Who are We

The Nordic e-Infrastructure Collaboration (NeIC) facilitates development and operation of high-quality e-infrastructure solutions in areas of joint Nordic interest. NeIC is a distributed organisation consisting of technical experts from academic institutions across the Nordic countries.

Nordic WLCG tier-1 facility (NDGF)

The Nordic distributed tier-1 facility for the worldwide computing grid serving the large hadron collider at CERN.

The LHC Experiment

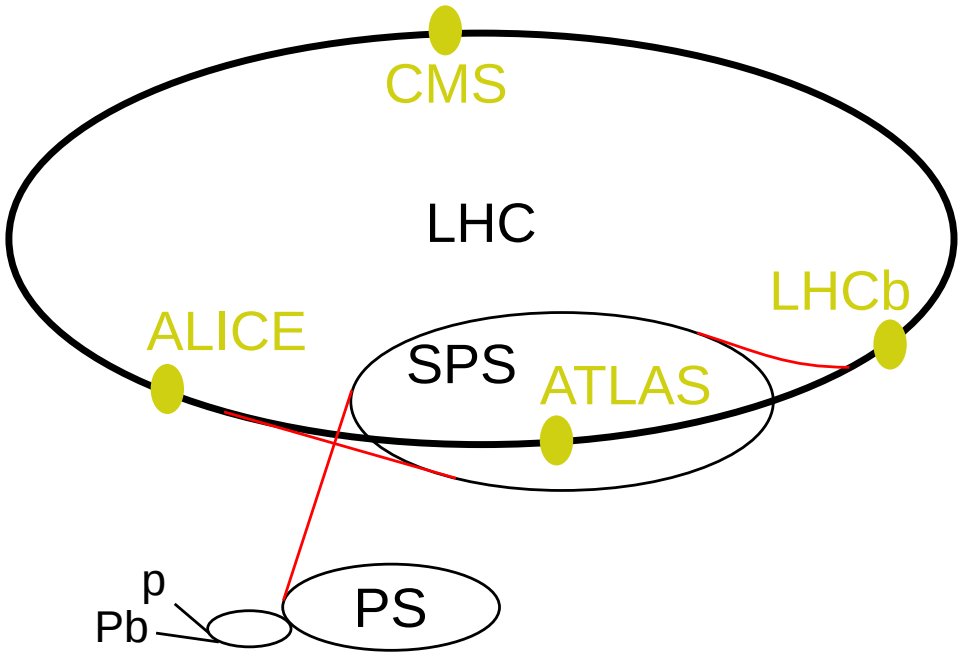
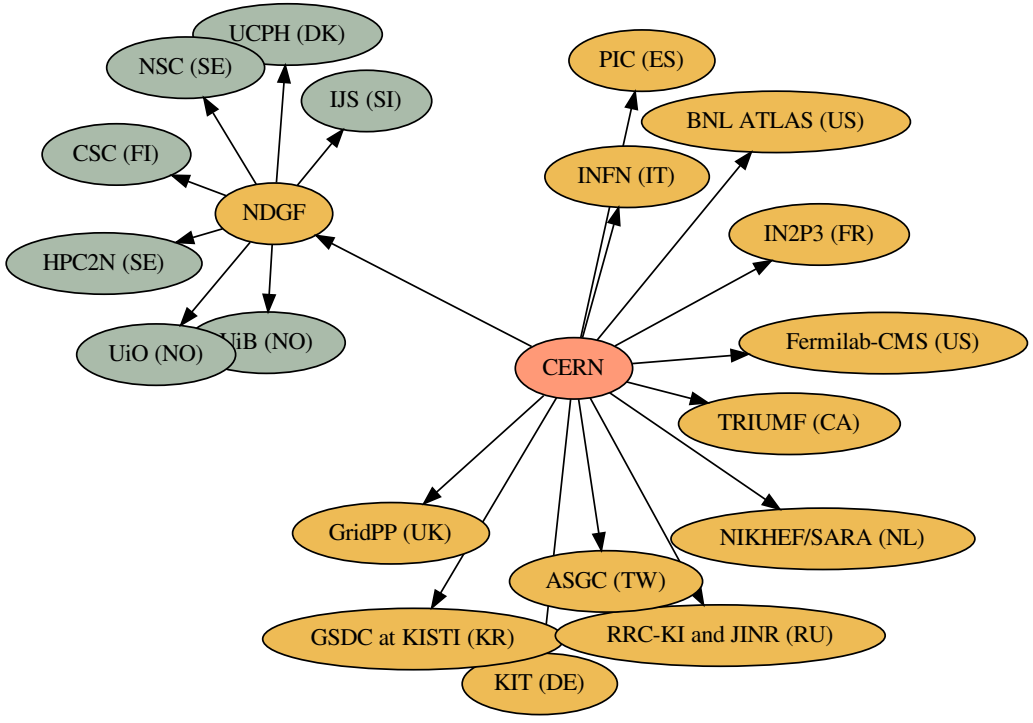


Figure 1: Overview of the LHC experiment. Illustration by Arpad Horvath.

- 150 million sensors triggering at 20-40 MHz each producing 1-2 MB.
- Triggers reduces the 20-80 TB/s by a factor 10^3 to 10^4.
- 50 PB per year: ATLAS 1 GB/s, CMS 1 GB/s, LHCb 0.6 GB/s, ALICE several GB/s during heavy ion runs

The LHC Tier 1 Sites



NDGF is a distributed Tier 1:

- A central dCache runs on Ganeti and coordinates storage activities
- Operations on Disk and tape storage is delegated to Nordic data centres.

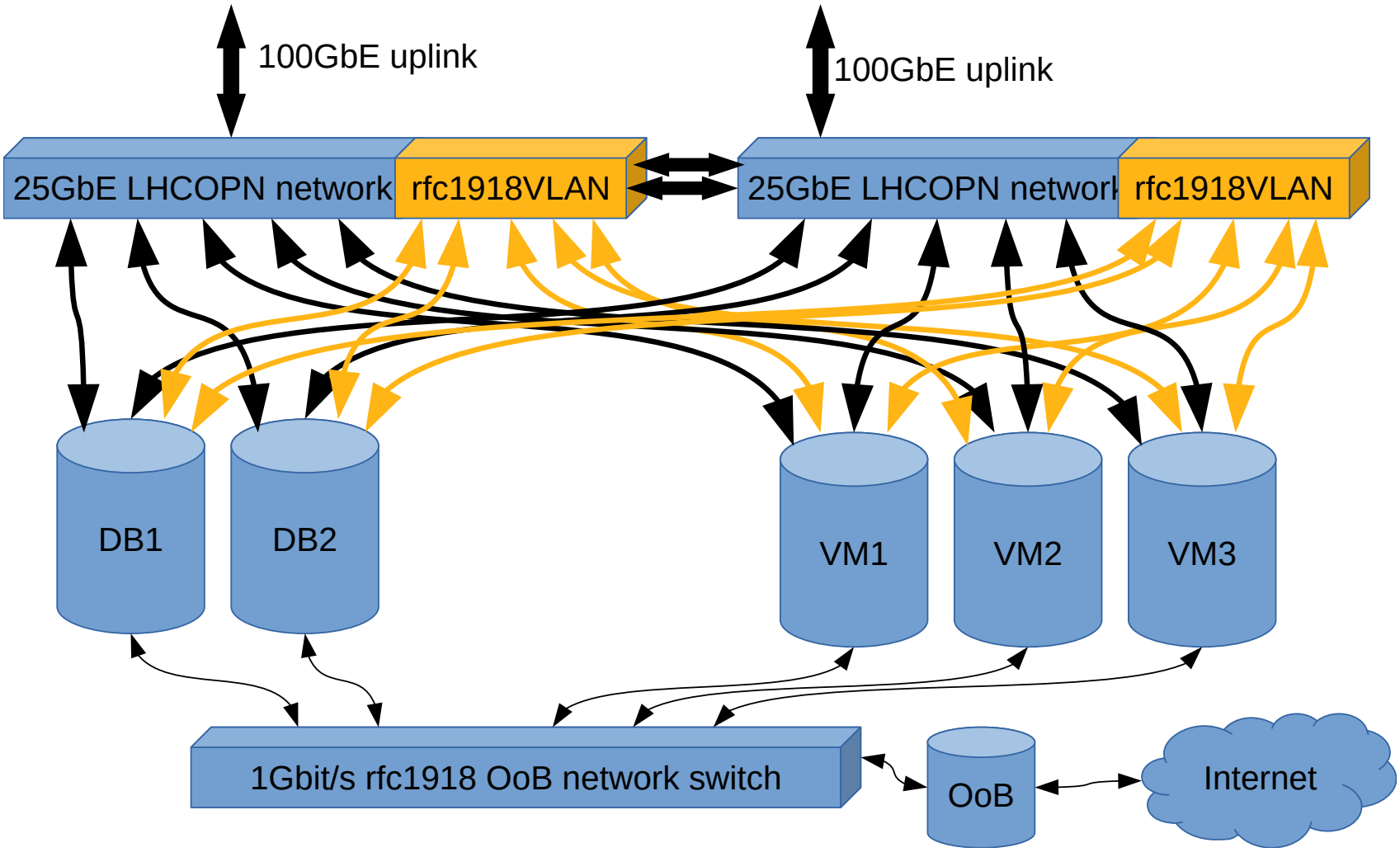
Our Use of Ganeti

The current situation:

- Critical (24×7 on-call) services run on a 2 node Ganeti cluster at NORDUnet in Denmark.
  - dCache headnodes (redundant, 2+ VMs)
  - Zookeepers (redundant, 3 VMs)
  - PostgreSQL run *outside* Ganeti (hot standby, both nodes)
  - DNS and core grid-related indices and other infrastructure
  - Monitoring
- Less critical infrastructure run on a 6 node cluster at HPC2N in Umeå.
  - preproduction dCache headnodes (redundant, 2+ VMs)
  - preproduction Zookeepers (redundant, 3 VMs)
  - preproduction PostgreSQL (hot standby, 2 VMs)
  - ElasticSearch (3 VMs with 1 TB DRBD devices each)
  - More monitoring

We plan replacing the former with 2 dedicated PostgreSQL nodes and 3 Ganeti nodes:

Plan for New Hardware (1/2)



The final solution may or may not have a redundant uplink. (Illustration: Mattias Wadenstein)

Plan for New Hardware (2/2)

2 switches with full LCAP redundancy:

- Dell S5212-ON switches, 12 $\times$ 25 Gb/s + 3 $\times$ 100 Gb/s

2 PostgreSQL nodes:

- Poweredge R640
- 2 $\times$ Xeon Gold 5222 4C/8HT
- 192 GB RAM
- 5 $\times$ 480 GB SSD
- 3 $\times$ 2TB 7.2K HDD

3 Ganeti nodes:

- Poweredge R640
- 2 $\times$ Xeon Gold 5222 4C/8HT
- 192 GB RAM
- 5 $\times$ 480 GB SSD

More Ganeti Details

Node setup:

- Node OS: Ubuntu 18.04 (but still 16.04 for the main cluster)
- Hypervisor: KVM
- Disk template: DRBD (c-values and resync-rate tuned)

VM creation:

- SNF image has works well, but looking into cloud-init.
- Ubuntu (16.04, 18.04) and CentOS (6, 7) VMs
- VMs optionally created with Ansible using on host/group variables
- Static IPs, need `net.ifnames=0` for reliable interface names.

Operation:

- Ansible: node and VM upgrade/reboot, image upgrade/dump

Ganeti Experience

Ganeti is easy to set up and user friendly but there are occasional snags:

- Wish: Emergency shutdown of VM. A combination of killing qemu and gnt-instance is currently needed.
- Disk migration at NBI very slow after upgrade. Tuning DRBD resync solved it.
- Cannot cancel scheduled disk migration tasks even if not started.
- Issue with VM migration not finishing for a very active VM.
- DRBD takes expertise when it breaks.

Ansible Modules

```
> GNT_INSTANCE
    This module can create and remove instances and manage
    their state using the gnt-instance command.

- allow_remove
- be_params
- hv_params
= name
- os_params
- os_params_private
- os_params_secret
- os_size
- os_type
= state = present|running|stopped|rebooted|absent|reinstalled

> DRBD_WAIT
    This module repeatedly polls /proc/drbd, checking if there
    are DRBD devices which are out of sync until all devices
    are in sync.  If a permanent issue is detected it fails.
```

Ansible Role to Upgrade and Reboot Cluster (1/3)

```
- name: Autoremove packages
  apt: {update_cache: false, autoremove: true}

- name: Upgrade host
  apt: {update_cache: true, upgrade: dist}

- name: Check whether the node needs a reboot
  stat: {path: /var/run/reboot-required}
  register: reboot_required

- when: reboot_required.stat.exists
  block:

    - name: Check if the current node is master
      shell: "gnt-cluster getmaster | tr -d [:space:]"
      register: master

    - assert: {that: master.stdout in ganeti_master_candidates}
```

Ansible Role to Upgrade and Reboot Cluster (2/3)

```
- name: Failover master if needed.
  command: "gnt-cluster master-failover"
  delegate_to: "{{ganeti_master_candidates|difference([inven
  when: master.stdout == inventory_hostname

- name: Evacuate primary instances
  command: >
    gnt-node evacuate -f --ignore-soft-errors --primary-only
             {{inventory_hostname}}
  delegate_to: "{{ganeti_master_alias}}"

- name: Confirm reboot
  pause: {prompt: "Press ENTER to reboot {{inventory_hostnam

- name: Scheduling downtime for this Ganeti node and its ser
  nagios: {action: downtime, minutes: 20, service: host,
           host: "{{inventory_hostname}}"}
  delegate_to: "{{nagios_host}}"
  with_items: [host, all]
  when: nagios_host is defined
```

Ansible Role to Upgrade and Reboot Cluster (3/3)

```
    - name: Reboot
      command: "shutdown -r +1"

    - name: Wait for host and ssh to be running again
      wait_for:
        timeout: 900
        delay: 320
        sleep: 5
        host: "{{inventory_hostname}}"
        port: 22
      delegate_to: "{{ganeti_master_alias}}"

    - name: Wait for DRBD devices to recover
      wait_for_drbd:

- name: Autoremove packages again
  apt: {update_cache: false, autoremove: true}
```

Upgrading Ganeti 2.15 to 2.16

Docker container to build packages:
https://gist.github.com/paurkedal/dfd240e3fa07e46f5fbbfeade8371999

- Install explicitly versioned 2.16 packages parallel to 2.15 packages:

```
> dpkg -i ganeti-2.16_2.16.0-1ubuntu1_all.deb \
          ganeti-haskell-2.16_2.16.0-1ubuntu1_amd64.deb \
          ganeti-htools-2.16_2.16.0-1ubuntu1_amd64.deb \
          ganeti_2.16.0-1ubuntu1_all.deb
> apt install -f
```

- Make sure the cluster is in a good state:

```
> gnt-cluster verify
```

- Upgrade:

```
> gnt-cluster upgrade --to 2.16
```

- Fix all crypto issues:

```
> gnt-cluster renew-crypto \
    --new-node-certificates \
    --new-spice-certificate \
    --new-rapi-certificate \
    --new-cluster-certificate
> gnt-cluster verify
```

Upgrading Ubuntu 16.04 to 18.04

```
# On master:
> gnt-cluster master-failover # if needed
> gnt-cluster verify
> gnt-node migrate node
> gnt-node list
> gnt-node modify -O yes node

# On node:
> systemctl stop ganeti
> apt remove \*ganeti\*
> do-release-upgrade
> reboot
> apt install ganeti
> systemctl status ganeti

# On master:
> gnt-node add --readd node OR gnt-node modify -O no node
```