HTCondor Week Europe 2025

UK Tier-1 GridPP

Tom Birkett

STFC, Rutherford Appleton Laboratory

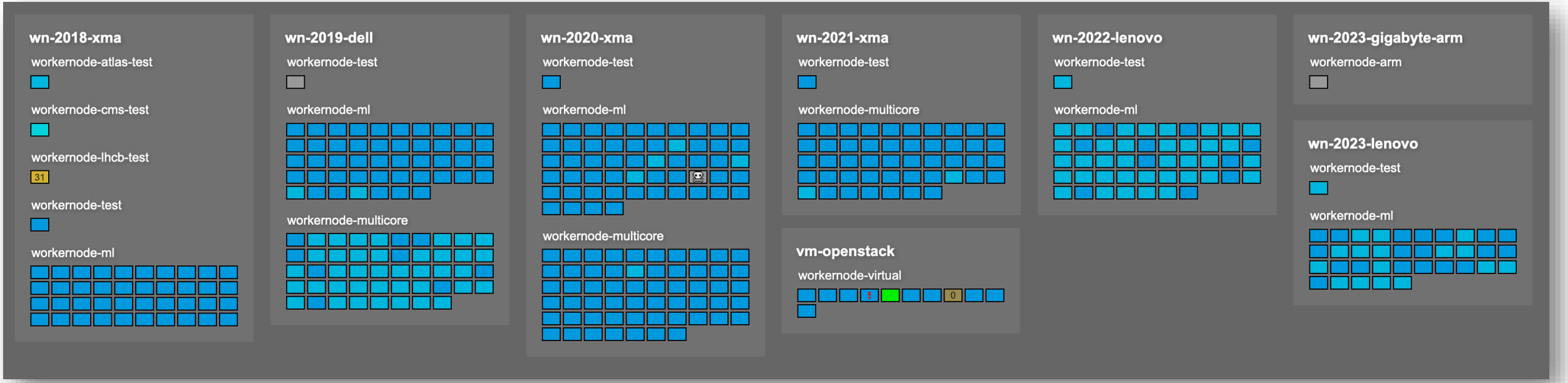
Science and Technology Facilities Council

Scientific Computing

GridPP

UK Computing for Particle Physics

Batch Farm overview

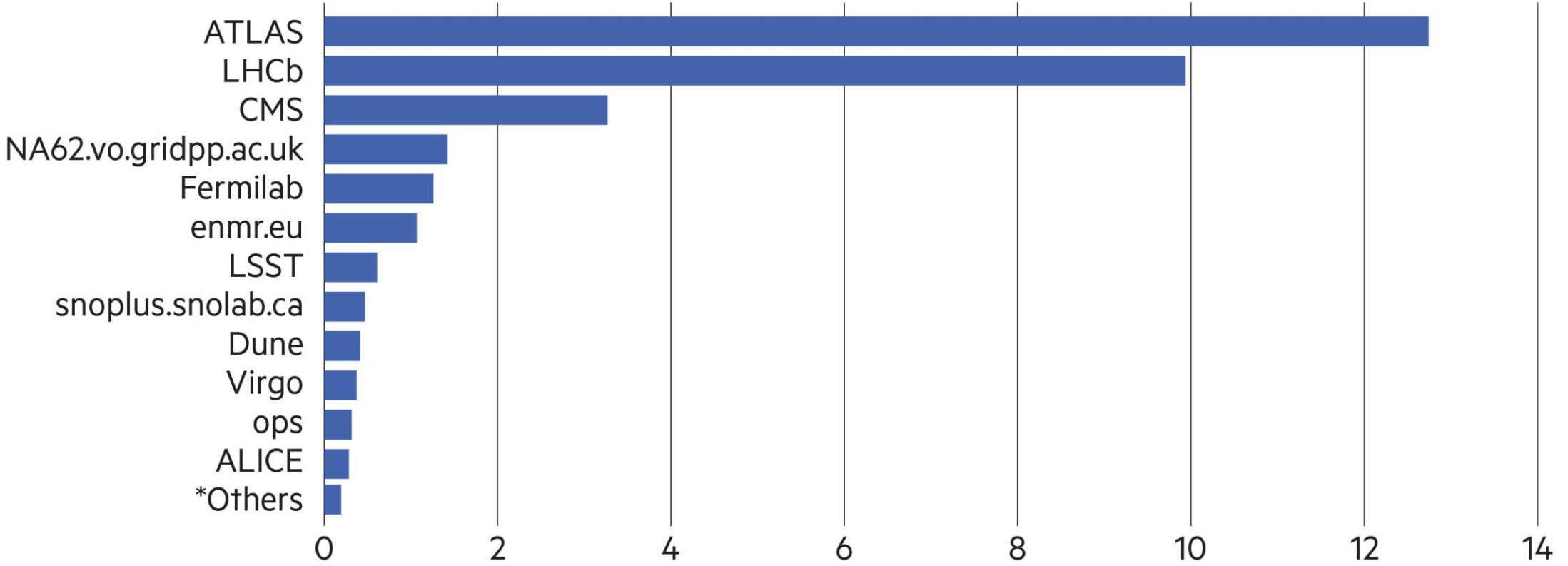


- Aug 2024:
  - 375 worker nodes.
  - ~ 72,000 logical CPU cores
  - HepScore: ~ 948,500
  - Job Slots: 66,296
- Aug 2025:
  - 375 worker nodes. + Cloud
  - ~ 72,000 + dynamic Cloud CPU cores
  - HepScore: ~ 948,500 +
  - Job Slots: 66,296 +

Mimic: https://github.com/stfc/mimic

Number of jobs per VO (million)

Aug 2024 - Aug 2025

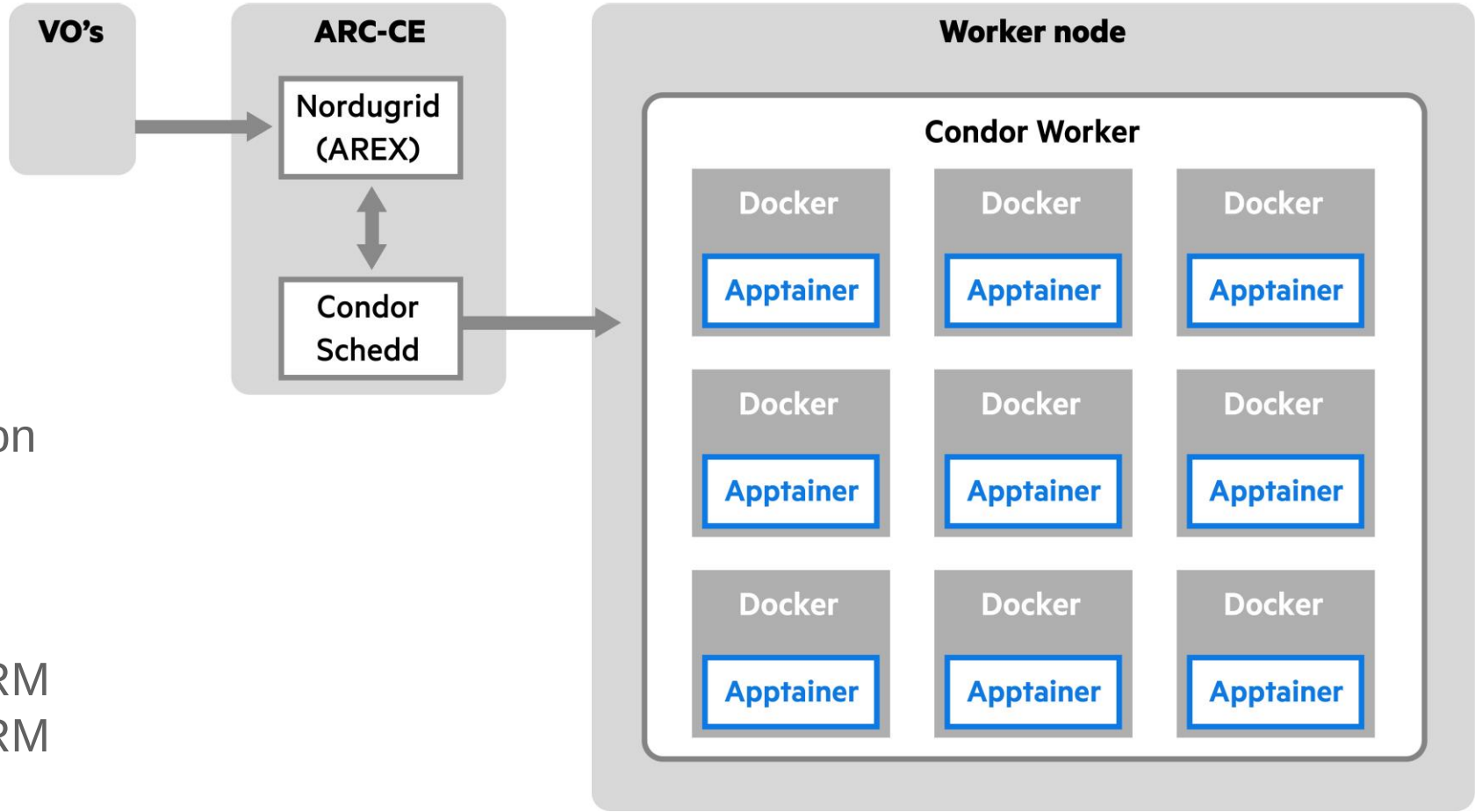


32,360,698 total jobs

*Including: Biomed, t2k.org, hyperk.org, comet.j-parc.jp, dteam, eucliduk.net, GridPP, ilc, pheno, vo.moedal.org

Job lifecycle

- Job factory submits to CE
- Job description is read by ARC-CE
- Worker (Execution Point) picks up job from schedd, starts Docker container
- Allows experiments to have tailored containerised environment for job execution
- RAL has four job queues:
  - `EL8`: Rocky 8
  - `EL9`: Rocky 9
  - `EL8-ARM`: Rocky 8 ARM
  - `EL9-ARM`: Rocky 9 ARM



Request to HTCondor team

- Conditional statements on Docker extra arguments (DOCKER_EXTRA_ARGUMENTS)
- Similar functionality to `DOCKER_VOLUME_DIR_xxx_MOUNT_IF`, maybe `DOCKER_EXTRA_ARGUMENTS_xxx_INCLUDE_IF`?
- The `docker.py` wrapper used at RAL, builds the docker run command that is executed by Condor. https://github.com/stfc/ral-htcondor-tools/blob/master/docker.py
  - `DOCKER=/usr/local/bin/docker.py`
- Our intention is to continue sunsetting the `docker.py` script and move to using native ClassAd language in Condor

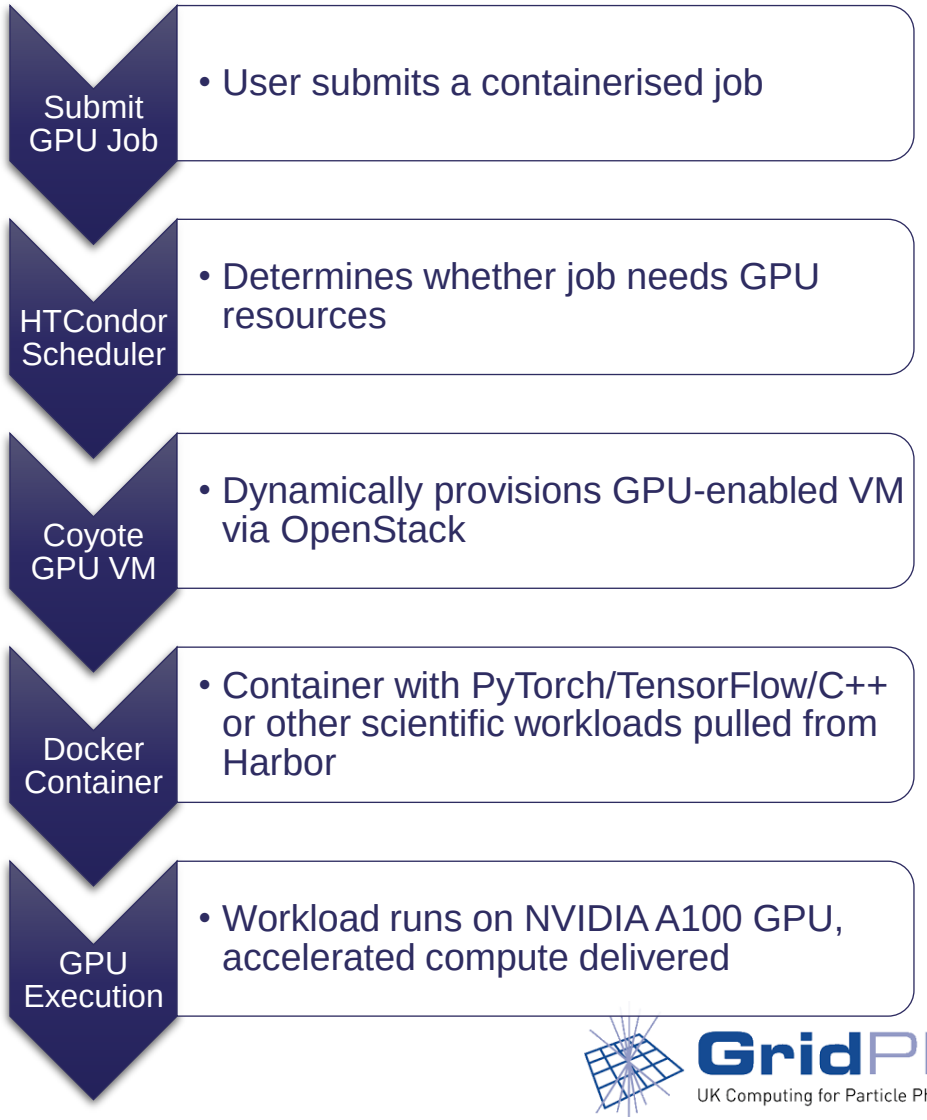
HTCondor transforms at RAL

```
# Convert job to Docker universe

SCHEDD_CLASSAD_USER_MAP_NAMES = $(SCHEDD_CLASSAD_USER_MAP_NAMES) nordugrid_queues_map
CLASSAD_USER_MAPFILE_nordugrid_queues_map = /etc/condor/maps/nordugrid_queues_map

JOB_TRANSFORM_NAMES = $(JOB_TRANSFORM_NAMES) DefaultDocker, MinMem, lsst

JOB_TRANSFORM_DefaultDocker @=end
   REQUIREMENTS (JobUniverse is 5) && (DockerImage is undefined)
   SET WantDocker = true
   EVALSET DockerImage = usermap("nordugrid_queues_map", NordugridQueue, "missing-default", "")
   COPY TransferInput = OriginalTransferInput
   EVALSET TransferInput = strcat(OriginalTransferInput, ",", Cmd)
   EVALSET JobMemoryLimit = ( JobMemoryLimit * 3 )
   SET Requirements = TARGET.HasDocker && $(MY.Requirements)
@end

JOB_TRANSFORM_MinMem @=end
   REQUIREMENTS RequestMemory < 4096
   SET RequestMemory = 4096
   SET JobMemoryLimit = 12582912
@end

JOB_TRANSFORM_lsst @=end
   REQUIREMENTS VOName is "lsst"
   SET Environment = "LSST_LOCAL_PROLOG=/etc/lsst/prolog.sh"
   SET Requirements = (TARGET.RalCluster is "wn-2020-xma" || TARGET.RalCluster is "wn-2021-xma" || TARGET.RalCluster is "wn-2022-lenovo" || TARGET.RalCluster is "wn-2023-lenovo") && $(MY.Requirements)
@end
```

Integrating GPU compute into the Tier-1 Batch Farm

Technical Achievements:

- **Provisioned GPU-Enabled VM:** Set up NVIDIA A100 GPU with CUDA and container toolkit for GPU workloads.
- **Built & Containerised GPU Applications:** Developed Docker images for C++, PyTorch, and TensorFlow models; deployed to Harbor for seamless access.
- **Enabled GPU Job Scheduling in HTCondor:** Configured HTCondor to recognise and allocate GPU resources efficiently.
- **Extended Coyote for GPU VM Automation:** Modified provisioning logic to dynamically spin up GPU-enabled worker nodes via OpenStack.



Science and Technology Facilities Council
Scientific Computing

GridPP
UK Computing for Particle Physics

IPv6 on Workernodes and Docker

- Docker natively supports IPv6. Implementation methods vary, RAL configures IPv6 in the following way:
- `/etc/docker/daemon.json`:

```
{
  "bip": "10.20.0.1/20",
  "experimental": true, (Not required since Docker 27)
  "ip6tables": true, (Default since Docker 27)
  "live-restore": true,
  "log-driver": "local",
  "storage-driver": "overlay2"
}
```

IPv6 Docker networking continued

- Setup Docker network with IPv6 config:

docker network create \
--driver=bridge \
--subnet=172.28.0.0/16 \
--ip-range=172.28.4.0/23 \
--gateway=172.28.5.254 \
--ipv6 \
--subnet=**2001:db8:54:3:82f6:dd83:1000::/112** \
--ip-range=**2001:db8:54:3:82f6:dd83:1000:8000/113** \
--gateway=**2001:db8:54:3:82f6:dd83:1000:1** \
ralworker

- Host IPv6 network config:

IPv4:
inet xxx.xxx.221.131/26

IPv6:
inet6 **2001:db8:54:3:82f6:dd83::/64**
inet6 fe80::63f:72ff:fed4:4688/64

Future work / plans

- Continue to keep in step with future HTCondor LTS releases
- Explore migrating from `PASSWORD` authentication between daemons to `TOKEN` authentication
- Continue deprecation of `docker.py`
- GPUs, GPUs and more GPUs!
- Expand capability of Coyote allowing for greater opportunistic resource within RAL
- More monitoring (many thanks to all presentations about monitoring so far!)

Science and Technology Facilities Council

Questions?

thomas.birkett@stfc.ac.uk