPS_use_case
2: input: ndvi_red, ndvi_nir, intervals, colors: String
3: output: ndvi_res, res: String
4:
5: [ndvi_red, ndvi_nir] NDVI:instance1 [ndvi_res]
6: [ndvi_res, intervals, colors] DensitySlicing:instance2
   [res]
7: end
```



BigEarth Project

Funded by ROSA (Romanian Space Agency) and ESA (European SA)

BigEarth project, http://cgis.utcluj.ro/bigearth/

- Description
  - Explore techniques and methodologies to develop and execute analytics on Big Earth Data
  - Provide flexible and interactive description of HPC processing (High Performance Compution)
  - Adaptive HPC based computation
- Focus on
  - Access massive data, knowledge, and information
  - User access to simple and complex processing algorithms
  - Processing scheduling
  - Execution performance
- HPC platforms
  - Grid, Cloud, Multicore, GPU Clusters

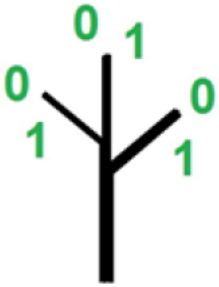
Publications

1. Gorgan D., Mihon D., Bacu V., Rodila D., Stefanut T., Colceriu V., Allenbach K., Balcik F., Giuliani G., Ray N., Lehmann A., Flexible Description of Earth Data Processing over HPC Architectures, Big Data From Space Symposium, Frascati, Italy, 5-7 June, Abstract Book, pp. 39 (2013).
2. Nandra C. I., Gorgan D., Workflow Description Language for defining Big Earth Data Processing Tasks, Proceedings of the ICCP-2015 Conference (in press).
3. Gorgan D., Giuliani G., Ray N., Cau P., Abbaspour K., Charvat K., Jonoski A., Lehmann A., Black Sea Catchment Observation System as a Portal for GEOSS Community, in International Journal of Advanced Computer Science and Applications (IJACSA), pp.9-18, (2013).
4. Mihon D., Bacu V., Colceriu V., Gorgan, D., Modeling of Earth Observation Use Cases through KEOPS System, Proceedings of the ICCP-2015 Conference (in press).
5. Bacu V., Stefanut T., Gorgan D., Adaptive Processing of Earth Observation Data on Cloud Infrastructures Based on Workflow Description, Proceedings of the ICCP-2015 Conference (in press).

ACKNOWLEDGMENTS

- This research is supported by ROSA (Romanian Space Agency) by the Contract CDI-STAR 106/2013, *BIGEARTH -Flexible Processing of Big Earth Data over High Performance Computing Architectures*.

- The scientific consultancy and technology transfer has been supported by MEN-UEFISCDI by Contract no. 344/2014, *PECSA - Experimental High Performance Computation Platform for Scientific Research and Entrepreneurial Development*.



OMICS International
conferenceseries.com

International Conference and Exhibition on
Satellite

Satellite-2015

Houston, USA August 17-19, 2015

Many thanks for your attention!

CGIS
Computer Graphics and Interactive Systems

TECHNICAL UNIVERSITY OF CLUJ-NAPOCA

Dorian Gorgan
Computer Science Department
Technical University of Cluj-Napoca
http://users.utcluj.ro/~gorgan
dorian.gorgan@cs.utcluj.ro