Sandbox apps and workflow: schema in Parsl: design, implementation and application lifecycle

1 Vincenzo Cimmino, 2 Dante D. Sánchez-Gallegos, 3,4 Yadu Babuji, 1 Diana Di Luccio, 3,4 Kyle Chad, 2 José Luis Gonzalez-Compean and 2 Raffaele Montella

dante.sanchez@cinvestav.mx

1 Department of Science and Technologies, University of Naples “Parthenope”, Naples, Italy.
1 Cinvestav Tamaulipas, Victoria, Mexico.
3 University of Chicago, Chicago, USA
4 Argonne National Laboratory, Lemont, USA
Introduction

• We extend workflow engine.
  
  • To support a new type of Sandbox App:
    
    • isolates each task in a scratch directory in a straightforward and transparent fashion.
    
    • using the workflow:// schema.
  
  • To address the following challenges:
    
    1. the definition of task dependencies.
    
    2. the execution of each task in an isolated sandbox.
    
    3. the data staging in the sandbox.
The workflow:// schema uniquely identifies a task instance in the workflow.

The workflow:// schema uniquely identifies a task instance in the workflow. Designed for Batch tasks, it is used for variables/files mapping for Native/Web/IoT.

- **Workflow**
  - The schema label
  - The workflow unique name
    - An UUID could be used
    - If empty means “current workflow”
  - The task unique name
    - Can be dynamically generated by the Python script when the workflow is created programmatically.

- **Stager**
  - Stager
  - One Of...
    - Local
    - Shared File System
    - Remote scratch directory on physical machine, virtual instance or container

- **External software sandbox**
  - One Of...
    - ln
    - scp
    - Grid
    - ftp

- **Scratch directory root**
  - workflow:///task_a/results.csv

- Designed for Batch tasks, it is used for variables/files mapping for Native/Web/IoT.
The Sandbox App in Parsl

The Sandbox App implements the following operations:

- Create a scratch directory.
- Resolve the workflow:// schema.
- Stage the input files.
- Execute the command in the scratch directory.

```python
@sandbox_app
def hello(workflow_app_name='', project=''):  
    return '""echo "hello world" > out.txt"

Listing 1: Definition of a Sandbox App
```
The Sandbox App in Parsl

- Sandbox:
  - _task_dep: dict
  - _app_name: str
  - _workflow_name: str
  - _workflow_directory: str
  + generate_unique_label(label: str): str
  + create_working_directory()
  + count_deps(script: str): int
  + find_task_by_name(task_name: str): int
  + resolve_workflow_schema(script: str):
  + define_command(script: str): str
  + operation()

- SandboxStager:
  - _hostname: str
  - _username: str
  + cp_command(src: str, dst: str): str
  + ftp_command(src: str, dst: str): str
  + scp_command(src: str, dst: str): str
  + _init__()

- AppBase:
  - wrapped_remote_functions
  - remote_fn
  - dataflow_kernel: DataFlowKernel
  - executors: str
  - cache: bool
  - ignore_for_cache: cost
  - func: function
  + __init__(func, data_flow_kernel=None, ...)}
  + __call__(*args, **kwargs)

- SandboxApp:
  - wrapped_remote_functions
  - remote_fn
  - dataflow_kernel: DataFlowKernel
  - executors: str
  - cache: bool
  - ignore_for_cache: cost
  - func: function
  + __init__(func, data_flow_kernel=None, ...)}
  + __call__(*args, **kwargs)
Conclusions

• We proposed an approach for integrating the workflow:// schema in Parsl.

• We designed and implemented a new type of Parsl App, called a Sandbox App.

• Using this approach, we automated the data staging phases of an external software task.

• We implemented these features and tested them using Parsl's LocalProvider.

• As a short-term future research plan, we will perform an extensive performance analysis.
Vincenzo Cimmino, Dante D. Sánchez-Gallegos, Yadu Babuji, Diana Di Luccio, Kyle Chad, José Luis Gonzalez-Compean and Raffaele Montella

dante.sanchez@cinvestav.mx

1 Department of Science and Technologies, University of Naples "Parthenope", Naples, Italy.
1 Cinvestav Tamaulipas, Victoria, Mexico.
3 University of Chicago, Chicago, USA
4 Argonne National Laboratory, Lemont, USA