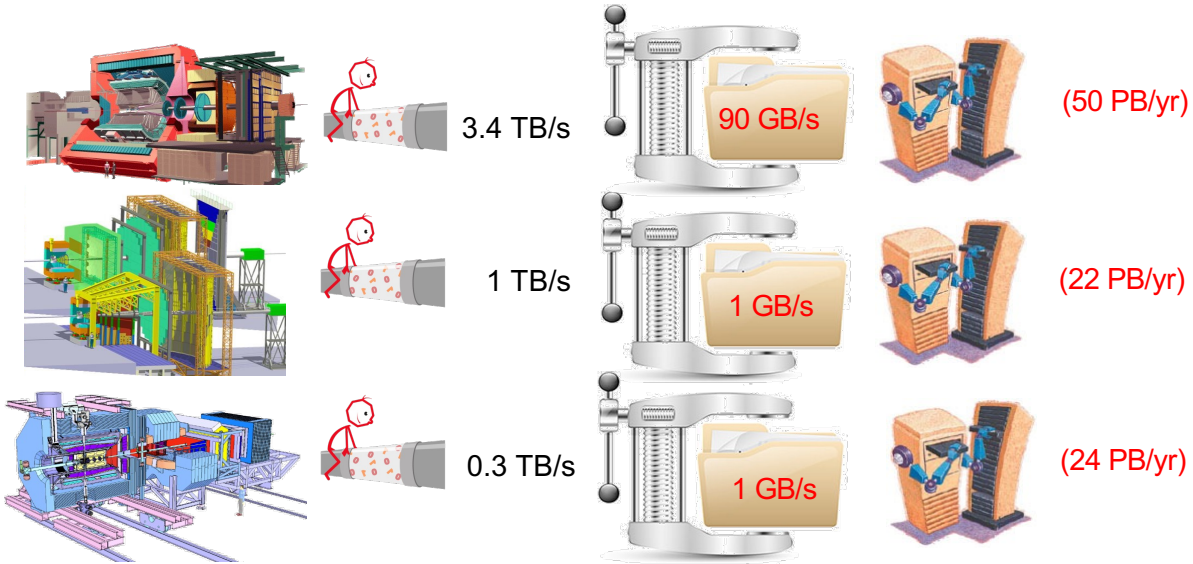
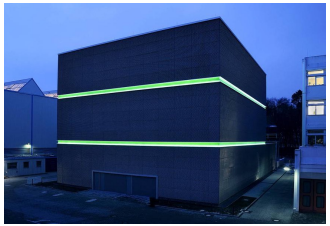
Modular toolsets for integrating HPC clusters in experiment control systems

Manafov, Anar , Al-Turany, Mohammad, Klein, Dennis , Rybalchenko, Alexey

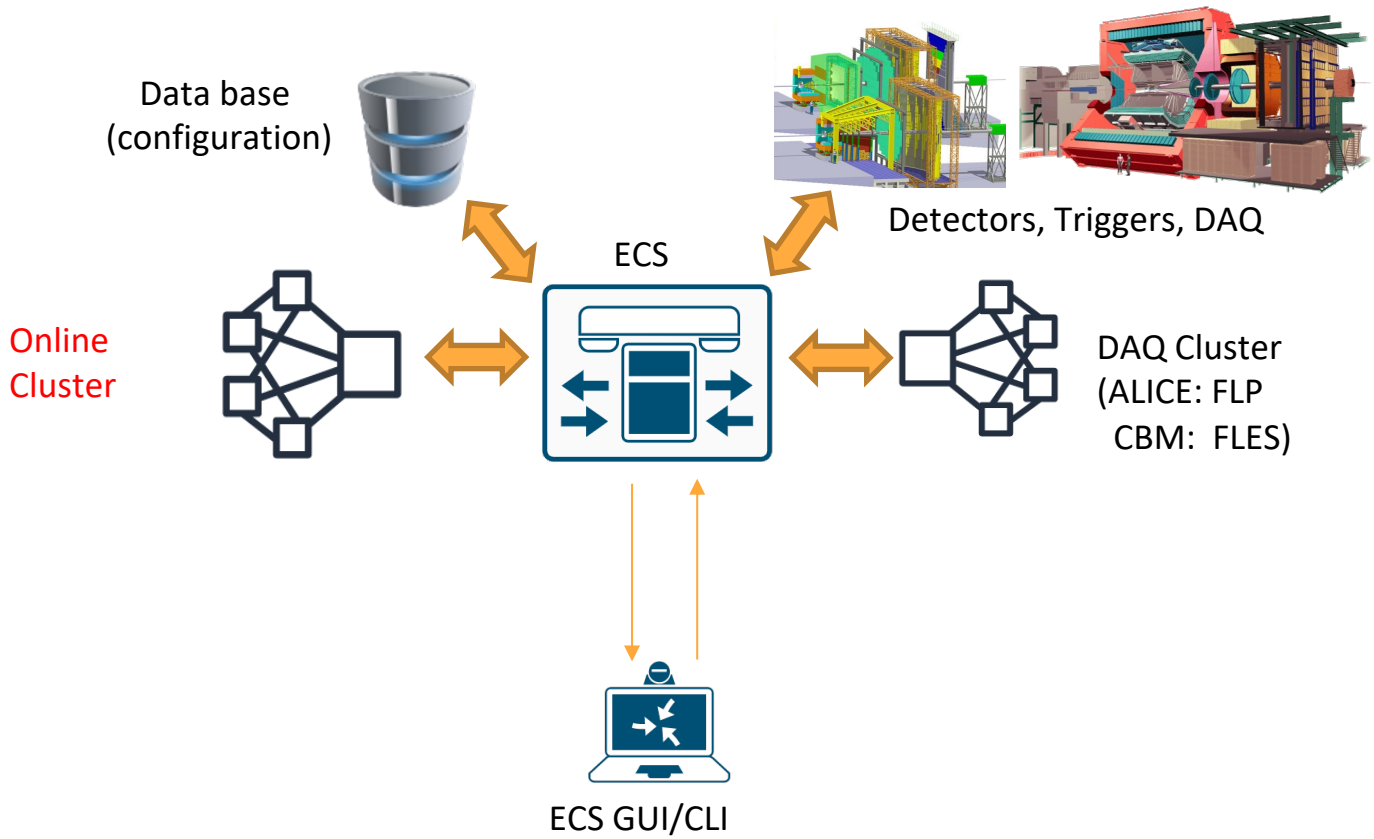
GSI Helmholtz Centre for Heavy Ion Research

Meanwhile HPC-Clusters are directly connected to the data acquisition systems and integrated into the online systems of the experiments

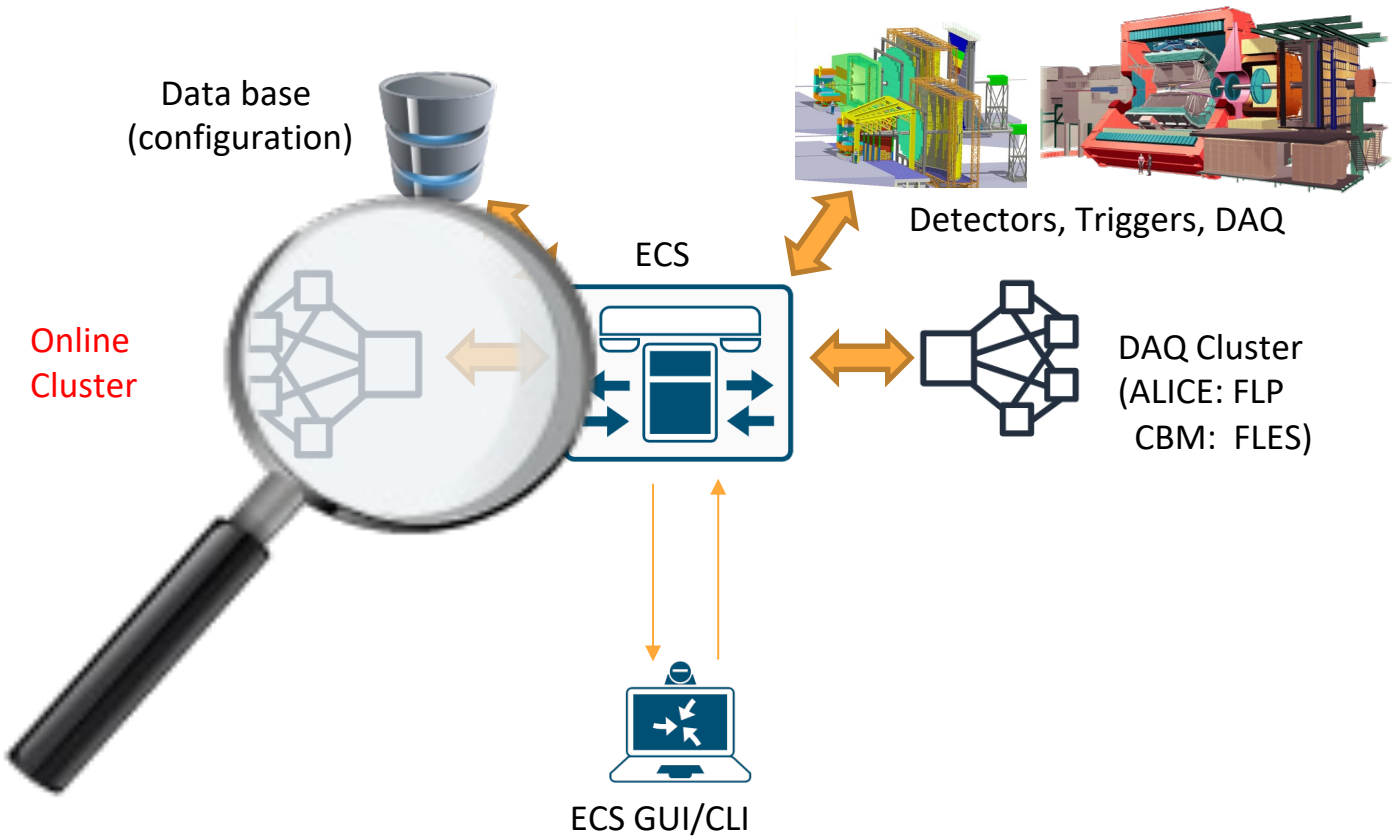




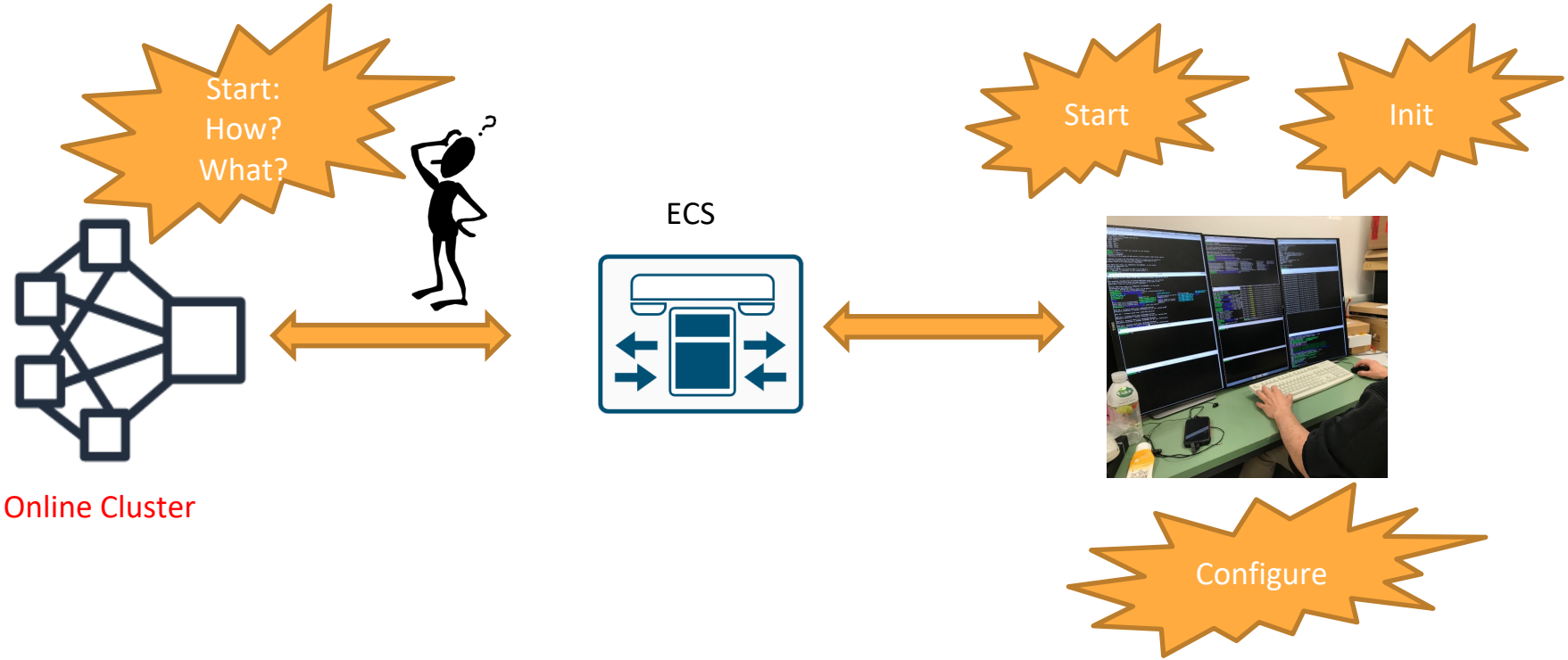
Typical Experiment Control System (ECS)



Typical Experiment Control System (ECS)



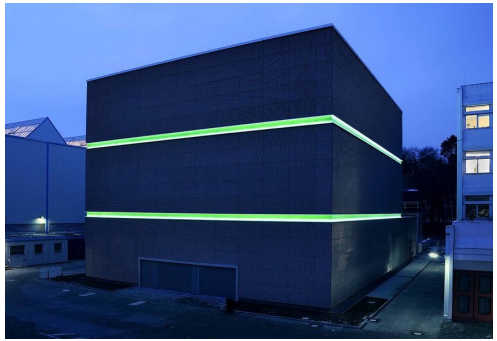
A compute cluster has to behave as one unit with a defined state



Online Clusters

- Dedicated Cluster (Alice)
  - Under the experiment control
  - Can have a resource management system (SLURM) but must not

- Generic Cluster with dedicated online partition (GSI)
  - Under IT control
  - Resource management system (SLURM)

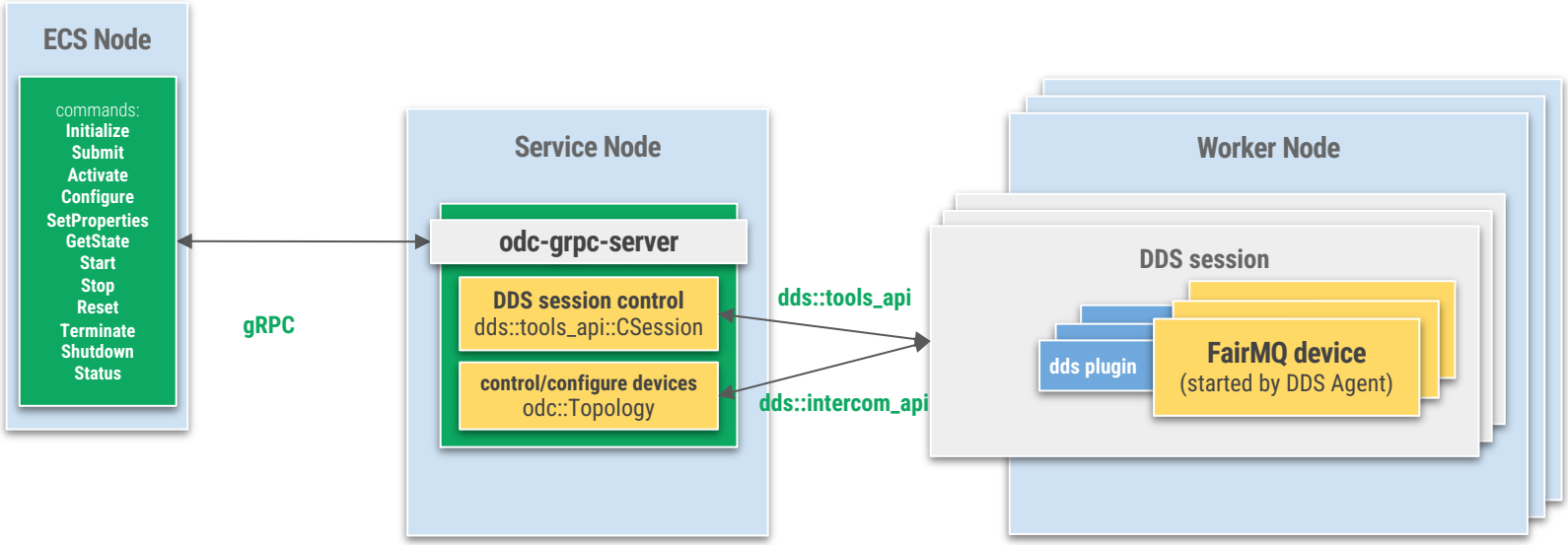


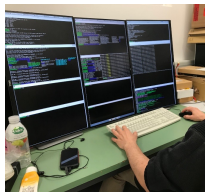
This talk is about ODC and DDS

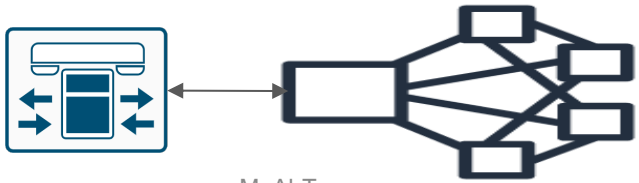
- DDS (Dynamic Deployment System) is a tool-set that automates and simplifies the deployment of user-defined processes and their dependencies using a predefined processing graph (topology)

- ODC (Online Device Control) is a tool to execute topologies of FairMQ devices via DDS, allowing control through ECS commands

Overview


ECS Node
commands:
Initialize
Submit
Activate
Configure
SetProperties
GetState
Start
Stop
Reset
Terminate
Shutdown
Status
gRPC
Service Node
odc-grpc-server
DDS session control
dds::tools_api::CSession
control/configure devices
odc::Topology
dds::tools_api
dds::intercom_api
Worker Node
DDS session
dds plugin
FairMQ device
(started by DDS Agent)





ODC comes with:

- A fixed set of commands is available to launch deployments with FairMQ topologies, control device states and configure device properties
- A single server process can handle multiple DDS/FairMQ sessions in parallel and reconnect to those which are running prior to server launch
- A simple command line server is also provided that can be used for testing without gRPC
- Resource management is delegated to the resource managers supported by DDS, including Slurm, custom SSH and localhost managers
- ODC provides a gRPC server component and a sample gRPC client

ECS Command Mapping:

Example

Command	Description
Initialize	Initialize DDS session
Submit	Submit DDS agents
Activate	Activate topology
Update*	Update topology (up or down scale and any other topo. change)
Configure	Device: InitDevice → CompleteInit → Bind → Connect → InitTask
Start	Device: Run
Stop	Device: Stop
Reset	Device: ResetTask → ResetDevice
Terminate	Device: End
Shutdown	Shutdown DDS session

*** The update command includes Reset, Activate and Configure commands**

New Features in ODC after the first run of ALICE Run3 (2022/23)

Support for **Slurm deployments** via a plugin that generates DDS configuration for Slurm:
-Task distribution to multiple Slurm partitions with individual config and different environment settings for each.
-Tasks and collections of tasks can now be allowed to fail, while maintaining homogeneous topology state. This can be done in two ways:

- **Marking individual tasks as expendable**, so that their failure can be ignored.
- Setting a **minimum number of error-free task collections**, where certain number of collections can be allowed to fail, without affecting a session (when the use case allows it).

Currently these checks are done **synchronously** - during state transition requests.
Asynchronous approach is in development.

New Features in ODC after the first run of ALICE Run3 (2022/23)

- **State & Status commands are processed asynchronously** for better feedback during ongoing commands.
- **Dynamic calculation of required resources** from the provided topology file:
  - In the addition to the existing manual resource request, user can now request that the resources are to be calculated based on the tasks in the topology definition file.
- Support for **core-based scheduling**, to allow for a more fine-grained task distribution on the available resources:
  - The core requirements are passed to DDS, which are then forwarded to the Slurm settings. The config can be done per task or task collection.
- Significant reduction in DDS command traffic between ODC and devices to reduce load, ease debugging and reduce log sizes.
- Extended debugging information on deployed devices and used resources.

DDS: Basic concepts

- A single responsibility principle command line tool-set and API;
- users' task is a black box – it can be an executable or a script;
- watchdogging;
- a topology-based execution of tasks;
- a plug-in system to abstract from RMS including SSH and a localhost plug-ins;
- doesn't require pre-installation and pre-configuration on the worker nodes;
- private facilities on demand with isolated sandboxes;
- key-value propagation and custom commands.

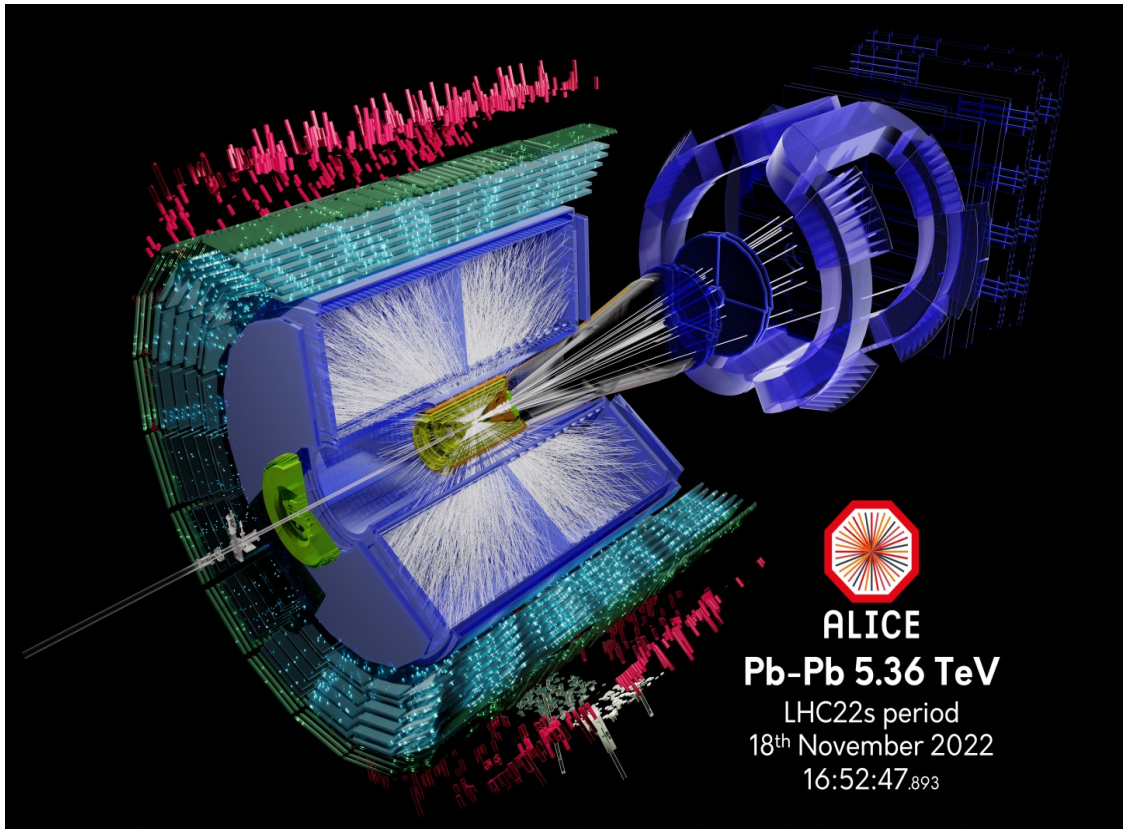
DDS highlights

- DDS works in sessions.
  - Each user session is isolated.
  - Users can have multiple sessions at time.
- DDS deploys lightweight agents to manage user tasks.
  - Users can run multiple agents per host with multiple tasks per agent.
- The 3rd gen. DDS deployment architecture can use only one agent for hundreds of user tasks.

ODC and DDS in RUN3

ODC and DDS is being used to deploy and control ~70 000 tasks on 200 nodes (2200 GPUs and few thousends of CPUs) in the online farm directly connected to the ALICE detector.

Thanks!



Summary

- ODC and DDS can be used on dedicated clusters as well as on generic ones provided the plugin for the RMS is available
- The system meanwhile in production by ALICE RUN3
- Tests where also successfully performed at the GSI Generic cluster

- Releases DDS v3.x : http://dds.gsi.de
- User's Manual & API doc.: http://dds.gsi.de/documentation.html
- Source Code:
  - https://github.com/FairRootGroup/ODC
  - https://github.com/FairRootGroup/DDS
  - https://github.com/FairRootGroup/DDS-user-manual
  - https://github.com/FairRootGroup/DDS-web-site