Integrating LHCb workflows on HPC resources

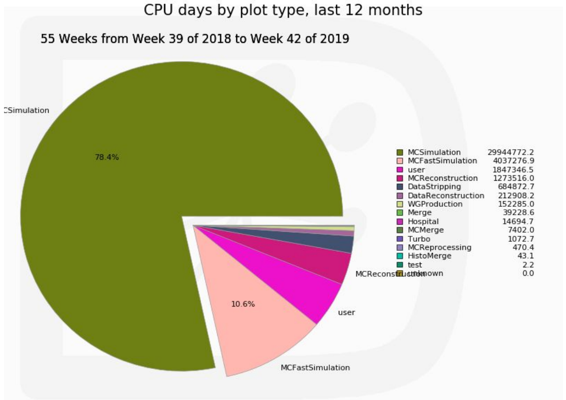
status and strategies

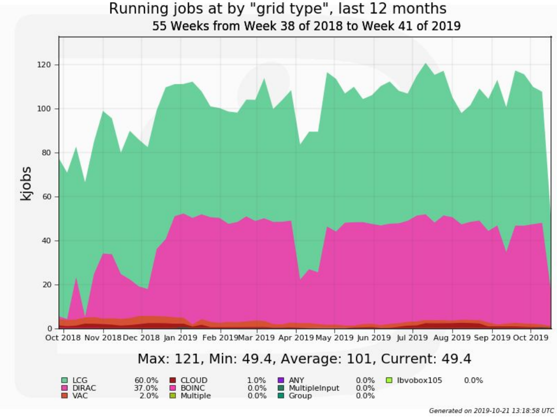
Federico Stagni (CERN)
Andrea Valassi (CERN), Vladimir Romanovsky (IHEP-Protvino/CERN)

4 November 2019 – CHEP 2019, Adelaide
https://indico.cern.ch/event/773049/contributions/3474807

LHCb overview

- MonteCarlo simulation jobs are, by far, the main consumers of the LHCb Grid computing capacity (will be >90% in Run3)
  - Includes event generation and detector simulation
  - Have ~no input data
  - Gauss, based on Geant4
  - Up to “yesterday”, productions only ran in
    - single processor mode
    - x86 CPUs
    - 2GB/processor
- (one of the) strategy - now, and later - extend computing resources to run more MC simulation
  - for non-simulations, hopefully we’ll find resources...





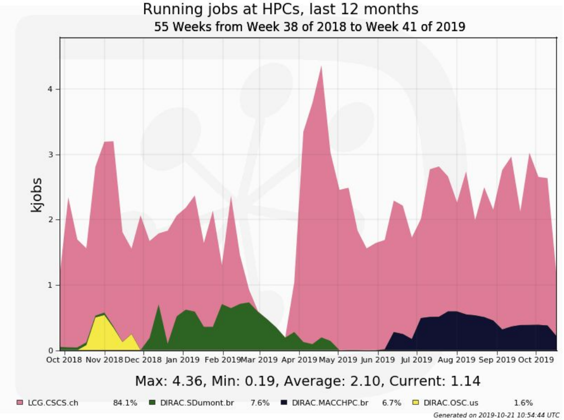
HPCs in LHCb

- As of today, some HPCs are available to LHCb

  And we run there x86, single processor jobs

  basically, “like the Grid”

- Today’s topic: running on Marconi HPC at **CINECA** (Italy)
  - Joint PRACE allocation for ALICE, ATLAS, CMS, LHCb
    - April 2019 to March 2020



See CHEP 2019 contribution #107
“Extension of INFN-T1 on a HPC system”
(this morning, 11:00)

General HPC challenges

- Software architecture challenges
  - HPCs include many-core (KNL@Marconi), non-x86 (ARM, Power9), GPUs…
    - And they are made for MPI, but in HEP we use individual nodes
- Distributed computing challenges
  - HPC site policies differ from those of HEP Grid sites
  - Authentication, authorization, network, storage, O/S, batch queues…
- Some HPCs are easier to exploit than others
  - e.g. LHCb already uses CSCS, which looks like an x86 Grid site
- HPCs are not the most natural fit for HEP today
  - For us, they are just clusters of individual nodes
  - But in HEP we must learn how to use them!

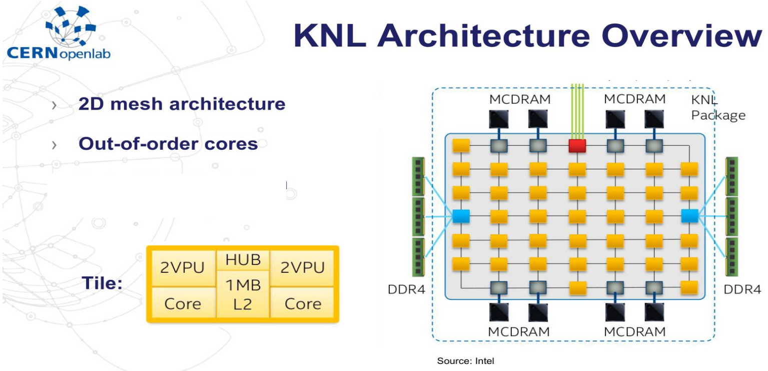
See CHEP 2019 contribution #55 "High Performance Computing for HL-LHC" (this afternoon, 14:00)

Software architecture challenges

CINECA/Marconi A-2

Marconi A-2: KNL

Luca Atzori, Dec 2016
CERN Openlab Technical Workshop



KNL Architecture Overview

- 2D mesh architecture
- Out-of-order cores

Tile:

Source: Intel

KNL on Marconi A2 at CINECA:

- ❑ 68-processors XeonPhi 7250
- ❑ 272 logical processors from 4x Hyper-Threading
- ❑ 96 GB DDR4 RAM
- ❑ i.e. *350 MB RAM per logical processor* (96/272)

Software challenge: <u>low memory per logical processor</u>

Need multi-process or multi-threaded software

GaussMP:
LHCb multi-process simulation

Focused on GaussMP

- LHCb MP simulation
- Interim solution until multi-threaded simulation (Gaussino) ready
  - We might also test this at CINECA later on

See CHEP 2019 contribution #285 “Gaussino - a Gaudi-based core simulation framework” tomorrow, R6

1st goal: validate GaussMP for production use

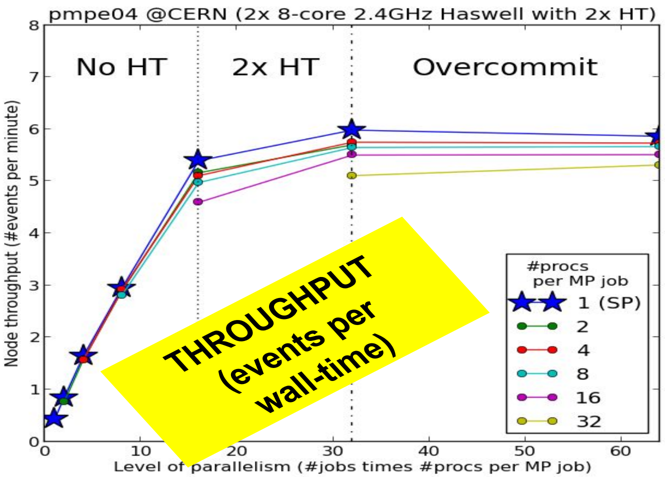
- Code from 2010, not used in production in LHCb previously
- Achieved same results event-by-event as in single-process

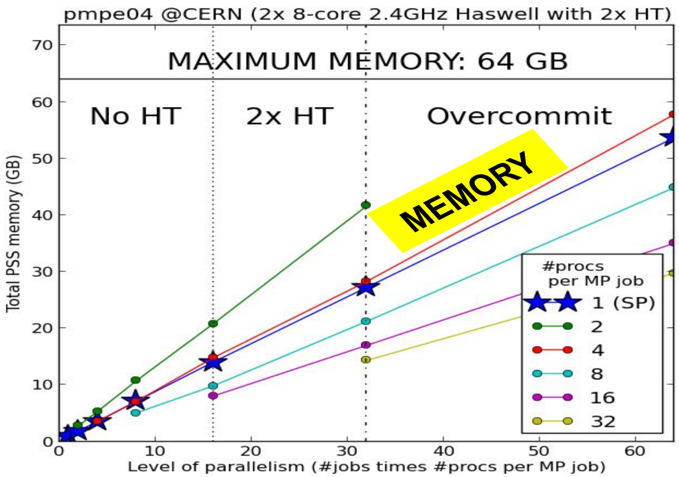
2nd goal: study performance scalability on KNL

- Using local batch
  see results on next two slides

GaussMP:
reference Haswell node at CERN

$B^+ \to J/\psi K^+$ events
(plus minimum bias spillover)





Throughput:
SP/MP scale well on 16 physical processors, ~10-20% extra gain from HT

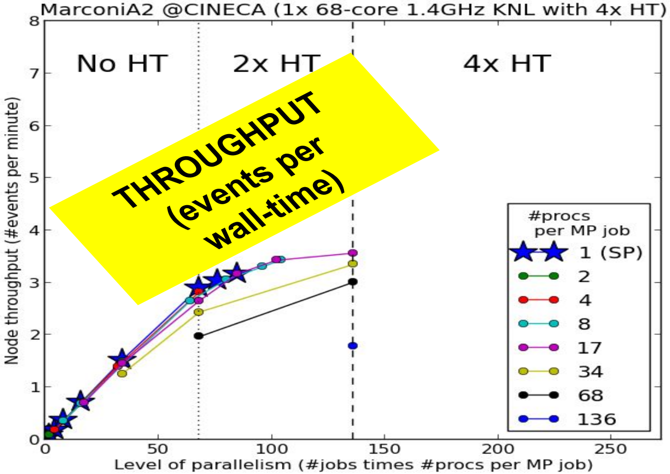
No throughput benefit from MP simulation with respect to SP if node memory is large enough

Memory:
SP is feasible with all virtual processors and even beyond the 2xHT region

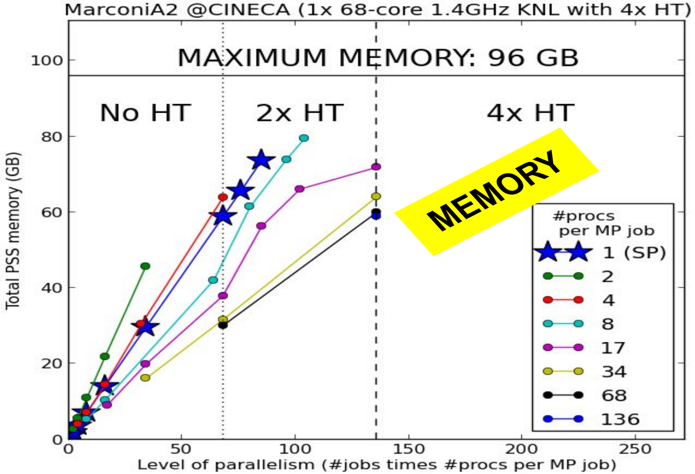
One SP job takes 0.9 GB memory, 64 jobs (on 16 processors) fit within the 64 GB budget

GaussMP:
Marconi KNL node at CINECA

$B^+ \to J/\Psi K^+$ events (plus minimum bias spillover)



THROUGHPUT (events per wall-time)



MEMORY

Throughput:

Maximum throughput for 8 jobs with 17xMP

MP gives ~10-15% extra gain with respect to SP

Memory:

SP cannot use more than 85 virtual processors, out of 272

MP reaches 136 (2x HT), but not more (out of memory)

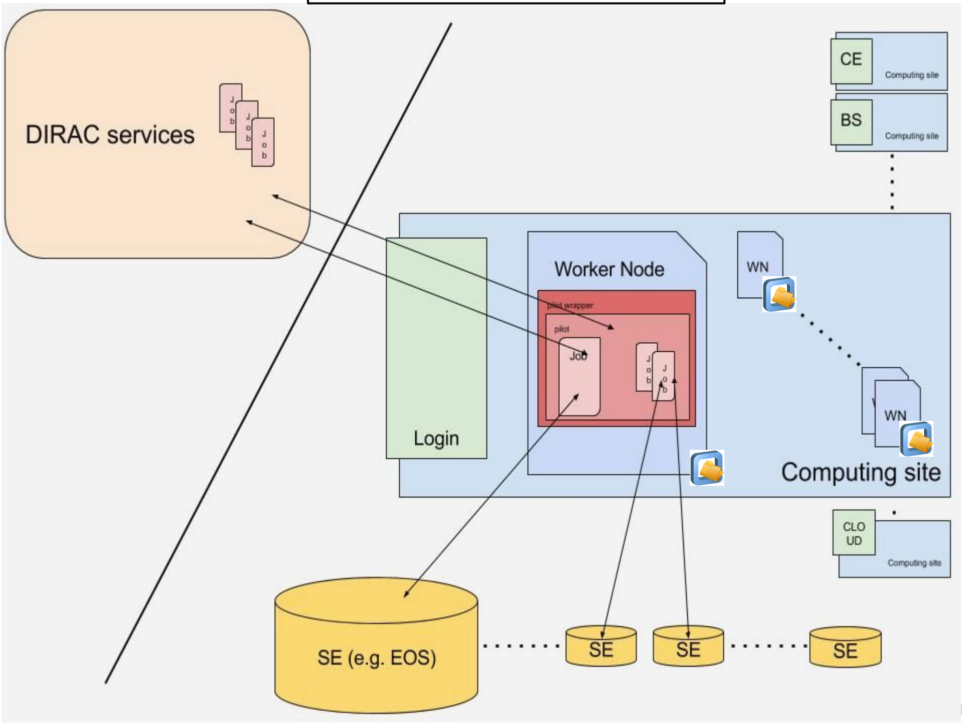
Earlier tests in 2018 on simpler events reached 272 (4x HT), but there was no throughput gain with respect to only using 136 (2x HT)

Distributed computing challenges

CINECA/Marconi A-2

General HPC challenges:
distributed computing

Schematic view of “the Grid”
and how it’s normally operated



~easy integration when:

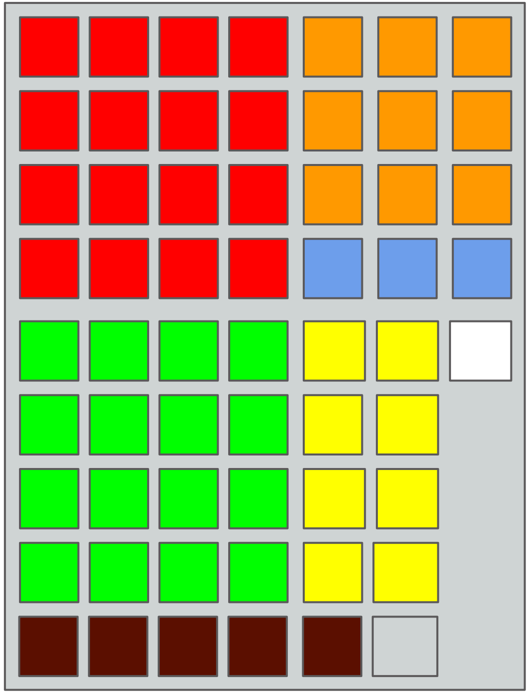
1. WNs have outbound connectivity
   a. Marconi: yes!
2. LHCb CVMFS endpoint(s) mounted on the WNs
   a. Marconi: yes!
3. SLC6 or CC7 “compatible”, or Singularity
   a. Marconi has Singularity, DIRAC pilots anyway run also on host OS

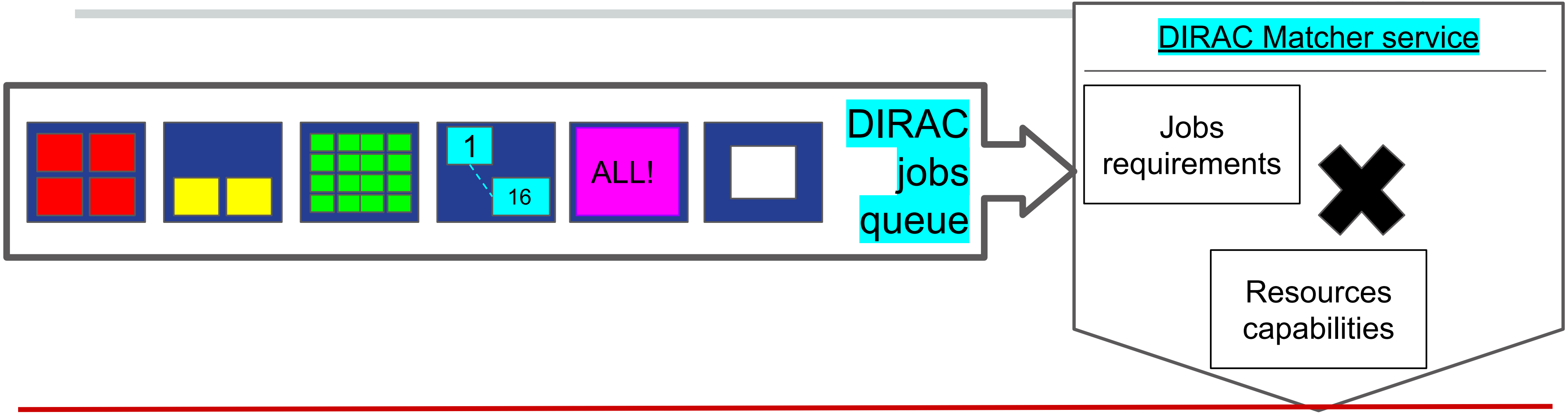
...everything's ~easy?

… but the jobs matching!

Fat nodes

- Exploiting many-core architectures
  - Never done in LHCb before
    - ...all SP jobs
- DIRAC needs to “partition” the node for optimal memory and throughput (and maybe only use a subset of the logical processors)
  - Use DIRAC “Pool”, an “inner Computing Element”
  - Parallel jobs matching





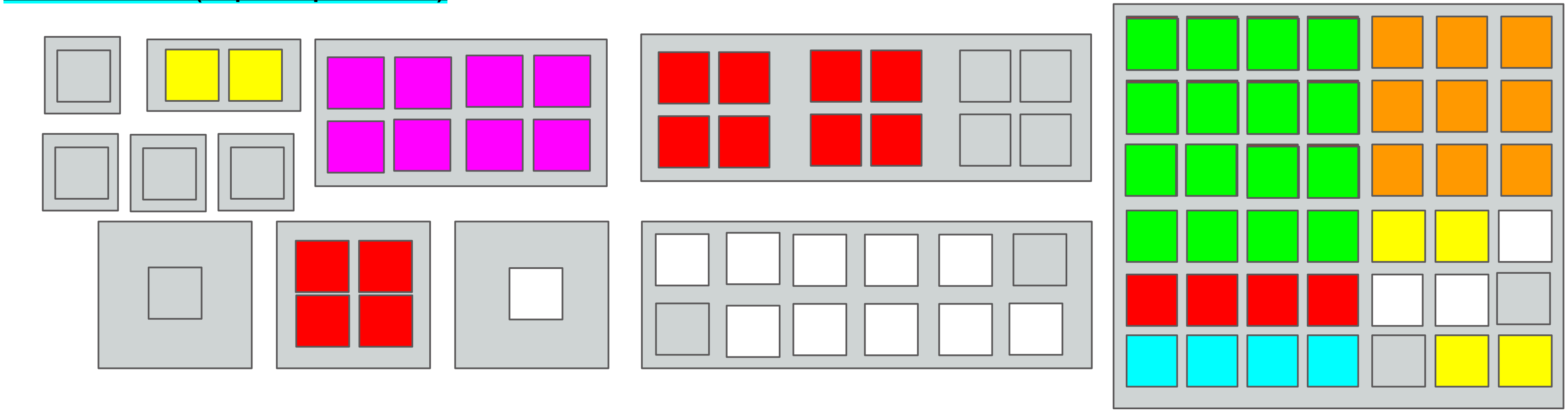
DIRAC Matcher service

DIRAC jobs queue

Jobs requirements

Resources capabilities

Resources (1 pilot per box)



Everything’s (now) possible

from DIRAC v7r0p4

will be useful not only for Marconi HPC

LHCb on Marconi A2:
status summary

- Software chain (GaussMP) is ready
  - Software validated, relevant hot patches released
- Distributed computing (LHCbDIRAC) is ready
  - Multi-processors aware Marconi site has been defined
- First integration tests have been performed
  - Successful parallel execution of GaussMP jobs on Marconi nodes, via DIRAC pilots

Outlook

- Next steps:
  - GaussMP production, compared with SP
    - for validating the physics
  - 2020: test multithreaded Gaussino on Marconi A2
  - Verify Job accounting
  - Memory as a matching parameter (~easy)

- HPC role predicted to rise in LHCb and HEP
  - We must be ready to exploit these new resources
  - Exploiting Marconi A2 took much effort but was ~easy, other supercomputers may be more complex
    - Collaboration with local site essential for computing integration
    - Main challenge ahead is porting software to GPUs (need G4)

Questions/comments

?

The 10th
DIRAC Users' Workshop

25th - 29th May 2020

KEK/IPNS
TSUKUBA

KEK

DIRAC
THE INTERWARE

HTTPS://INDICO.CERN.CH/E/DUW10

Organizers: FEDERICO STAGNI (CERN)
TAKANORI HARA (KEK/IPNS)
IKUO UEDA (KEK/IPNS)
ANDREI TSAREGORODTSEV (IN2P3)

diracgrid.org DIRACGrid dirac-grid dirac.readthedocs.io

10th DIRAC
Users' Workshop

indico.cern.ch/e/DUW10

25-29 May 2020
KEK, Japan

DIRAC approach

Pilots are the "federators"

Send them

as "pilot jobs" (via a CE)

Or just **Run them!**

e.g. as part of the contextualization of a (V)M

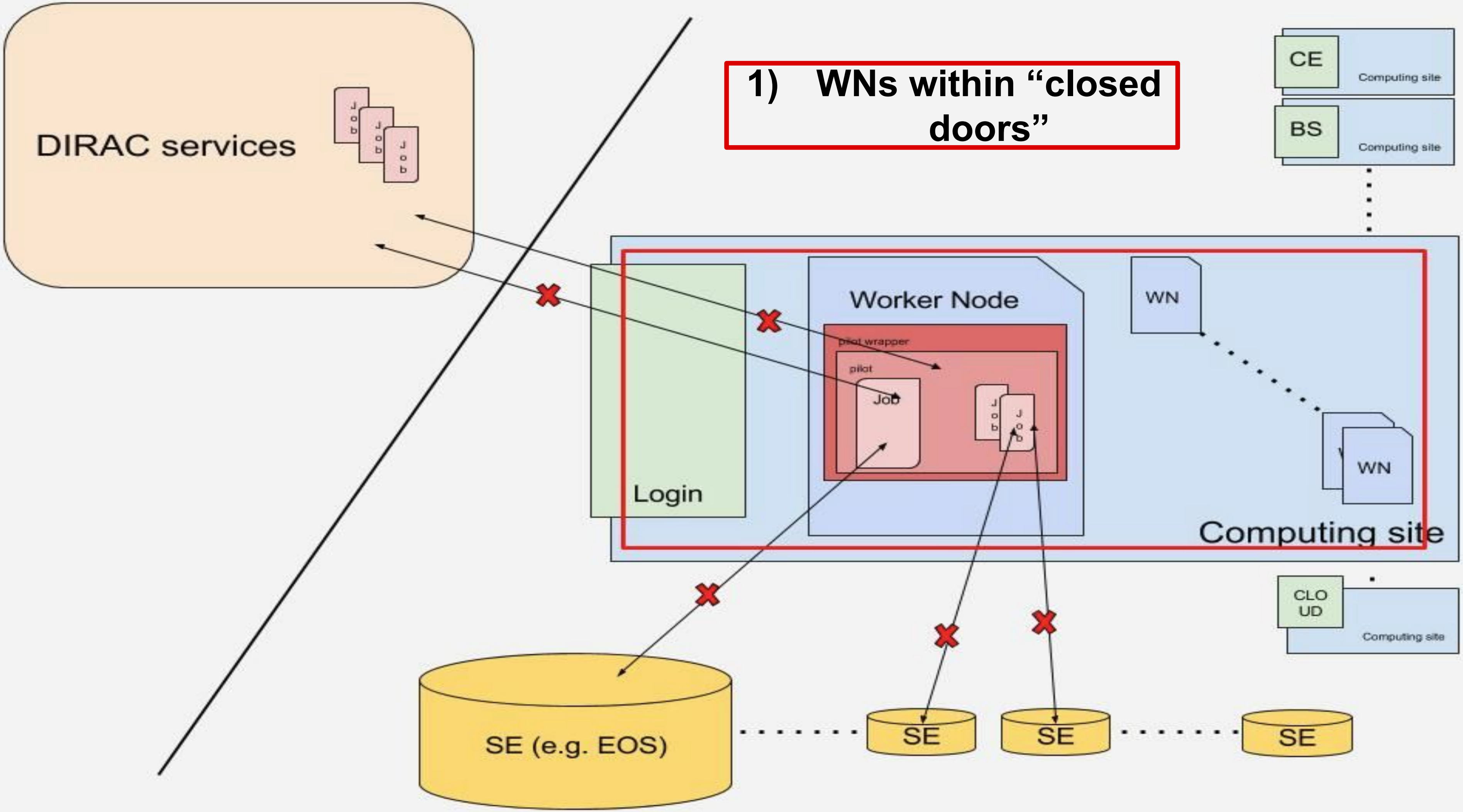
OR

"Make a machine a pilot machine, and you are done"

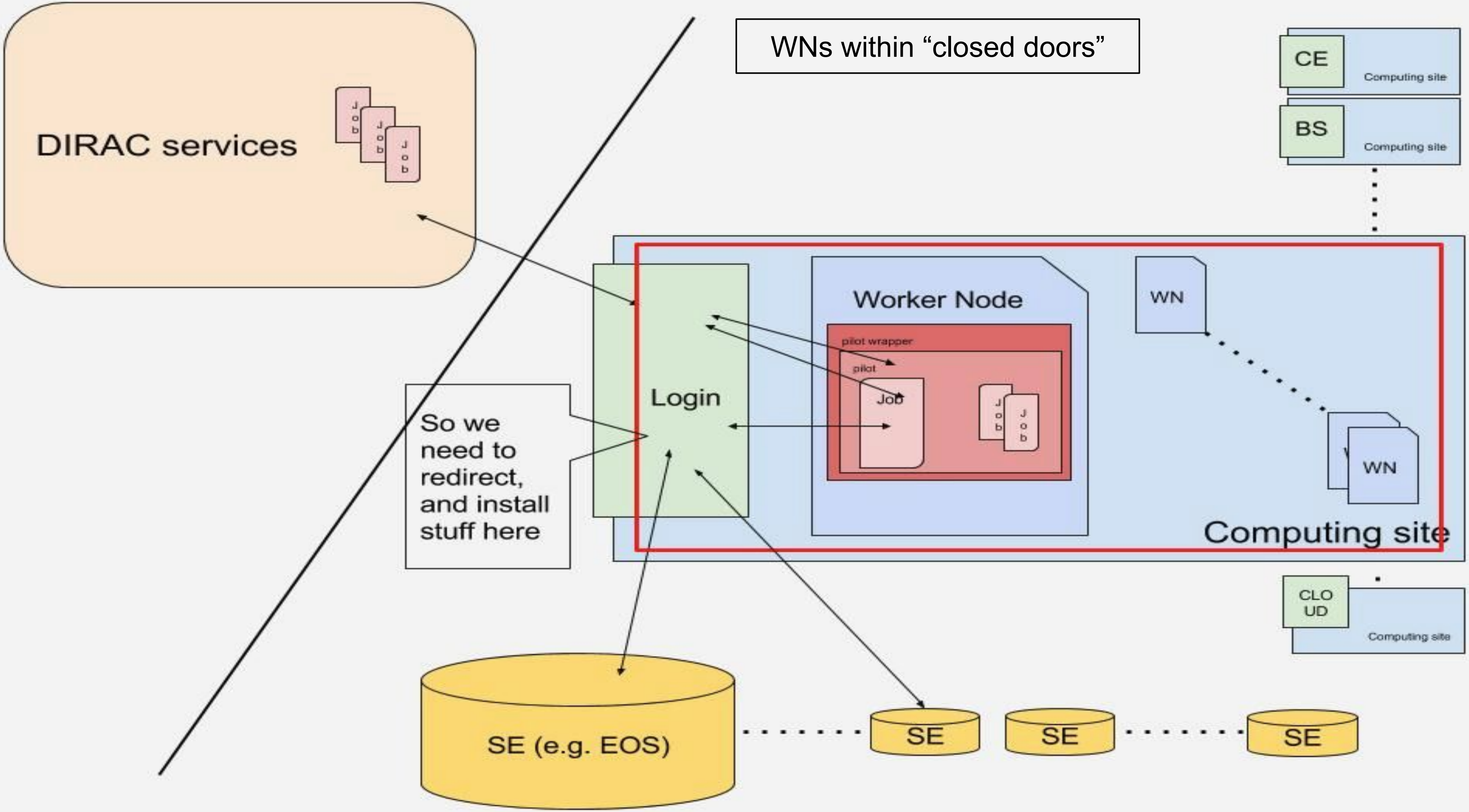
Once started, pilots will:

1. Install a DIRAC client
   - together with dependencies
     - the “container” is shipped → a “container” is not necessarily an image
2. Self-discover WN capabilities
   - Including CPU power and capabilities
     - Using DB12 or MJF
     - And #processors
   - And memory
3. Use a “JobAgent” to match the capabilities of the WN with the requirements of the waiting jobs.
4. Send monitoring info
   - A list of messages like
     - "I've booted up" ...
     - "I found the DIRAC pilot ok" ...
     - "I'm about to shutdown"...
   - Self-upload their own logs before shutting down

1) WNs within “closed doors”

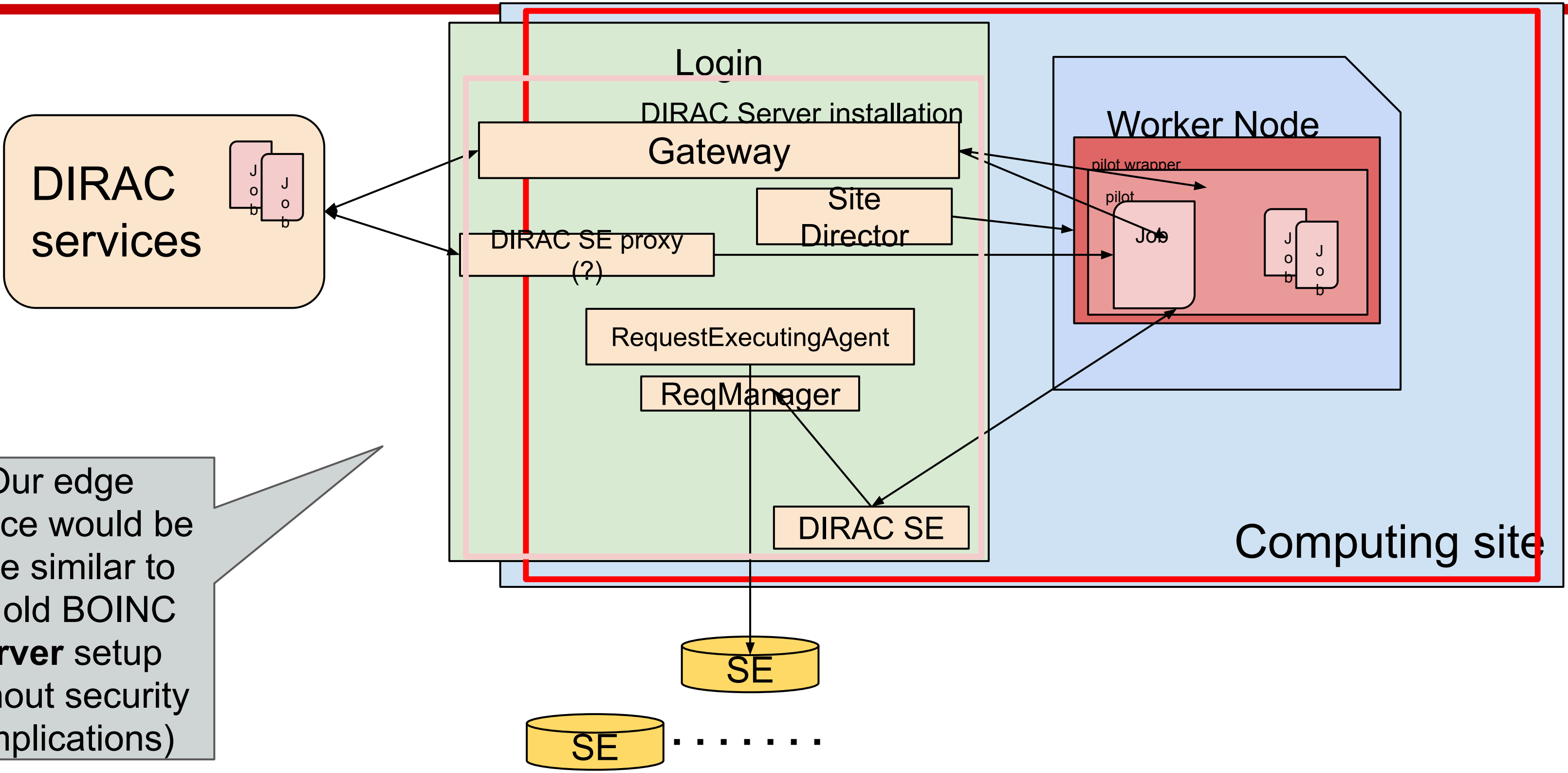


WNs within “closed doors”



So we need to redirect, and install stuff here

Zooming in...



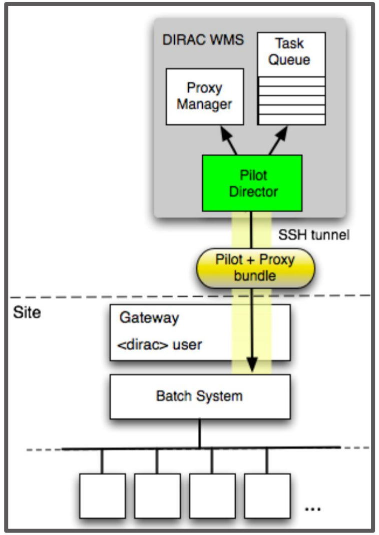
Our edge service would be quite similar to our old BOINC **server** setup (without security complications)

Computing Element

CE or no CE:

- At CINECA there’s a HTCondorCE
  - so, on our side, we simply added its backend
- If there would have been no CE, there would have been 2 possibilities:
  - the first is using a “DIRAC SSH CE”, which is a very simple virtual Computing Element that only requires a SSH key pair to be established.
    - and then DIRAC would talk directly with the Batch System (SLURM for CINECA)
  - a DIRAC pilot factory can be setup, local to the HPC.

SSH CE



DIRAC capabilities
jobs requirements

From a users' perspective:

- certain jobs may be able to run only in SP mode
- certain jobs may be able to run only in MP mode (meaning: need at least 2 processors)
- certain multi processor jobs may need a fixed amount of processors
- certain jobs may be able to run both in SP or MP mode
  - depending on what's possible on the WN/Queue/CE
- for certain jobs we may want to specify a maximum number of processors to use

Last 2 bullets have been added specifically for running at Marconi HPC

It's possible to describe the jobs precisely enough to satisfy all use cases above.

More info about matching here and here

DIRAC capabilities
resources description

Resource providers

- may give their users the possibility to run on their resources:
  a. only single processor jobs
  b. only multi processor jobs
  c. both single and multi processor jobs
- may ask their users to distinguish clearly between single and multi processor jobs
- may need to know the exact number of processors a job is requesting
- may ask for only "wholeNode" jobs

It's possible to describe CEs and Queues precisely enough to satisfy all use cases above.