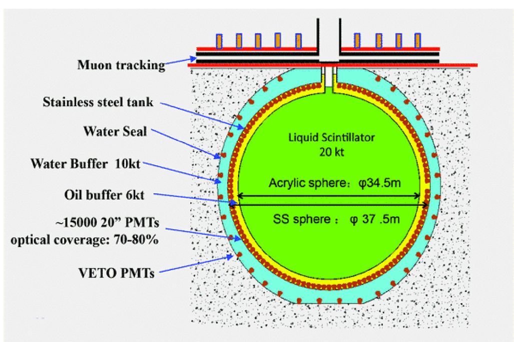
Multi-core supports in JUNO distributed computing

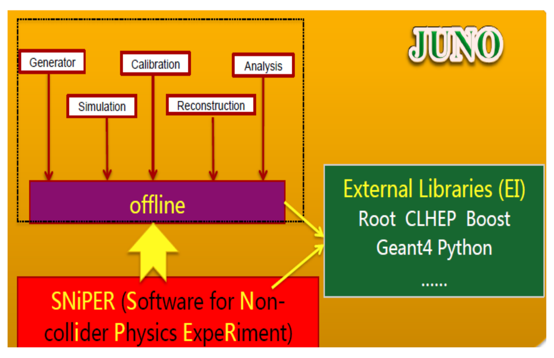
Xiaomei Zhang, Xianghu Zhao, Kang Li

Institute of High Energy Physics, Chinese Academy of Sciences, Beijing 100049, P. R. China

JUNO

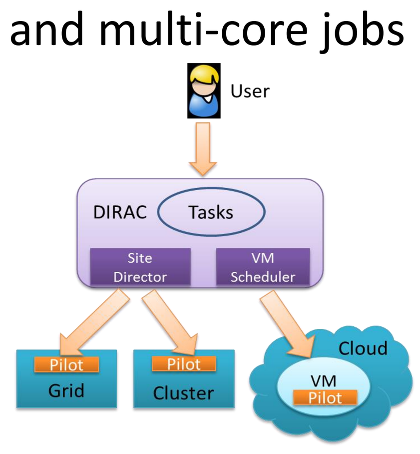
- Jiangmen Underground Neutrino Observatory (JUNO)
- A multi-purpose neutrino experiment designed to measure the neutrino mass hierarchy and mixing parameters
- Start to build in 2014, operational in 2020
- JUNO offline software system is being built on SNiPER
- With traditional single-core programming, JUNO data processing meet some limitations on memory and processing time
- To fasten JUNO data processing and fully use modern multi-core and many-core hardware, SNiPER is introducing parallelization based on TBB to enable multi-thread and multi-process simulation and reconstruction
- Event-level parallel processing is already in prototype phase

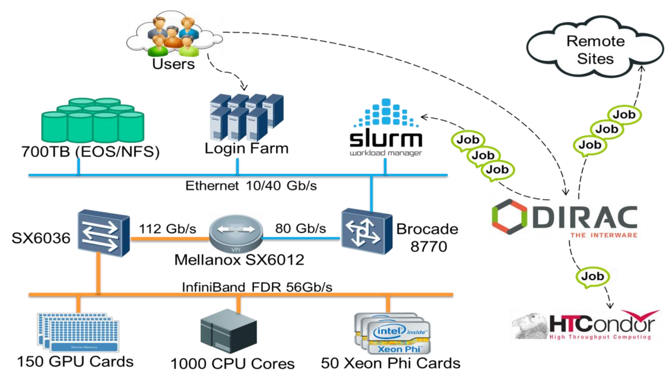




JUNO distributed computing

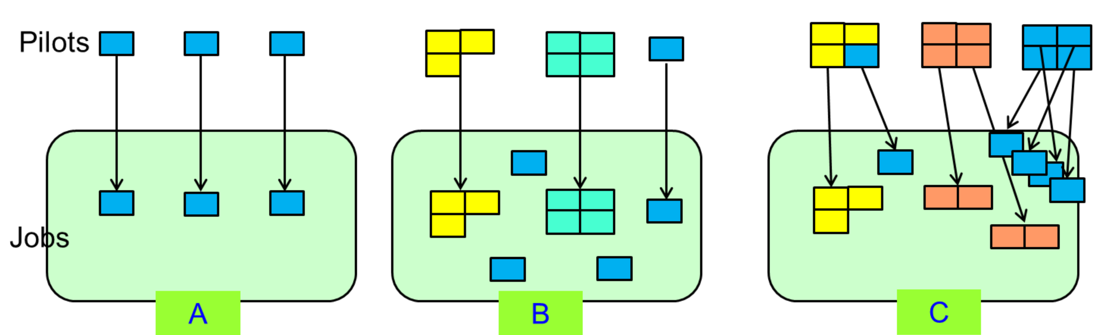
- JUNO is estimated to produce 2PB data/year for 10 years
- Prototype of JUNO Distributed computing has been built on DIRAC to organize heterogeneous and distributed resources for JUNO data processing
- The platform currently is able to integrate with Cluster, Grid and Cloud
- More and more HPC resources are emerging and ready to provide multi-core and many-core resources
- Single-core and multi-core JUNO jobs are expected to coexist in foreseeing future
- New workload scheduling strategy has to be introduced to support mixture of HPC and HTC resources , mixture of single-core and multi-core jobs





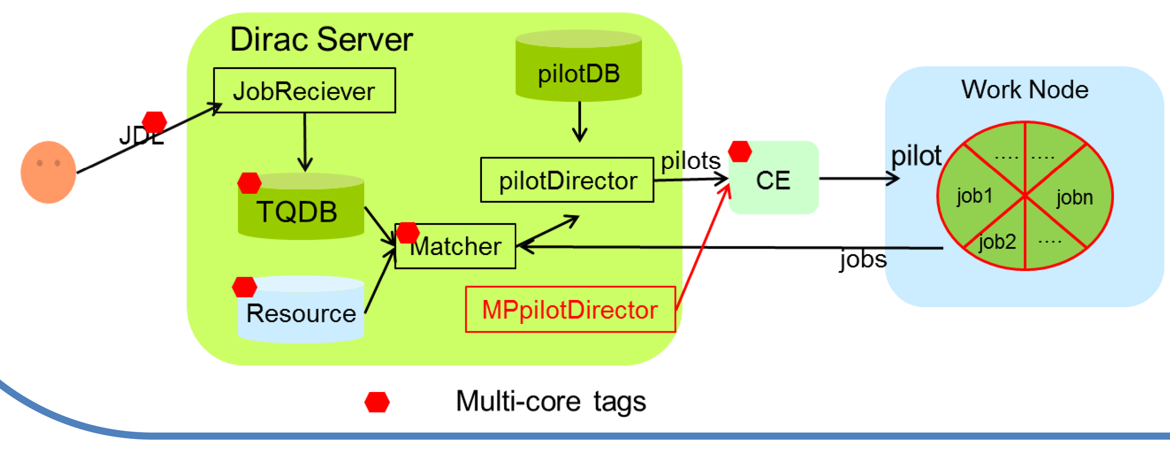
Multi-core designs

- DIRAC workload scheduling is based on pilots strategy
- In single-core mode, each pilot takes one slot from local batch system and pull one job from job pools , shown in plot (A)
- To add multi-core's support, two ways are considered:
- (B) Separated pilots created for single-core and multi-core jobs
-    M-core pilots pull M-core jobs, but Pilot "starving" happened when matching with mixture of different-core jobs
- (C) Shared partitionable pilots for both single-core and multi-core jobs
-    M-core Pilots pull N-core jobs (N<=M) until internal slots used up, Pilot s are standard-size, can be whole-node, 4-node, 8-node….



Multi-core Implementation

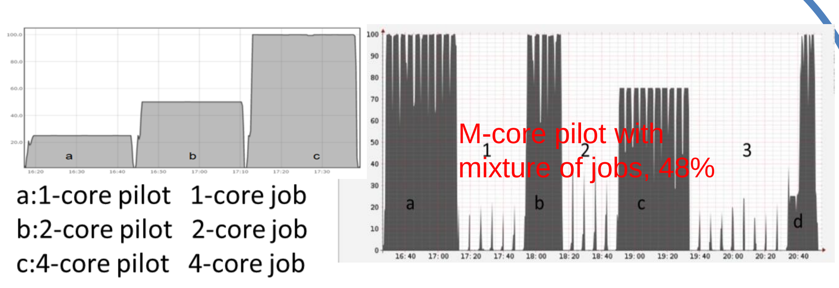
- Users are allowed to define jobs in JDL with multi-core tags
-    NumberOfProcessors: number of cores the jobs request
- Jobs are registered into Task Queue Database (TQDB) with these tags
- Sites registered in configuration service are added with multi-core tags to accept the jobs which can fit in the resource
-   NumberOfProcessors : number of cores in the pilots which can be accepted
-   RequiredTag: number of cores the jobs which can be pulled
- In the "separated pilots" way (B), new multi core pilot Director (MPPilotDirector) is introduced to submit M-core pilots for M-core jobs
- In the "shared pilot" way (C), PilotDirector decide and submit proper number of standard size pilots according to the number of matching jobs from the Matcher and available pilots from pilotDB.
- New type of pilot is introduced in the "shared pilot" way, so that one pilot can accept more than one job synchronously and do the rescheduling internally
- Matcher matches jobs and resources based on multi-core tags to give pilots the proper jobs from task queue

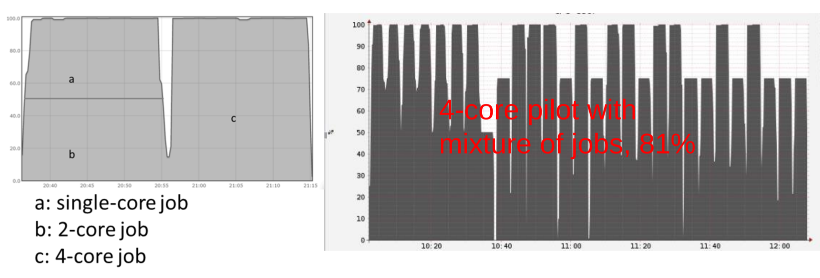


- In Cluster and Grid case, multi-core tags pass to the site computing element(CE) to submit multi-core jobs
- In Cloud case, multi-core VMs has to be created to hold multi-core pilots

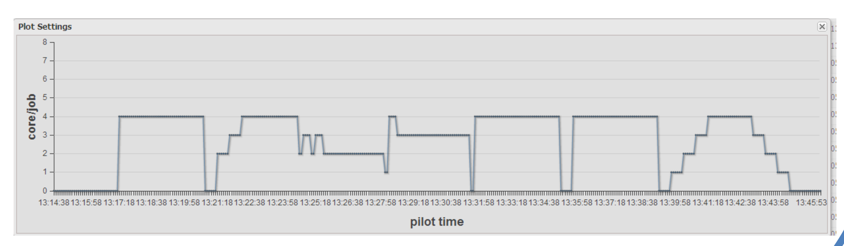
Tests and efficiency study

- Basic tests have been done on SLURM and HTCondor with whole-node, mixture of multi-core and single-core jobs
- CPU utilization rate has been measured and compared among three types (A)(B)(C) mentioned in "Multi-core designs" section
- With mixture of jobs, the utilization rate of (C) is higher than that of (B), but still lower than the original single-job case (A) which are 92%
- In (B), the efficiency is low, mostly because some pilots are idle without proper jobs to pull
- In (C), some pilots did get jobs, but not fulfill the whole node
- In (A) (B) (C), small overhead come from the pilot itself who need time for its life cycle
- Efficiency test plots are taken from ganglia





- Multi-core Monitoring has been added to the DIRAC job and pilot monitoring
- The following plot shows the life cycle of a multi-core pilot



Summary and Future Improvements

- The prototype of multi-core supports in JUNO distributed computing platform has been set up
- Basic tests has shown that it is working with multi-core jobs
- With efficiency tests, the way with shared partitionable pilots is accepted for further optimizations
- More tests will be done with real JUNO multi-core jobs
- With mixture of different-core jobs, it is much more difficult to decide the proper number of pilots to submit for jobs in queues than single-core case. In Cloud case, unnecessary VMs will be created due to that.
- More ideas come up to cope with efficiency problems met. One is sizeable job, in which the size of jobs are expected to be adjustable to available cores in pilots when they arrive in WNs. The other is to allow jobs running across WNs to fully reorganize free cores across WNs.

中國科學院高能物理研究所
Institute of High Energy Physics
Chinese Academy of Sciences