Real-Time Micro-service Operating Infrastructure for Scientific Workflows

Klara Nahrstedt (klara@Illinois.edu)
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Background: Increasingly data-driven and interdisciplinary scientific research

- **Key enabling factor**: Network connected scientific instruments capable of real-time data capture

![Digital microscope](image-url)
Example: Typical experimental process in material science research

1. Create physical experimental sample
2. Prepare analytical sample from the physical experimental sample
3. Analyze the analytical sample
4. Convert digital analytical data
5. Transport the converted analytical data
6. Interpret the converted analytical data
7. Modify the procedure for creation of the physical experimental sample

It typically takes 20 years to go from the discovery of new materials to fabrication of new and next-generation devices*

*Source: National Science and Technology Council’s report, 2011
Challenges

- Heterogeneous scientific data management and processing
- Support ad hoc and complex data analysis workflows
- Shorten time from digital capture to interpretation & insights
- Real-time data capture and acquisition
- Analytics support to gain insights from data
From Nanofabrication to Nano-manufacturing

ZEISS ORION NanoFab at the Centre for Neural Engineering, University of Melbourne. Courtesy of Peter Hines, QUT, Queensland, Australia

Helium-ion microscopy of ZnO nanorods/Ag nano-particles as biosensors materials. Courtesy of Eric Flaim, University of Alberta.

Our approach

✓ Micro-service execution environment

✓ Feedback control resource adaptation
Long-tail scientific data processing challenges

- **Challenges**: Support execution of heterogeneous types of data processing & analysis workflows

- Previous work often employs a monolithic approach in workflow implementation and execution
  - E.g.: Pegasus, Taverna, Kepler, etc.
  - Run on large-scale & homogeneous datasets

Executing workflows on grid infrastructure
Micro-service execution environment

- **Micro-service**: software development technique – variant of Service-oriented architecture style that structures app as a collection of loosely coupled services
- **Micro-services over monoliths**: Each task is modeled as a micro-service
  - Use publish-subscribe middleware to connect between micro-services
- Separate task dependencies from task implementation & deployment
  - Enable flexible workflow composition
  - Task-level resource provisioning
Task A’s Request Queue

Task A’s Consumer

Forward request to next task

Task B’s Request Queue

Task B’s Consumer

Workflow Invoker

TDS Ensemble

Submit analysis workflow

Workflow Composition

Curation Service

Upload raw scientific data

Task A

Workflow description

Task coordination

Distributed resource management system

Example: Executing scientific data processing workflow

Workflow Invoker

TDS: “Start at A”

Task A’s Request Queue

Round-robin dispatching

Task A’s Micro-service

TDS: “Next task is B”

Task B’s Request Queue

Task B’s Micro-service

Curation Service

Upload raw scientific data

Return curation results

Start  A  B  End
Our approach

- Micro-service execution environment

- Feedback control resource adaptation
Micro-service management: Challenges & opportunities

- **Challenges**: Heterogeneity in data formats, workflow structures, and usage patterns

- **Opportunities**:
  - Micro-service architecture enables new abstraction of resources
  - Advanced control, optimization, and learning capabilities over system data for resource adaptation

Number of files generated on a JOEL instrument from Sep 2015 to May 2016 (MRL Lab @UIUC)
Self-adaptive micro-service infrastructure

Micro-service adaptation layer
- Control optimization
- System model
- Updated control policy
- System states
- System identification

Micro-service monitoring layer
- Alert engine
- Resource actuator
- Operational log DB
- Visualizer
- Micro-service log collector

Micro-service execution layer
- Workflow invoker
- Task dependency service
- Task A
- Task B
- Task C

Infrastructure layer
- Distributed resource management system

Administrative interactions

Heterogeneous Data processing & analysis workflow requests
Micro-service system model & parameters

- **Micro-service resource model:**
  - Resources are represented by the number of task consumers

- **Workflow performance metrics:**
  - Average delay of each workflow type & over all types

---

**Workflow type 1**

- Task A’s No. of consumers
- Task B’s No. of consumers
- Task C’s No. of consumers
- Task D’s No. of consumers

**Workflow type 2**

- Task A’s consumer
- Task B’s Consumer

**Workflow type 3**

- Task A
- Task B

**Micro-service Workflow Execution**

- Workflow type 1 average delay
- Workflow type 2 average delay
- Workflow type 3 average delay
Adaptive micro-service system implementation

Micro-service adaptation layer
- SciPy
- TensorFlow
- Updated control policy
- System model
- System states

Micro-service monitoring layer
- Kapacitor
- InfluxDB
- Grafana
- Resource actuator
- Micro-service log collector

Micro-service execution layer
- Workflow invoker
- Apache Zookeeper
- Docker
- RabbitMQ

Infrastructure layer
- Resource management system
- Kubernetes

ILLINOIS
Evaluation: Micro-service resource adaptation

• Data processing workflows:
  – MDP: material data processing workflows (to process output of digital microscopy, such as DM3, AFM, etc.)
  – LIGO: analyze data to study stars and black holes

MDP workflows

LIGO workflows
Effectiveness of neural network-based system identification

(a) MDP

(b) LIGO