Plancton

an opportunistic computing project
based on Docker containers

Matteo Concas\footnote{matteo.concas@cern.ch}, Dario Berzano\footnote{dario.berzano@cern.ch}, Stefano Bagnasco\footnote{stefano.bagnasco@cern.ch}, Stefano Lusso\footnote{stefano.lusso@cern.ch}, Massimo Masera\footnote{massimo.masera@cern.ch}, Maximilian Puccio\footnote{maximilian.puccio@cern.ch}, Sara Vallero\footnote{sara.vallero@cern.ch}

\textsuperscript{1}Università degli Studi di Torino • \textsuperscript{2}CERN - Genève • \textsuperscript{3}INFN - Torino

A single tool for two use-cases

\begin{itemize}
\item \textbf{Plancton}: github.com/mconcas/plancton
\item \textbf{Docker}: docker.com
\item \textbf{Parrot (CCTools)}: ccl.cse.nd.edu/software/parrot
\item \textbf{HTCondor}: research.cs.wisc.edu/htcondor
\item \textbf{CVMFS}: cernvm.cern.ch/portal/filesystem
\item \textbf{Work Queue (CCTools)}: ccl.cse.nd.edu/software/manuals/workqueue.html
\item \textbf{AliEn}: alien.web.cern.ch
\item \textbf{AliEn-wq} (GitHub): github.com/alisw/alien-workqueue
\end{itemize}

The Plancton daemon

- Continuously spawn pilot containers
  - they execute a task then die
- Opportunistically use commodity resources
  - spawn containers when user does not use computer
- Just a container scheduler
  - full use-case implementation stays inside the container

Worker nodes as containers

- Minimal configuration which can be changed at runtime
- RAM, swap and CPU are capped (cgroups + cfs)
- Containers run inside VMs (CentOS 7); VM layer required by HLT experts
- Jobs are run in a single-shot mode → container dies when done, allows Plancton to launch a new one
- ALICE Grid middleware unmodified → using AliEn-WorkQueue

Results

- A lightweight scheduler for schedulers: completely independent, only takes care of container deployment
- Suitable for disposable tasks: input and output on external storage
- Plancton can be updated/restarted without affecting current running containers

Worker nodes as containers

- Volunteer computing: only Docker and Plancton required
- Jobs running on bare metal
- Opportunistic resource utilisation (configurable) → quickly given back to user when reclaimed
- Dedicated HTCondor submission node on a static resource