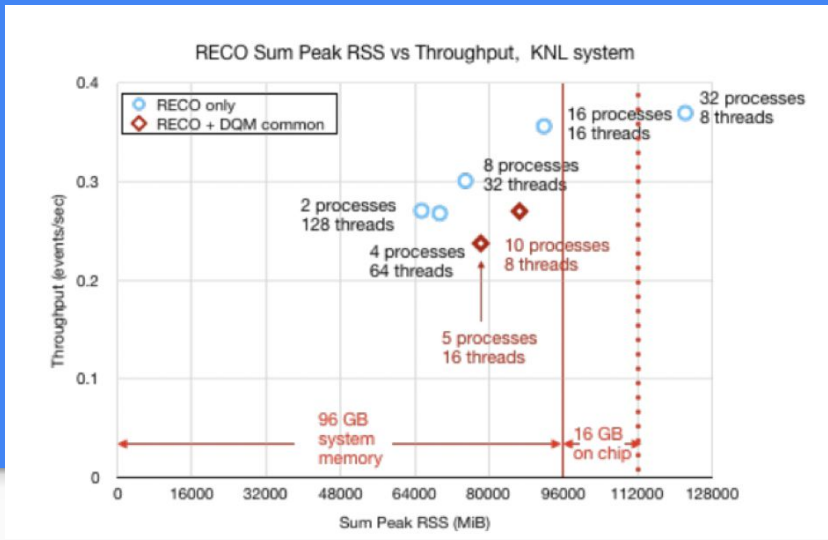
CMS - Prace meeting

Dirk Hufnagel (FNAL) for the CMS Collaboration

Introduction

- CMS, like other LHC experiments, has its “own” grid computing infrastructure
- This computing model has worked well for us so far
- For HL-LHC, extrapolations of this model show a significant resource deficit (assuming no significant funding increases)
- We received direction from our funding agencies (different ones from different countries) to look at HPC. HL-LHC will roughly happen at the same time as the push for Exascale in HPC, we could benefit of this.
- This is the motivation behind the efforts in recent years to use HPC

CMS Workflows



Special setups for KNL:
Reconstruction down to less than 1 GB/thread with CPU penalty

- CMS workflows have a quite large spectrum of CPU. Memory, I/O requests
  - **Simulation**: mostly CPU intensive, low I/O (except for digitization step which is comparable to reco), with multithreading can fit down to 1 GB/thread
  - **Reconstruction**: data and CPU intensive, high I/O (~200kB/s per core), fits well into 2 GB/thread, but can go down with tricks
  - Analysis: mostly data intensive (0 to 10 MB/s per core)
- While running specific workflows on HPC is interesting, the medium-long run aim is to be able to execute **ALL CMS workflows on every machine**
  - We want to process chained workflows (GEN+SIM+RECO) in a single task
  - Otherwise, data management and movement becomes problematic

Handshaking with PRACE

- CMS workflows have peculiar aspects with respect to more standard “HPC” workflows
  - **Architecture** (difficult to overcome x86_64 as primary arch)
  - **Data intensive**: from low I/O to very high I/O
  - Need for **remote data accesses** (possibly mediated by **edge caches**)
  - Need for **local virtualization** (docker, shifter, singularity, real VMs, …)
  - Need to **access remote services** (possibly mediated by **edge services**)
  - **CVMFS** preferred software distribution solution (but can be worked around with container)
  - CMS Workflow Management systems currently **absolutely require outgoing network** from the worker nodes due to the tight coupling of CMS WM with HTCondor.
- CMS is preparing a document detailing the
  - Needs
  - Desiderata
  - Possible fallback solutions (at which price)

Current known status with PRACE centers

- **PizDaint - CSCS**
  - Integration complete; talk @ Hepix
  - Site fully in production; scale tested up to 10k cores
  - CVMFS ok, Singularity ok, remote access ok
- **Marconi - CINECA**
  - Handshake ongoing with test jobs already running
  - CVMFS ok, Singularity ok, remote access mediated by edge services @ CNAF (trusted site)
- **Joliot Curie - CEA**
  - No known attempt
- **JUWELS - JSC**
  - No known attempt
- **MareNostrum - BSC**
  - Blocked by lack of external connectivity
- **SuperMUC - LRZ**
  - No known attempt

General thoughts

- A fast look at PRACE systems show that many of them are usable by CMS with just a few minor system adjustments
  - Those with base arch x86_64 are “simple”
  - Mostly **policy** changes wrt routing policies and user accesses
  - We hope the successful attempts with CSCS and CINECA are convincing to the rest of the PRACE community
- In the long term, CMS (and HEP) are committed to a greater utilization of the HPC installations - as requested by most of our funding agencies
  - If this is the directions we are pushed to by the FAs, we expect this to come via **multi-year guaranteed allocations**
- We would like to start a more organized handshaking procedure with the largest HPC providers globally (PRACE, DOE, NSF, ….)

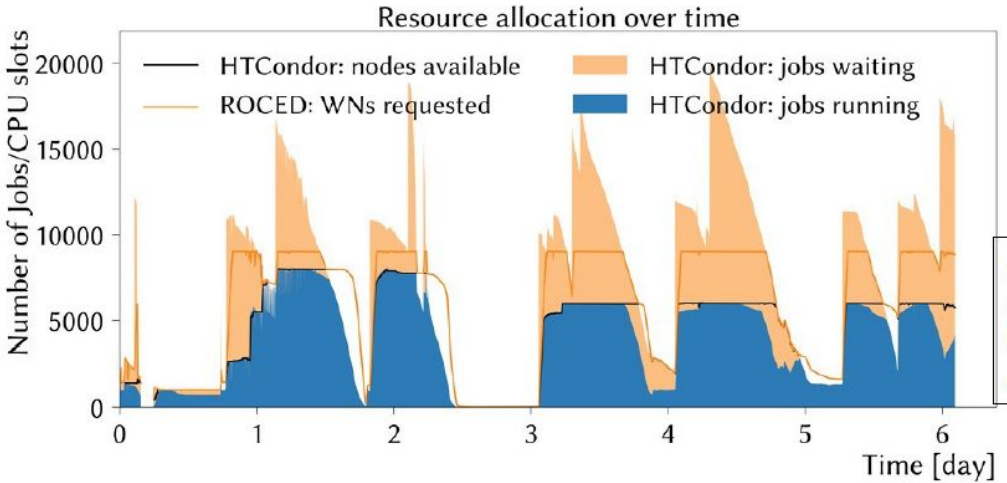
CMS related HPC Activities in Germany

- Dynamic integration of regional HPC resources
  - Freiburg HPC Center NEMO, shared among four user communities) *OpenStack*
  - KIT HPC Center ForHLR II *Singularity*
- Share: 10 Million CPU h/a for CMS-groups with Karlsruhe participation: MC production, τ-embedding and NNLO-calculations as well as user analysis jobs

Approach for resource integration and deployment presented at

CHEP 2018:
https://indico.cern.ch/event/587955/contributions/2937900/

ACAT 2017:
https://indico.cern.ch/event/567550/contributions/2696132/



Example of dynamic HPC usage: allocation of resources is driven by job load in local HTCondor

U.S.CMS HPC efforts

- Opportunistic processing of CMS parked data at Gordon(SDSC) in 2012
- First CMS allocation at DOE HPC in 2014 Carver (NERSC).
- In 2016 switched to **Edison and Cori** (NERSC) and also moved HPC access commissioning into the **HEPCloud** project.
- **HEPCloud** is a facility evolution project to provide a portal to an ecosystem of diverse computing resources commercial or academic. It will allow to seamlessly use resources that are local or remote.
- Current status in 2018: **We access resources at Cori (NERSC), Bridges (PSC) and Stampede2(TACC) through HEPCloud**. All of these sites are integrated into the CMS Workflow Management systems and run normal production jobs. **No special selection on workflow type except for some limited exclusions.**

U.S.CMS HPC efforts

We recently hit some of our scaling goals for NERSC. PSC is limited by the size of the allocation. TACC is used at very limited scale due to their batch queue policies (requires many-node jobs to scale higher).