ROOT on IT Spark Clusters
An Update

E. Tejedor, D. Piparo, P. Mató (EP-SFT)
in collaboration with IT-DB, IT-ST

ROOT meeting
7/11/2016

What this is about

- Integration of **ROOT** and **Spark** for distributed execution
  - Trees seen as RDDs (collections of items)
  - Logical split in ranges of entries
  - Map-reduce chains to extract and aggregate information
- User-friendly interface
  - High-level API, hide complexity
  - Python / **C++** tasks
- Leverage existing CERN technologies
  - **EOS** for data
  - **CVMFS** for software
- Leverage CERN infrastructure
  - **IT Spark clusters** (+ IT container service?)

Challenges

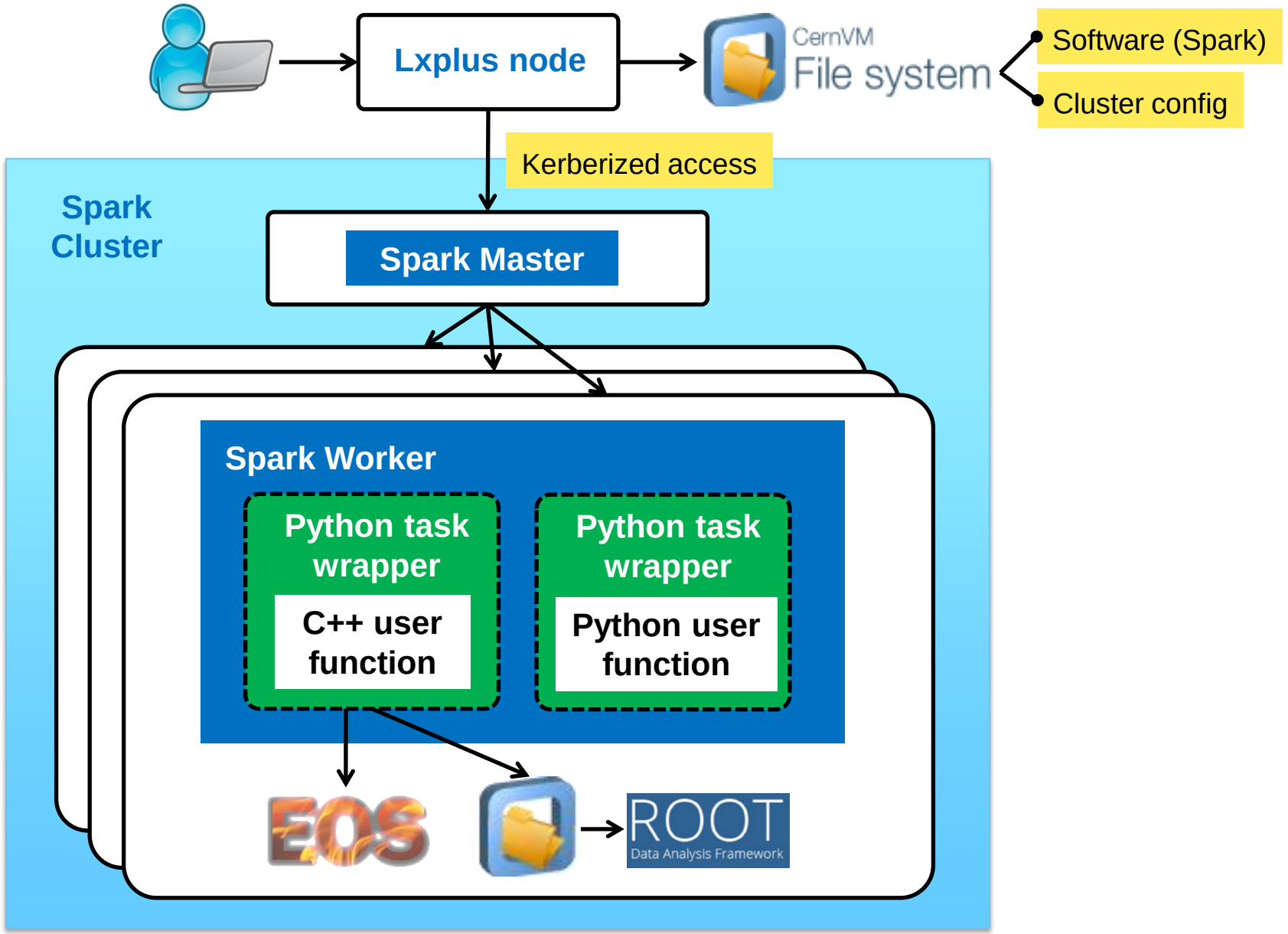
Software

- Offer a simple API to do map-reduce on a tree
  - User provides map and reduce functions
  - Map function operates on a sub-range of entries (receives a TTreeReader)
  - Full example here
- Add a layer between Spark and user code
  - Build logical ranges of entries
  - Make partitions independent of number of files
- Allow to run C++ (jitted) functions as mappers and reducers
  - Wrapper tasks receive the function code and jit it in the workers

Infrastructure

- Use the same software environment on client (lxplus) and server (IT Spark clusters)
  - Completely based on CVMFS
  - LCG view + cluster configuration
- Allow access of data on EOS
  - Preferably, authenticated access

In a picture


Lxplus node
CernVM
File system
Software (Spark)
Cluster config
Kerberized access
Spark Cluster
Spark Master
Spark Worker
Python task wrapper
C++ user function
Python task wrapper
Python user function
EOS
ROOT
Data Analysis Framework

What is missing

- Authenticated access to EOS
  - Work in progress with IT-DB, IT-ST
- Big use case for a big run
  - Work in progress with CMS
- C++ functions in libraries as Spark tasks
- Decide on user API
  - Expose functional programming to user or just high-level API?
  - ROOT file as a Spark DataFrame? Prototype by IT-DB