RADU POPESCU, EP-SFT

TO BOLDLY GO…
APPLICATION DEPLOYMENT FRAMEWORKS FOR NEW COMPUTING ENVIRONMENTS

EXPLOITING HPC RESOURCES FOR HEP IS NOT SIMPLE

HPC machines will represent an increasing percentage of the compute resources available for high-energy physics HEP, in particular in runs 3 and 4, when a large increase in workloads is expected.

We are able to make some use of HPC machines, but the deployment/integration story is not nearly as simple as on the grid:

- Different I/O profiles to native HPC workloads
- Limited connectivity in certain cases
- HEP has complex stack with many moving parts
- HPC sites are less flexible than grid sites
- Most allocations on HPC are opportunistic

How can we make it easier to use HPC resources?

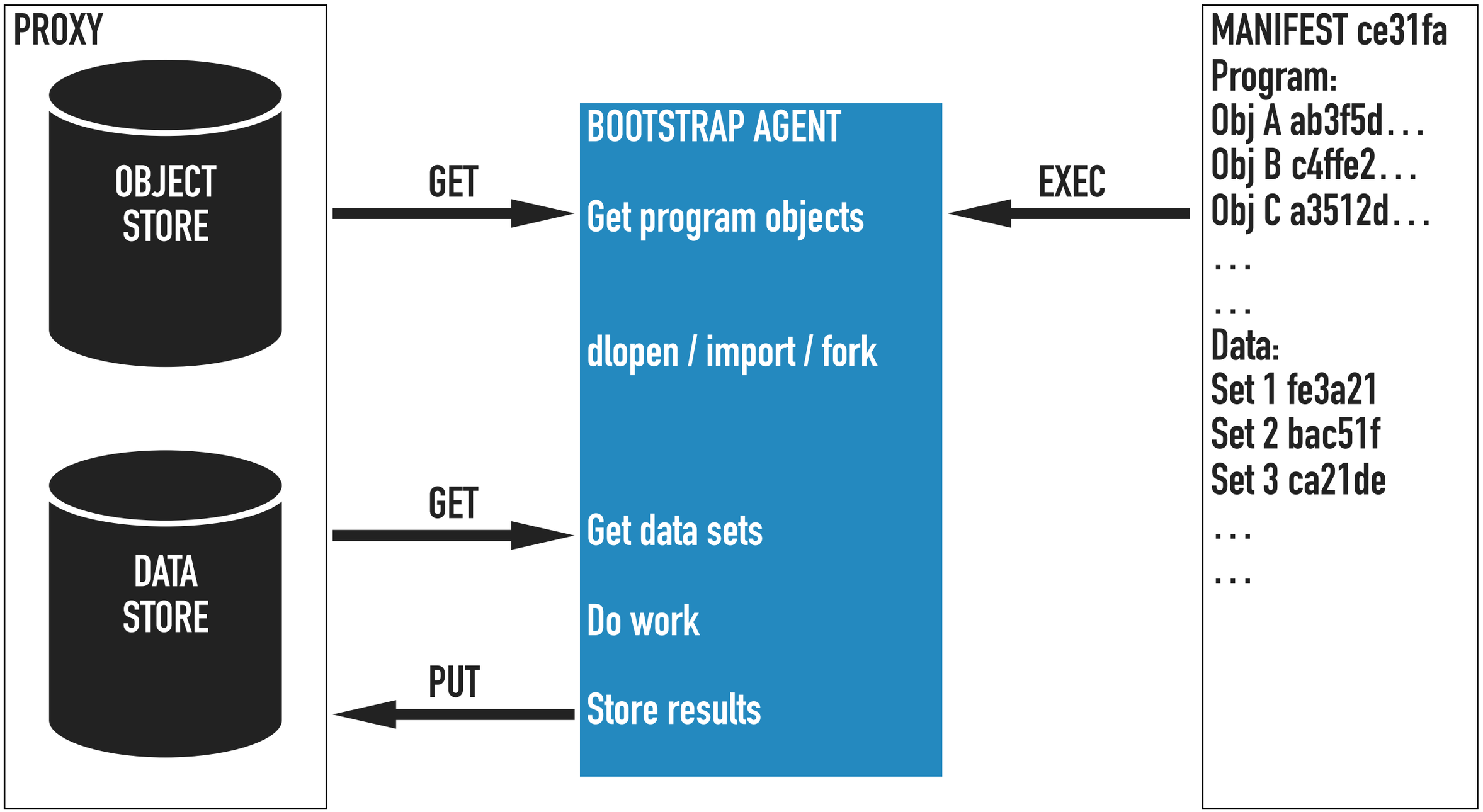
A PROPOSED SOLUTION

The idea is to have fewer *apparent* moving parts in our stacks, move more logic into the application (similar to unikernels):

- Make use of data stores instead of file systems (see ML - HDFS/Spark)
- Provision applications through a lightweight bootstrap agent - "business logic" is brought in through plugins, also retrieved from object stores
- Cryptographically secure manifests + Merkle trees can leverage content-addressed storage and on-demand transfers, while offering guarantees to site operators
- Consolidate the existing "past-the-gate" solutions in use today, into a robust, general proxy running at the HPC site

  Should make it easier to go to a new site, get in, do the work, get results out!

WORKFLOW


PROXY
OBJECT STORE
DATA STORE
GET
GET
PUT
BOOTSTRAP AGENT
Get program objects
dlopen / import / fork
Get data sets
Do work
Store results
EXEC
MANIFEST ce31fa
Program:
Obj A ab3f5d...
Obj B c4ffe2...
Obj C a3512d...
...
...
Data:
Set 1 fe3a21
Set 2 bac51f
Set 3 ca21de
...
...

SYNERGIES / BEYOND HPC

There may be benefits outside HPC, in other environments which don't adhere to POSIX:

- There is already a project proposal for using object stores instead of file systems for data

- Running in the browser sandbox where file systems are only emulated - the approach described here could be applied there as well

- Mobile devices have similar restricted environments

Finally, a related problem is that of distributing an arbitrary analysis workload - in the form of a closure in a ROOT session - on multiple remote nodes.

- We'd have to solve a similar problem: distributing program and data objects, serialisation/deserialisation of programs

CLOSING REMARKS

Improving the quality-of-life when using HPC resources would be bring considerable benefits to the HEP community, which is in need of compute resources.

Developing a simple and robust toolset for exploring opportunistic resources would be useful for approaching other environments (browser, mobile, IoT?)

The tools and workflows developed here could be used to solve other connected problems, like distributed computation in ROOT analysis sessions.