

f(X)

funcX

Serverless Supercomputing

Yadu Babuji, Tyler Skluzacek, Ryan Chard, Kyle Chard, Ben Blaiszik, Anna Woodard, Zhuozhao Li, Steve Tuecke, Ian Foster

yadu@uchicago.edu



THE UNIVERSITY OF
CHICAGO



Scientific Computing Landscape

New class of data-intensive research

Scientific data

- Exploding volumes and velocities
- Acquired at various locations/times
- Analysed on distributed resources
- Interactive (on-demand) analyses

Research computing infrastructure

- Typically high barriers to entry
- Batch queues with long/unpredictable waits
- Heterogeneous software & hardware environments
- Strict usage requirements
 - Batch queues, specific request sizes, walltimes

Aim: to easily compute wherever it is most suitable



Quick Primer on Serverless

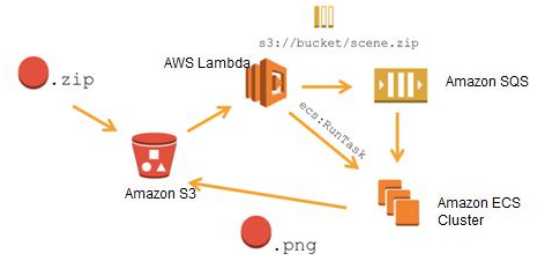
Cloud-computing execution model, where the provider dynamically and transparently manages allocation of machine resources.

E.g., Function as a Service (FaaS)

- Pick a runtime (python/JS/R etc.)
- Write function code
- Run (at scale)

Low latency, on-demand, elastic scaling

Combine functions to solve complex problems



FaaS is a great match for Science

- Functions as the building block for pipelines that run anywhere.
- FaaS offloads the complexities of research CI from the user
 - Auth
 - Batch systems
 - Data transport
- Rich programming models built on top of FaaS can address workflows, event driven compute and more.

funcX: Serverless Science

Turn **any** machine into a function serving endpoint

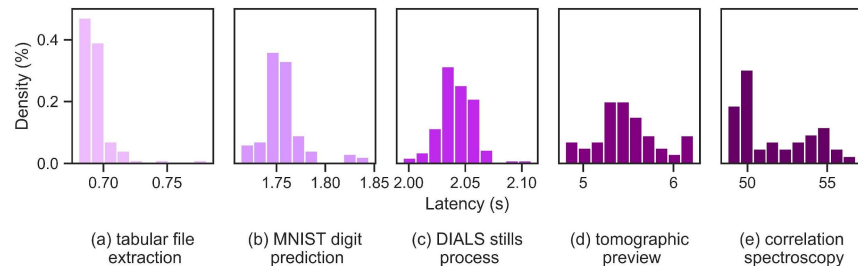
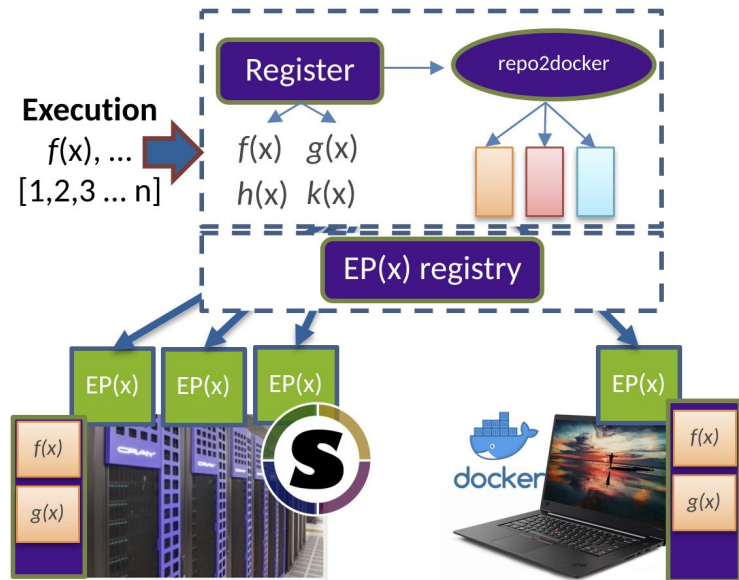
Remove barriers to using diverse and distributed infrastructure

Functions:

- Register once, run anywhere
- Encapsulated in a container
- Authn/z for execution and sharing*

Endpoints:

- Lightweight agent that can be deployed by users
- Abstracts underlying resource and elastically scales to demand



funcX: Service

REST Web interface

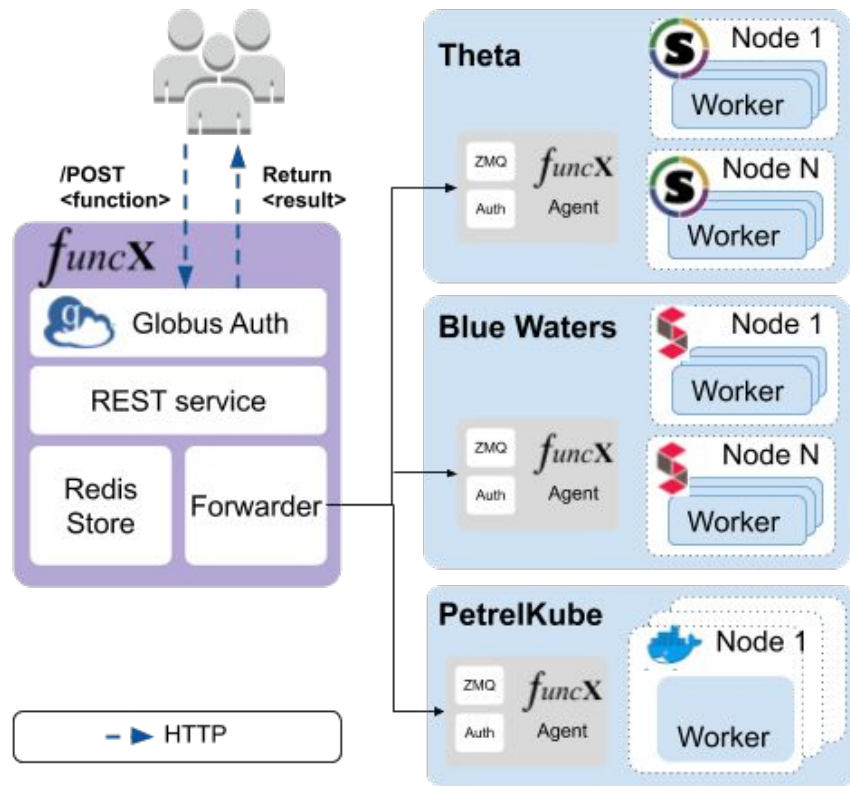
- Register and manage endpoints
- Publish and invoke Python functions
- Globus Auth for authn/z

Redis store

- Store and share functions
- Track and allocate tasks
- Reliable endpoint task queues

Endpoint forwarders

- Forward serialized functions and inputs for execution



funcX: Endpoint agent

Secure communication

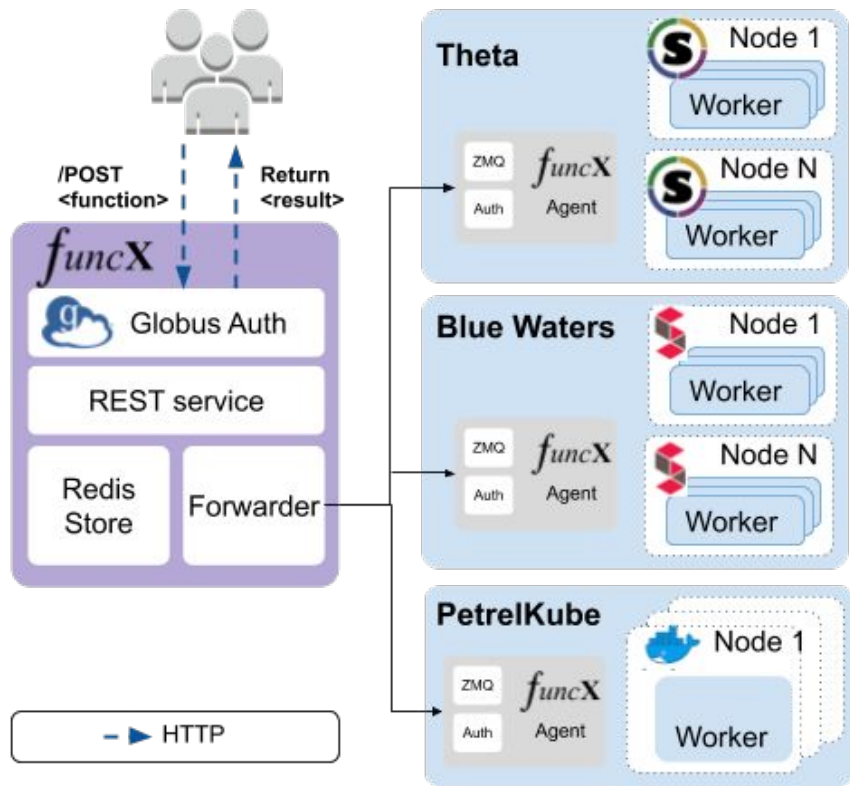
- Securely connect **out** to forwarder for registration
- ZeroMQ for low latency comm.
- Retrieve and queue tasks

Compute abstraction

- Acquire nodes from diverse compute resources (using Parsl)
- Deploy workers inside containers to nodes

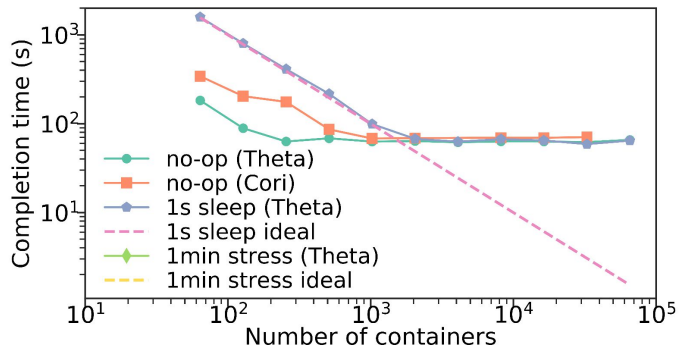
Endpoint

- Report usage stats and liveness

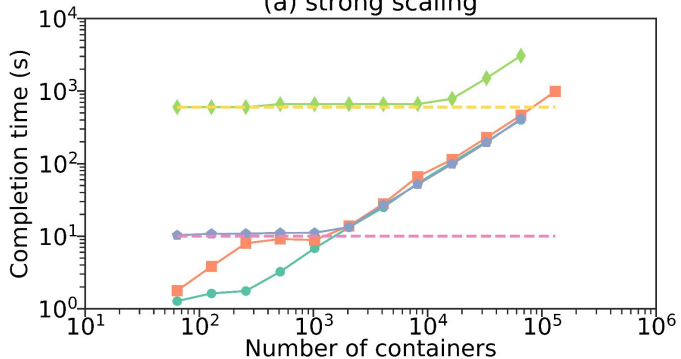


funcX: Performance

Strong and Weak scaling

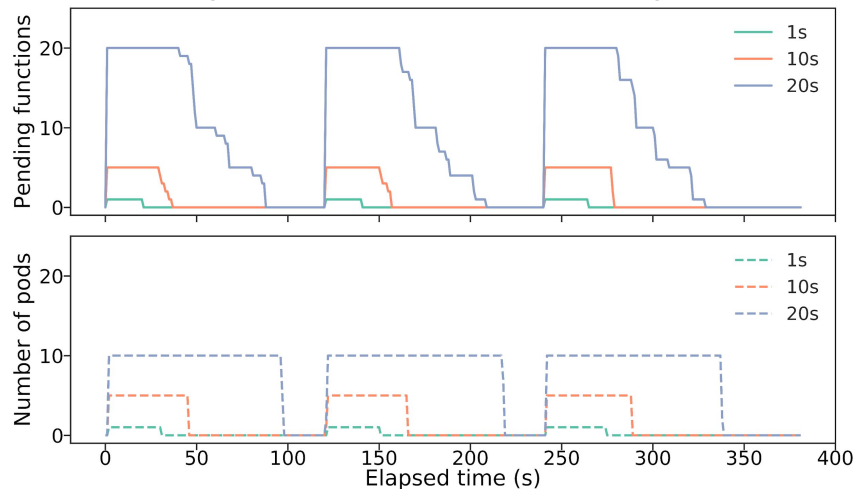


(a) strong scaling

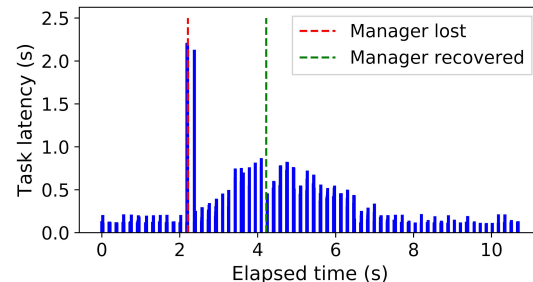
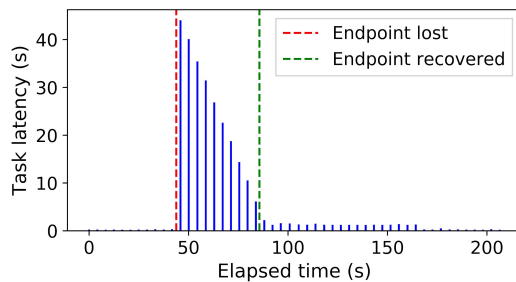


(b) weak scaling

Autoscaling over k8s (petrelkube@Argonne)



Fault tolerance and recovery at multiple components



Demo

<https://funcx.org>

