Scientific workflow optimization using systemlogs
Provenance data integration for workflows.

B.A. Blaauwgeers
University of Amsterdam
alexander.blaauwgeers@os3.nl

Research Project 1 Presentation
Supervisor: Zhiming Zhao

July 5, 2018
These days workflows becoming more complex because of new techniques like **distributed cloud based systems**. There is a need to integrate the different sources of provenance data generated by workflows.

- Workflow management is there to control the **flow** of information.
- The service is a **black box** for the Workflow management system.
- Workflow management aims to create an **abstract** of more than one autonomous systems and their functions.
The main question for this research is:

*How can different sources of provenance data generated by scientific workflows be integrated to allow analysis?*

The research question can be divided into multiple sub-questions:

1. What main types of (scientific) workflow exist?
2. What kind of system logs are available according specific use cases?
3. How can the workflow system be integrated with system logs?
Theory: Complexity of workflow (1)

- Amount of distributed autonomous systems.
- Accessibility and availability of logs. (Application, Syslog, PROV)
- Geographical and logical location. (Local, Remote, Cloud, ...)
Theory: Chain of logs (2)

- **Network**
  - Network infrastructure
  - Routers, switches
  - Route table, NAT

- **OS**
  - Physical machines
  - Hypervisor and VM management
  - Java Virtual Machine (JVM)
  - Kernel and OS, CPU, memory, operating system log, ...

- **Storage**
  - Physical storage
  - Logical storage
  - Shared or distributed storage

- **DB**
  - Performance
  - Accessibility
  - Consistency
  - Integrity

- **Middleware**
  - Message Queuing
  - Enterprise Service Bus (ESB)
  - Application Interfaces (e.g. RESTapi / SOAP)

- **Application**
  - Input
  - Output
  - Errorlog

Source: Blaauwgeers B.A., November 2015
Theory: Types of workflows (3)

There are different types of workflows, e.g.

- linear
- recursive
- parallel
- decision-based
Workflow: An example of an workflow
Workflow: Workflow with integrator
$fname["xml"] = "post." . $_GET["flowlabel"] . "." . time() . ".xml";
$fname["log"] = "post." . $_GET["flowlabel"] . "." . time() . ".log";
$fname["data"] = "post." . $_GET["flowlabel"] . "." . time() . ".data";

$fdata = file_get_contents('php://input');
file_put_contents($fname["xml"],$fdata);

# get the file
$flowout = file_get_contents($fname["xml"]);

#Replace the lilagal character before parsing to the simple object.
$flowbuffer = str_replace(".","-",$flowout);

#Convert the xml-string value to an array via a simple xml object.
$flowxml = simplexml_load_string($flowbuffer);
$flowarray = XML2Array($flowxml);
$flowarray = array($flowxml->getName() => $flowarray);

#grep the wps data.
$flowwps = $flowarray["wps-ExecuteResponse"]["wps-ProcessOutputs"]["wps-Output"]["wps-Data"]["wps-ComplexData"][

#grep the locator of d4s
$flowdfs["data"] = str_replace("http-", "http:=",$flowwps[0]["d4science-Data"]); #Log of the computation
$flowdfs["log"] = str_replace("http-", "http:=",$flowwps["d4science-Data"]); #output_file_name

file_put_contents($fname["log"], fopen($flowdfs["log"], 'r'));
file_put_contents($fname["data"], fopen($flowdfs["data"], 'r'));
?>
Results: Integrator log

Inspecting log files on the integrator hub:

```
ablauweers@pa:/var/log/nginx$ sudo zgrep "rpa" access.log.6.gz | grep java | tail -n 8
```

B.A. Blaauwgeers (University of Amsterdam)  Provenance data integration for workflows  July 5, 2018 10 / 16
Results: Intergrator flow data

Inspecting flow files on the integrator hub

```
ablaauwgeers@ipa:/var/www/html/rpa$ ls -la | grep "post.0020T"
-rw-r--r-- 1 www-data www-data 900278 Jun 29 08:52 post.0020T.1530276728.data
-rw-r--r-- 1 www-data www-data 4441 Jun 29 08:52 post.0020T.1530276728.log
-rw-r--r-- 1 www-data www-data 2270 Jun 29 08:52 post.0020T.1530276728.xml
-rw-r--r-- 1 www-data www-data 10 Jun 29 08:13 post.0020T.init.1530274439.init
```
Results: Logdata on external application

Logfiles of the example application of the research group\(^1\).

```
ablauwgeers@desktop-30:~/Downloads/cue_service_logs$ grep "0020T"
localhost_access_log.2018-06-29.txt
geospatial_lat_min=38.000&geospatial_lat_max=38.200&geospatial_lon_min=147.000&geospatial_lon_max=147.100&flowlabel=0020T HTTP/1.1" 200
28172
geospatial_lat_min=38.000&geospatial_lat_max=38.200&geospatial_lon_min=147.000&geospatial_lon_max=147.100&flowlabel=0020T HTTP/1.1" 200
28172
```

\(^1\)With thanks to Spiros Koulouzis
Log data might be influenced by (business-)policies, containerization, time-zones, namespacing, (NAT-) translation.
  ▪ Can be reduced by adding a flowlabel and timestamp to the requests.

Access rights to the log files are required.

Some integrator functions might break when components of the workflow change.

Apache Taverna was used as Workflow Manager for the experiment.
The usage of an **integrator** hub as service is useful during the integration of workflows.

It is recommended to add a **flowlabel** and **timestamp** to the API calls.

This integrator **collects and enriches** the log files during the execution.

The collection of log files can be used to create a **timeline** on which **analysis** can be performed.

The **accessibility** of log data is important for completeness of the picture.
Future work

- Should be tested with different kind of Workflow Managers.
- Improvement on the integration to support more services.
- To make the integrator more resilient to workflow changes.
- Distribution strategy of the integrator and limit the single point of failure.
Questions?