Computing Strategy for Hyper-Kamiokande

Francesca Di Lodovico

27-28 January 2014
Fourth Hyper-Kamiokande Open Meeting
Kavli IPMU, Kashiwa

Work done in collaboration with Grid and iRODS computing experts in the UK and Japan

erc Queen Mary University of London

Outline

- Agreement w/ Super-Kamiokande and T2K
  - Description of the Agreement
  - Current Status
- Release
  - First Hyper-K release
- Production
  - Overview
  - Simulation
  - Reconstruction
  - Data Retrieval
- Web Pages (documentation)
  - Wiki
- Computing Model Strategy

Y.Suzuki@Super-K meeting on May 10th, 2013

Report

- Request from HK working group for an use of SK softaware and SK data in the HK software delopement work.
  - They asked to make a similar agreement between SK and T2K

Answer to the request

1) The SK Executive Committee feels that Hyper-Kamiokande is a natural extension of Super-Kamiokande, therefore, we should cooperate as much as possible.

2) We think that a HK working group will develop their own simulation code based on the 'real' HK geometry and other parameters since the HK detector size and shape are quite different from SK. As HK develops its own simulation code, then HK software becomes less dependent on SK software. We believe that HK should start their code development now.

3) Certainly, we agree that HK may adopt concepts of the SK software and if necessary to copy SK codes into HK codes freely. Once done, HK may modify the codes as they wish without reporting all the modifications back to SK. We would like to be informed of any modifications that could also improve the SK code.

4) SK software should not be used for any other purpose than for HK simulation and analysis software development.

5) You mentioned the T2K-SK agreement as an example, but the situation of T2K is quite different from HK. T2K uses SK detector as a far detector, therefore, T2K definitely needs SK software and SK data, which is the reason we have made such an agreement. HK will not use the SK detector, so we do not need the same kind of agreement, as you have proposed.

6) SK will not form a group to have responsibility for maintaining analysis software of the SK detector for the HK group.

7) SK does not see any reason that the HK working group needs to use SK data for the HK software development. SK believes that HK needs only the MC generator, detector simulator and data analysis codes, which we are willing to provide, but not data.

Hyper-K code vs Super-K vs T2K

Vector Generation

Beam/ND: modified T2K code
Atmospherics: Honda flux
Nucleon Decays: SK code (vec_gen_b8)
Solar: SK code (ndecay_vc)

Simulation

WCSim

Expected to be supported for Hyper-K soon
- NEUT
- GENIE

- Most of the code is open source or will be in the near future.
- Whilst writing the TDR, we can write new Hyper-K code for atmospherics and solar ν and proton decays.
- I expect the beam and ND280/INGRID code to be shared with T2K in the future, although specific changes will be made for Hyper-K

Reconstruction

fiTQun, BONSAI

Analysis

Atmospherics ν: OSC++
Beam ν: PROB++
Solar ν: SOLFIT, spallation cuts and related libraries

Hyper-K Software Release

- Use git (https://github.com/) as code-repository
  - Git is a modern code management system, being used by several particle physics experiments (eg. SNO+)
  - Held at QMUL. Aim to mirror to other sites.
  - To get an account email your public key to both:
    - Alex Owen r.a.owen@qmul.ac.uk
    - Francesca Di Lodovico f.di.lodovico@qmul.ac.uk

- Add the relevant packages to the repository.
  - All open-source packages are treated as “third party” packages (no developement in git), similarly to what currently happens in T2K.
  - Production scripts for HK are currently added in git.

First Hyper-K Release

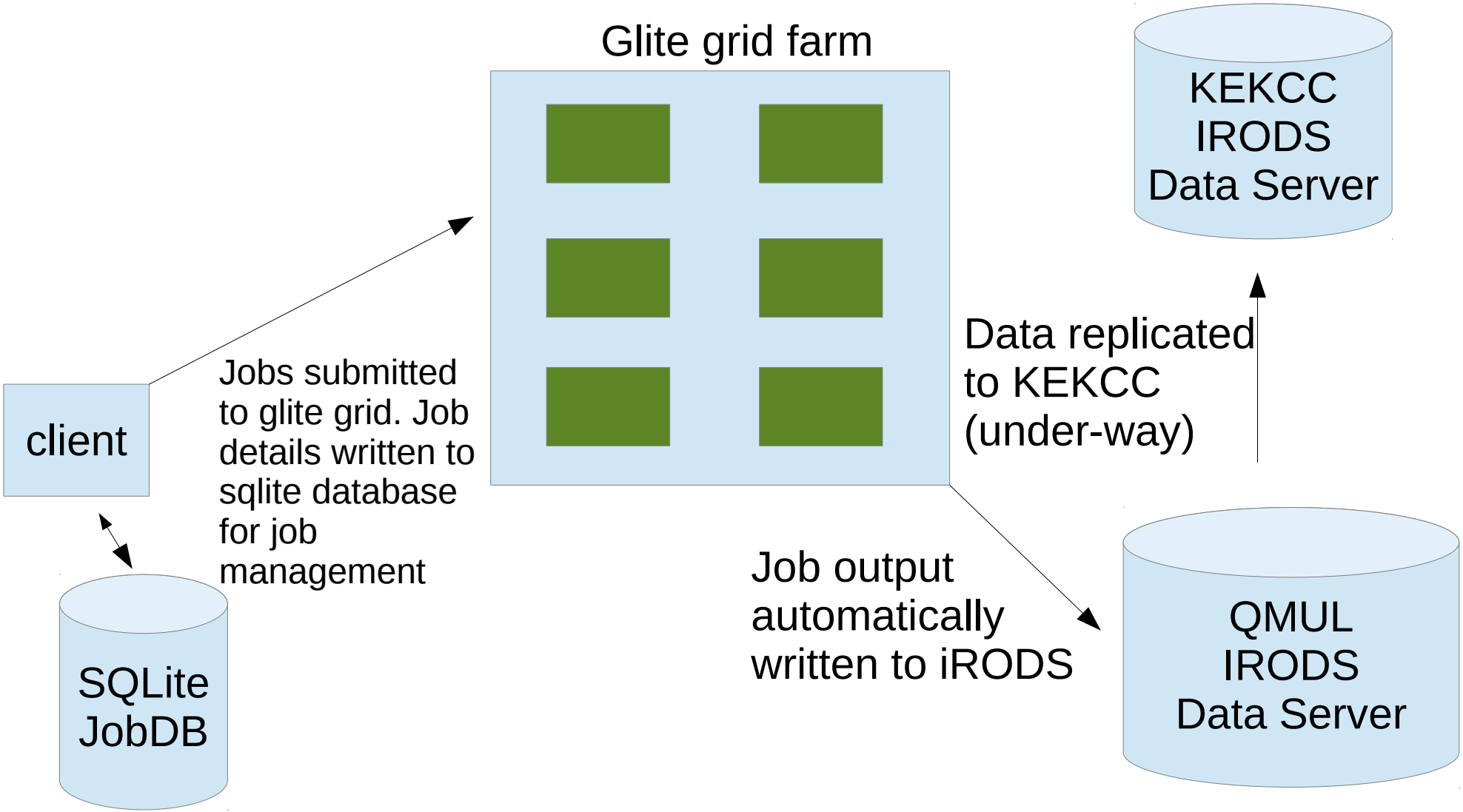
- Created first Hyper-K software release.
- Five external open source packages:

  ROOT CLHEP GEANT4 WCSim fitTQun

- BONSAI will be added when interfaced with WCSim.
- Six repositories on Git so far:
  - hk-clhep for CLHEP
  - hk-eventdisplay for event display
  - hk-fitqun for fiTQun
  - hk-geant4 for GEANT4
  - hk-hyperk for general scripts (management package)
  - hk-root for ROOT
  - hk-utils for production scripts
  - hk-WCSim for WCSim

- Plan to add a package for analysis work.
- Instructions to download the release are on the Hyper-K Wiki https://wiki.hyperk.org/ (see later)

Hyper-K Production

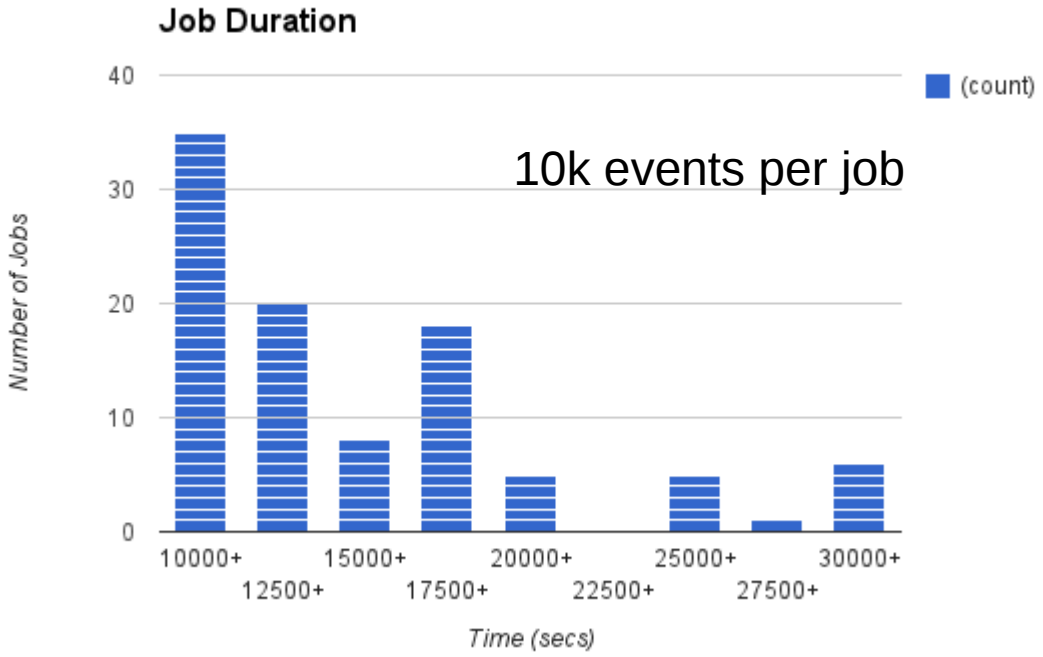


Hyper-K Production

- Release deployed to CVFMS (distributed read-only file system – new!). Binaries read from there
  - Code put onto one CVFMS is visible typically within 2 hours
  - Initially saw problems with filesystem not visible, more stable now
  - Currently releases have to be uploaded via web interface (very slow). Will try recent command-line api
- Jobs managed through scripts that interface to local sqlite database
  - Database schema has hooks for cloud jobs (not implemented yet)
  - Schema keeps track of job status
  - In principle allows running production at different centres (although need to use run-ranges to avoid clashing jobs)

Hyper-K Simulation

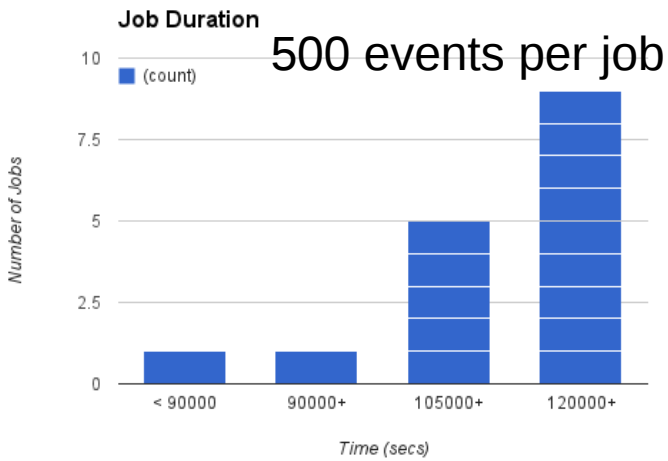
- Data files read from CVFMS along with executable
- Output written to iRODS
  - Log files written to 'success' collection or 'failed' collection depending on return code
- Currently run 900K nu-mode beam events (could have run more, stopped to focus on reconstruction):
  - 90 runs (10K per run). 5 jobs failed due to expired proxy or filesystem not visible (resubmit was successful)



Job duration exhibit exponential decay – long tail due to other processes running on grid Nodes, filesystem activity etc. Majority of jobs take <18s/evt.

Hyper-K Reconstruction

- Read simulated data from iRODS to compute node, process and write out to iRODS
- Quite a few problems running jobs on Grid:
  - Needed long running time. Max limit of 2 days/job on grid nodes at QMUL
  - Proxy certificate valid for 12hrs by default, but can be extended
- Split jobs:
  - 1 simu job = 20 reco jobs (500 events/job)
- Split jobs into smaller chunks (100evt/run) in the last days. Problem is many small root files, but faster throughput.



On limited stats most jobs take ~4min/evt. Mike suggest turning off Second Pi0 fit which should shave off 30secs/evt.

<10k recoed nu-mode events available at the moment.

Hyper-K Data Storage

- Data available in iRODS (no Grid certificate needed):
  - /QMULZone2/home/hyperk/production/simu/V00-00-01
  - /QMULZone2/home/hyperk/production/reco/V00-00-01
- Data is being replicated to KEKCC (under-way)
- Production version maps to frozen versions of WCSim and fiTQun. File names have form
  - neutvect-<run#>-<sim#>.root
  - fitqun-<sim#>-<rec#>.root
  - In this case 1 run = 1 sim file
  - Log files written to 'success', 'failed' collections.

Next Steps for Production

- Changes to reco schema needed to account for merging process
- Implement data replication
- Setup web page that will automatically be populated with current production
- Investigate deployment on other Grids

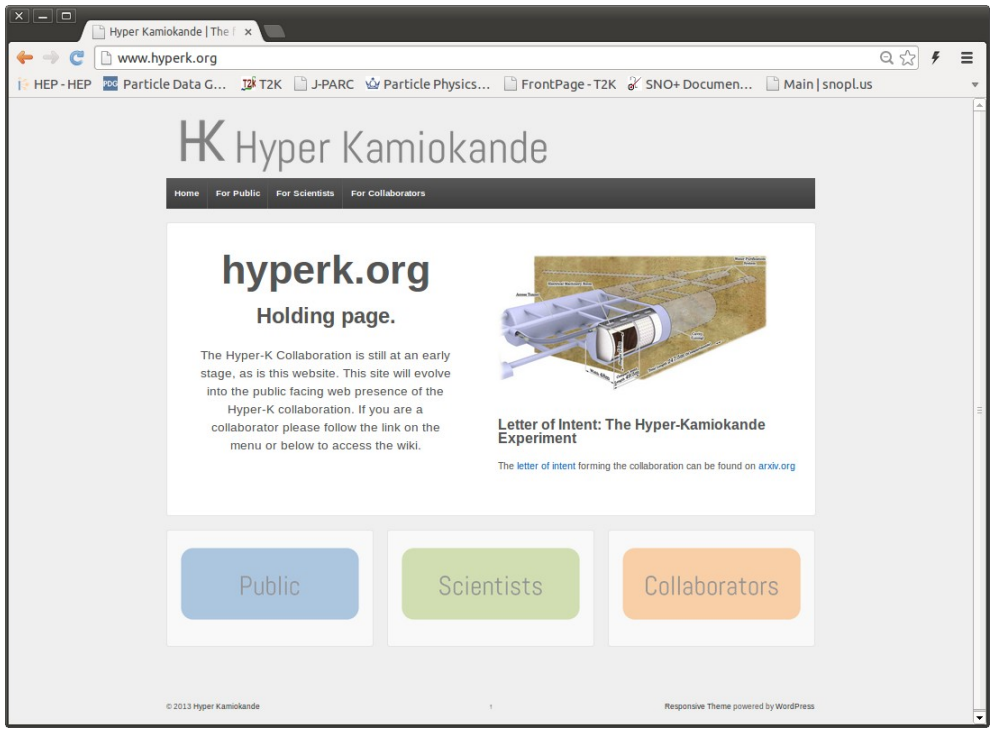
Web Site

As a by-product of the Grid VO (Virtual Organization), we have a domain hyperk.org.

Two web sites:

Public web site: http://www.hyperk.org/

Working wiki web site: http://wiki.hyperk.org



Added basic info to the public web site.

We will need to buy more domains for Hyper-K

Hyper-K Wiki

Working wiki web site: http://wiki.hyperk.org

The Wiki is meant to be the working web site for the experiment.

Currently contains all the needed software documentation:

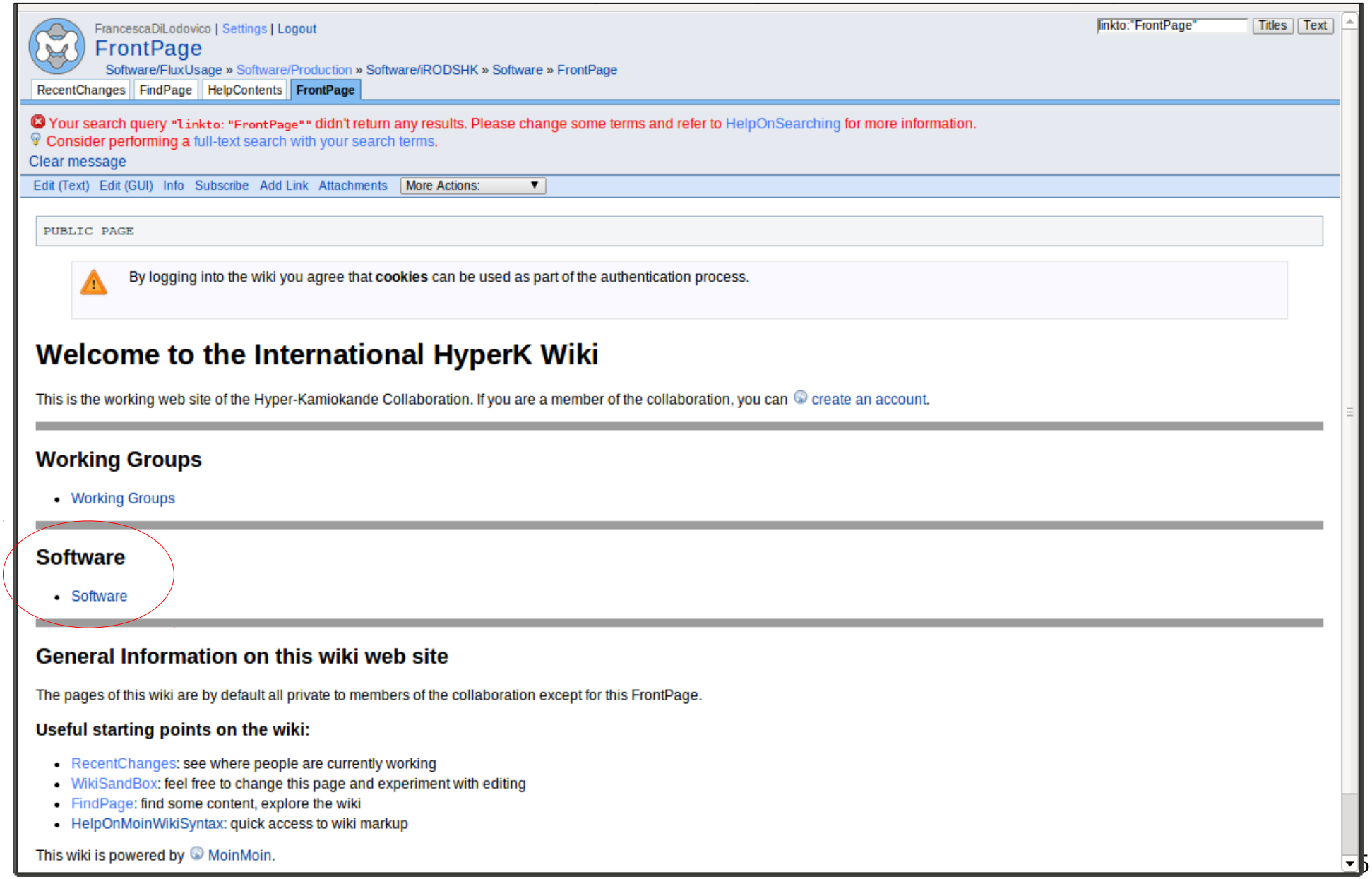
- how to install a release

- how to retrieve the production files

If you are a T2K member and use the same email address as for t2k.org you will be automatically registered, otherwise I will get an email requesting to approve the account.

For questions on the regustration, email Alex Owen (r.a.owen@qmul.ac.uk) and myself (f.di.lodovico@qmul.ac.uk)

International wiki https://wiki.hyperk.org



FrancescaDiLodovico | Settings | Logout

FrontPage

Software/FluxUsage » Software/Production » Software/iRODSHK » Software » FrontPage

linkto:"FrontPage" Titles Text

RecentChanges | FindPage | HelpContents | FrontPage

Your search query "linkto: "FrontPage"" didn't return any results. Please change some terms and refer to HelpOnSearching for more information.

Consider performing a full-text search with your search terms.

Clear message

Edit (Text) Edit (GUI) Info Subscribe Add Link Attachments More Actions:

PUBLIC PAGE

By logging into the wiki you agree that **cookies** can be used as part of the authentication process.

Welcome to the International HyperK Wiki

This is the working web site of the Hyper-Kamiokande Collaboration. If you are a member of the collaboration, you can create an account.

Working Groups

- Working Groups

Software

- Software

General Information on this wiki web site

The pages of this wiki are by default all private to members of the collaboration except for this FrontPage.

Useful starting points on the wiki:

- RecentChanges: see where people are currently working
- WikiSandBox: feel free to change this page and experiment with editing
- FindPage: find some content, explore the wiki
- HelpOnMoinWikiSyntax: quick access to wiki markup

This wiki is powered by MoinMoin.

Computing Model

Working on two major timelines:

Short Term future:

- Need to have a system automated and up and running for the current simulation production: use the Grid (VO: hyperk.org) - ongoing

Middle/Longer Term Future:

- Develop Computing Model for Hyper-Kamiokande
- Current proposal presented @ the Second Open HK Meeting: http://indico.ipmu.jp/indico/contributionDisplay.py?contribId=5&confId=10
- Main aspects:
  - Cloud
  - Virtualization
  - Digital Preservation
- Starting to test the Cloud.
- It will be developed in the next years.

Summary

- Developed a fully comprehensive system for
  - Release code management
  - Processing on the Grid
  - Data Storage
  - Documentation
- The system is complete, but being fine-tuned.
- About production:
  - Reconstruction requires more tuning
  - Test more streamlined and automated deployment of production releases
  - Can provide soon all 900k simulated data and <10k reco'ed nu-mode events.
  - Can provide early February all needed simulated data.

Additional Slides

Distributed Processing

- Generally want to keep jobs within few hours
  - Limit exposure to infrastructure 'glitches' (network problems, disk problems etc)
- For very long jobs ideal to have checkpointing
  - If job fails 90% into processing don't lose everything
- For reco is it possible to have a number of detailed passes:
  - Find 'obvious' rings output results
  - Read in results in next pass and look for more combinations output results
  - iterate

Data Replication

- Currently setting up iRODS system at KEKCC for HK
  - But, total available space will be 1TB until HK approved expt at KEK
  - Can cache most recent data there for people at KEK to use and have 'older' data only available at QMUL
  - Can setup another iRODS system and replicate data to there as well to provide larger pool of storage.

How to get the Data

- There are currently 90K simulation events in iRODS. The reconstruction is processing and will match.
- Make sure you have the iRODS icommands in your path.
- Make sure you have an .irodsEnv file for the HYPERK_Reader account
- Log onto the QMUL iRODS system:
  - iinit
  - Ask me for the account details
- To dowload simulation data:
  - iget /QMULZone2/home/hyperk/production/simu/V00-00-01/neutvect-<run>-<sim>.root
- To download simulation data:
  - iget /QMULZone2/home/hyperk/production/reco/V00-00-01/fitqun-<sim>-<rec>.root

Short-term needs: MC Production