ATCF WG Activity Report

Andria Arisal, Brij Kishor Jashal, Masahiko Saito, Puneet Kumar Patel,
Sang-Un Ahn, Syam Budi Iryanto, Vikas Singhal

1 November 2023 @ ATCF7

ATCF WG

Building a Support Model

- A working group was proposed to discuss challenges and proposed actions as well as directions for launching the Asian Big Science Cloud (ABC), a strong Asian site collaboration.
  - ABC - a ATCF driven version of collaboration (envisaging a firm foundation) to support Asian computing sites involving in LHC/non-LHC experiments
- Challenges: (operators) lack of knowledges, expertises, training or tracking opportunities on technologies / (sites) difficulties on securing resources, manpower, budget and infrastructure ...
- Proposed actions:
  - supporting participation to various tutorials/workshops, inviting experts, organizing WG for technology tracking, training, commissioning, developing and jointly taking part in activities in WLCG, and so on
  - applying to inter-country cooperation programs (e.g. ID-RoK), hardware donation, etc.

Mandate of Working Group

- Discuss in detail the proposed actions (with any further ideas and suggestions) to form a support model in Asia
- Share the list of actions for the Asian support model within ATCF and finalize them
- Propose joint-sessions (a few times) in related conferences or workshops, e.g. ISGC, APAN, CHEP, WLCG workshop, etc. to share and discuss a draft of support model for completion
- Apply to S&T cooperation programs available in ASEAN, ASEAN+3, EAS, etc. for knowledge transfer, HR training, IT collaboration and so on.
- WG members: Andria and Syam (BRIN), Masahiko (ICEPP), Brij and Puneet (TIFR), Vikas (VECC) and Sang-Un (KISTI)

WG Meetings

- Three times of online meetings (about an hour)
  - 27 Feb 2023
  - 27 Jun 2023
  - 5 Sep 2023
- Thanks to WG members for sparing their time

Support Model

Organized actions dealing with challenges

- Human resource development: technical skill training support, information exchange, sharing expertise, knowledge transfer
- Service infrastructure / environment development: cloud-like/-native interoperable service provisioning, related technology development (or implementation)
- Liaison with inter-country S&T cooperation programs
- WLCG (or similar scientific computing collaboration that Asian sites participate) communication channel

Strategy

- Step 1: help each other (current)
  - Information exchange, sharing expertise, knowledge transfer (might use inter-country exchange programs)
  - Inviting experts from outside
- Step 2: pursue common technology
  - Batch system, storage management (ex: htcondor, eos for the moment...) and infrastructure resource management (like Kubernetes)
  - Easy to operate cooperatively and integrate resources (hurdles might be internal/campus restrictions)
- Step 3: develop common tool
  - Service / resource provisioning automation (e.g. Ansible playbook, Openshift/Kubernetes, etc.)
- Step 4: apply for funds (APTCF) to sustain the model

Limitation

- Lack of manpower, expertise → stick to step 1 & 2
- Tracking technologies + extra manpower could help achieve step 3

Strategy 1 & 2

"Help each other and pursue common technology"

- Important to keep in touch through a place like ATCF
- "How to setup a grid site" kind of tutorial
  - "SIMPLE" is a good example but it seems to be outdated (no longer activities on Github since 2020)
    - https://simple-framework.github.io/
  - "New Grid Site Guideline" drafted
- A repository for sharing common tools and useful tips via CERN GitLab
  - https://gitlab.cern.ch/ppatel/T2_IN_TIFR/-/tree/master/HTCondor
  - https://gitlab.cern.ch/atcf
- Tech search for tools or development projects that are useful for site operations

Strategy 4

- Korean-ASEAN S&T Cooperation Program
  - Indonesia-ROK program of HR training (or exchange) applied once
  - ASEAN-Plus-Three or Inter-asian-country related S&T programs appears to be suspended
- Other countries cases?

Sustained Model ?

- Supporting such (before-mentioned) activities, personnel
- Hiring (or partially supporting) manpower for certain activities (e.g. development/implementation)

  ➡ Step 4 should be required in this context

  ➡ KISTI could continue to support ATCF as well as individual activities until realizing step 4

New Grid Site Guideline

Draft v0.1

ATCF WG

Intro

- Structured in 3 parts
  - Local Resource Management
  - Grid Services
  - Sustainability

Essential Processes

Some budgets + manpower assumed

- Securing Space - electricity, cooling, access control
- Hardware Purchase - server, storage, router/switch, firewall, console, rack, cabling, containment, sensors
- System Installation - OS provisioning, hardware discovery, configuration management, identity management, security, monitoring
- Enabling Services - package installation, service configuration/integration/validation

Essential Services for a Grid site

Common Components for VOs

- Local Resources
  - Batch system, Storage, Networking
- Grid Services
  - WMS, Computing Entity, Storage Element, Information Service, Accounting

Example

KISTI ALICE T1 Structure Overview

Simulation
Organised Analysis
Reconstruction
User Analysis

Job Queue

WLCG CRIC

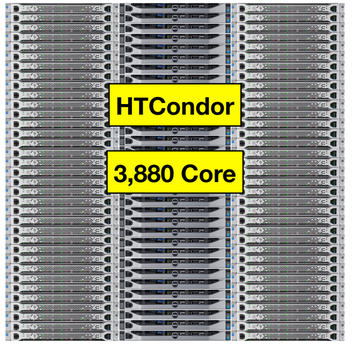
publish

Accounting

APEL

statistics

Job
Data
CERN



Job Broker

allocation

VO-Box

submission
query

HTCondor-CE
CE

scheduling

Batch Cluster

HTCondor

collect
AuthN/AuthZ

EGI.eu

BDII/Argus

read only

data i/o

80 PB
Erasure-Coded storage

Detector

ALICE O2
80PB/yr

O2 Buffer

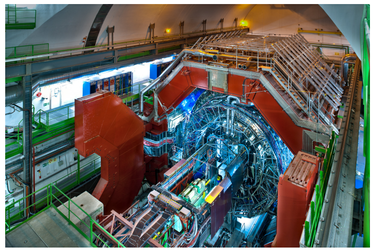
real-time data transfer (> 25Gb/s required)

Archival Storage

raw

Disk Storage

processed



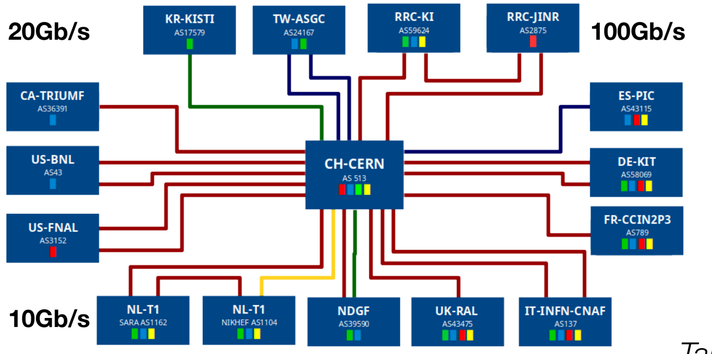
archive

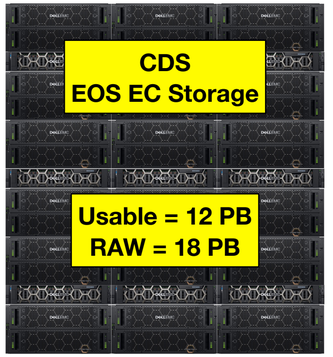
~ 1EB

T0

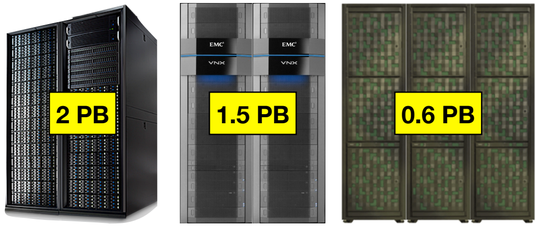
LHCOPN

KISTI-CERN 20Gbps





Tape Library decommissioned Nov. 2021



Tape Buffer

Local Resource Management (1/3)

Batch System

- HTCondor, SLURM, PBS, ...
  - Quick start guide provided from those of developers would not be enough
  - Advanced administration is required - efficient resource sharing, quota/ access control, multi-core submission, ...
  - Sharing sites' configuration or users community channel can help
    - GitHub repo, HTCondor users mailing, HTCondor or HEPiX workshop

Local Resource Management (2/3)

Storage

- Most cost-demanding part of site operation (WLCG Survey, 2015)
- Different storage types available in market: DAS, NAS, SAN, ...
  - Architectures and solutions are mostly vendor-driven
    - Ceph/Lustre/GPFS Appliances
  - Open-source based storage management is possible
    - Ceph, Lustre, EOS, xrootd
    - highly-skilled admins are mandatory
- Archiving storage is an another story...

Local Resource Management (3/3)

Network

- Critical part of running a local resource as well as a global site
- (Linux) system administrator ≠ network administrator (vice versa)
  - Should rely on campus network administrator or external providers
    - Connecting to LHCONE/LHCOPN
  - Should make a contract with storage or network vendor for enabling and validating connectivities (not just buying devices)
    - E.g. storage volumes (or devices) should be seen at OS level via ethernet (NAS) or fibre-channel (SAN) or SAS (DAS), etc.
    - Storage and/or network engineers should work together

Grid Services (1/5)

WMS - Workload Management Service

- VO-Specific: JAliEn (ALICE), GlideInFactory (CMS??), PanDA (ATLAS??), Dirac (LHCb), ...
- Should communicate with Computing Element (CE) at a site
- (ALICE) VO-Box runs JAliEn CE daemon to accept ALICE jobs and HTCondor JobRouter to submit (route) jobs to local HTCondor-CEs
  - https://jalien.docs.cern.ch/site/vobox_htc_arc/
  - CERN-IT ALICE Liaison (Maarten Litmaath)
- (ATLAS) (ICEPP) Direct submission to ARC CE, then jobs are routed to HTCondor clusters
- (CMS?)
- (LHCb?)
- (Belle II and others?)

Grid Services (2/5)

Computing Entity

- CE mostly authenticates and authorizes job pilots from VOs and routes to LRMS so that the job pilots occupy compute resources and download payloads
  - ARGUS recommended
- HTCondor-CE & ARC-CE install guides are well documented including troubleshoots
  - HTCondor-CE: https://htcondor.com/htcondor-ce/
  - ARC-CE: https://www.nordugrid.org/arc/arc6/
  - Additional configurations are required for Information Service and Accounting
    - For HTCondor-CE,
      - BDII: https://htcondor.com/htcondor-ce/v6/configuration/optional-configuration/#enabling-bdii-integration
      - APEL: https://htcondor.com/htcondor-ce/v6/configuration/optional-configuration/#uploading-accounting-records-to-apel
    - For ARC-CE,
      - BDII: https://www.nordugrid.org/arc/arc6/admins/arc6_install_guide.html#information-system
      - APEL: https://www.nordugrid.org/arc/arc6/admins/details/accounting-ng.html#accounting

Grid Services (3/5)

Storage Element

- Access endpoint to local storage
  - WLCG prefers http(s) and xrootd protocols
- Most popular storage management systems: dCache, EOS, XRootD, ...
  - Quite well documented in most cases
  - Active developer-driven user community forums
  - Annual dedicated workshop for each of solutions
  - Other sites' experiences and round table of developers and users are extremely useful

Grid Services (4/5)
BDII & ARGUS

- Simple configuration to deploy Site-BDII based on YAIM (node_type = "BDII_site", package = "emi-bdii-site") *needed to check
  - GIP provider references /etc/bdii/gip/*.conf
    - Including SITE_NAME, SITE_BDII_HOST, LDAP URLs
- ARGUS installation and configuration are well documented
  - Quick Start Guide: https://argus-documentation.readthedocs.io/en/latest/quick_start/index.html
  - Using YAIM: https://argus-documentation.readthedocs.io/en/latest/misc/configuration_with_yaim.html
  - Manual configuration: https://argus-documentation.readthedocs.io/en/latest/misc/manual_configuration.html#argus-manual-configuration

Grid Services (5/5)
Accounting

- APEL - WLCG oriented accounting system
  - Sites should install APEL client to publish their accounting data parsed from local clusters via APEL parsers
  - Basically CE or Server of a local cluster runs APEL parser to extract User(VO), Wallclocktime, CPUclocktime, start/end time, number of CPUs etc. from the local cluster history daily and to send the data to APEL client database
  - APEL client collects JobRecords sent by APEL parsers and sends summaries to APEL central server
  - Basic DB administration skill requires in case of troubleshoot on APEL client DB or one should contact APEL support via GGUS ticket
  - APEL for HTCondor-CE: https://twiki.cern.ch/twiki/bin/viewauth/LCG/HtCondorCeAccounting

Sustainability of Grid Site Operations (1/2)

Automation is the key

- Manifests of puppet modules, ansible playbooks, helm charts that CERN publishes and shares created for their own operations are very useful and helpful to glimpse how services can be configured
  - EX1) Learned how to effectively upgrade EOS from v4 to v5 based on EOS documentation which is basically the snippets of EOS puppet module created by CERN EOS operation team (not by EOS developer team)
  - EX2) Helm charts are publicly available to develop ScienceBox upon Kubernetes cluster by CERN
  - EX3) Several puppet modules for HTCondor-CE and/or HTCondor can be found on the Internet

Sustainability of Grid Site Operations (2/2)

How to automate??

- Basic principle for orchestrating multiple nodes
  - (OLD) Bash or python scripts with loop via SSH → (NEW) Ansible (essentially the same)
    - Choose your preferences other than Ansible, e.g. Puppet, a bunch of bash scripts, or python modules
- A complete recipe described for a service (e.g. Ansible playbook)
  - Carefully crafted one helps recover or re-install the service whenever needed
  - Mind configuration file, data (e.g. databases, history files) and their location so that they are persistent while recovery or re-installation
- Bare metal, VM, container, ...
  - Fancier orchestration, VM clusters (oVirt or OpenStack) or Kubernetes (or OpenShift) can be possible with additional efforts and resources however a small site cannot easily afford to maintain
  - Still feasible to mimic similar level of orchestration with the combination of home-grown scripts, playbook, etc.

Outro

- KISTI plans to migrate legacy infrastructure to VM and/or Container based operations
  - Most of all core Grid services are running on VMs currently upon oVirt cluster
  - We will re-write service deployment codes in Ansible and container-based
  - As long as any of those service deployment codes are ready then we will publish them on ATCF GitHub with readme documentation
- Need your feedback & contributions