Evolution of the Resource Manager of the ATLAS Trigger and Data Acquisition System

Igor Alexandrov[1], Giuseppe Avolio[2], Andrei Kazarov[3], and Igor Soloviev[4]

[1]Joint Institute for Nuclear Research, Joliot-Curie 6, RU-141980 Dubna, Russia
[2]European Organization for Nuclear Research, CH-1211 Geneva 23, Switzerland
[3]University of Johannesburg, PO Box 524 Auckland Park 2006, South Africa
[4]University of California, Irvine, CA 92697, 949-824-5011, USA

MMCP, Yerevan, Armenia, 20-25 October, 2024

Outline

- Motivations and goals
- **Resource Manager(RM) Context**
- **Types of the RM resources**
- **RM resource scope**
- **Main use cases**
- **Resource Manager Architecture**
- **New RM requirements**
- **Resource Manager evolution trends**

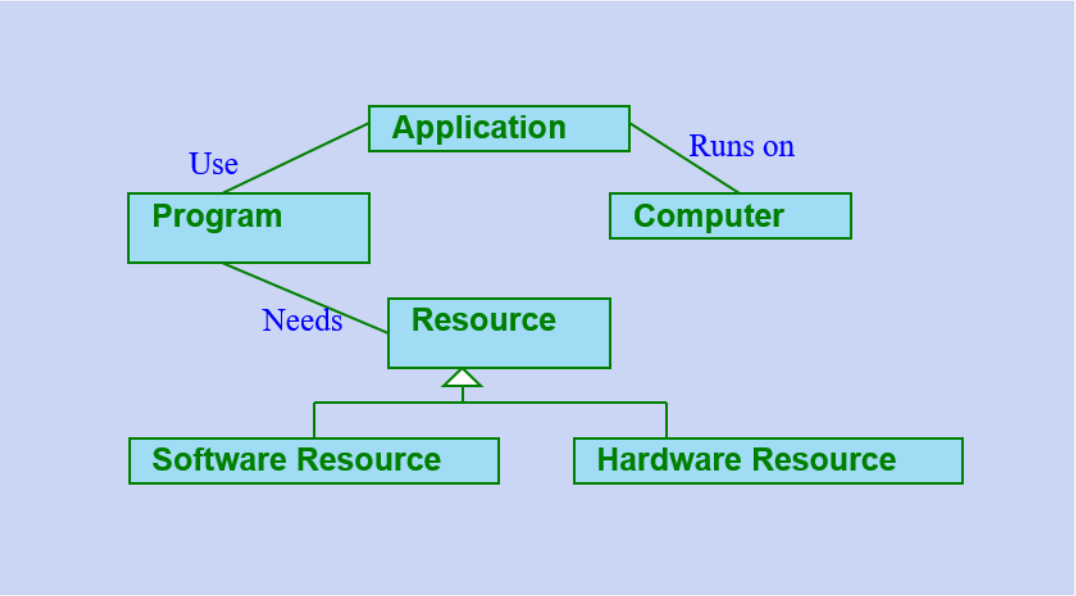
Motivations and goals

- The **Trigger and Data Acquisition** (TDAQ) system of the **ATLAS** detector at the **Large Hadron Collider** (LHC) at CERN is composed of a large number of distributed hardware and software components which provide the data-taking functionality of the overall system.
- The **ATLAS** experiment contains hardware and software resources which usage must be controlled to avoid conflicts during online operations.
- The **Resources Manager** (RM) is one of the services provided by the TDAQ. The purpose of the **RM** is to marshal multiple accesses to such resources.

Motivations and goals

- The Resource Manager marshals the rights for applications to access resources which may exist in multiple but limited copies in order to avoid conflicts due to program faults or operator errors.
- The access to resources is managed in a manner similar to what a lock manager would do in other software systems.
- The Resource Manager is queried about the availability of resources every time any application needs to be started.

Resource Manager Context



All the available resources that are used by the RM and their associations to software processes are **described** in the **configuration database**.

- Software resources
- Hardware resources

- The main clients of the RM are the DAQ online services allocating resources for all online processes during their start up.
- Some processes are managing resources themselves using RM API.

Types of the RM resources

- There are two types of RM resources in the ATLAS TDAQ configuration database:
  - **RM Software Resource** is used when only limited copies of some software may run in the ATLAS TDAQ system on **any** host.
  - **RM Hardware Resource** is used when only one application can start on the same host.
    - RM Hardware Resources are associated with a software object which has no association with a computer
      - greatly decreases the number of resources that should be stored in the database.
      - The RM creates whatever hardware resources are needed on the fly when an application starts.

RM resource scope

- The partition is a subset of the hardware and software components providing DAQ functionality
  - The ATLAS partition is one of the partitions combining all detector components
- The RM resources have total and partition scopes
- A resources can be allocated, if it is available in both total and partition scopes

Main use cases

Use case 1

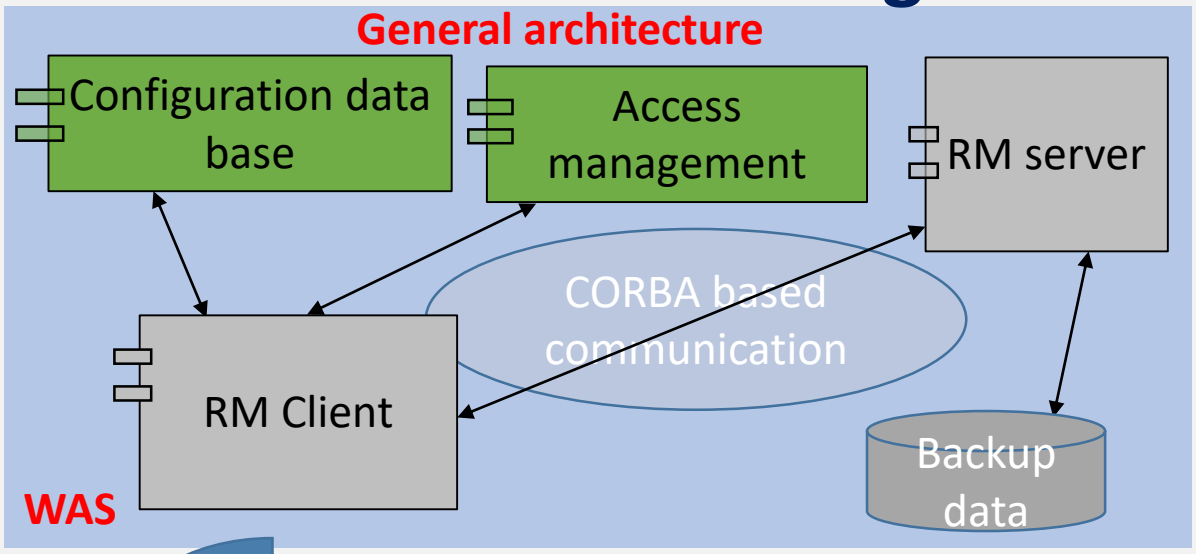
- When a process is going to be started, the DAQ service (PMG = Process Manager) asks the RM to allocate its resources first
- RM allocates these resources if all of them are available and returns positive handle to the PMG
- PMG starts process if resources are available (got positive handle)

- When process is started PMG sends to the RM the process ID (new requirement)
- PMG releases correspond resources when process is stopped
  - Use handle or process id and computer name

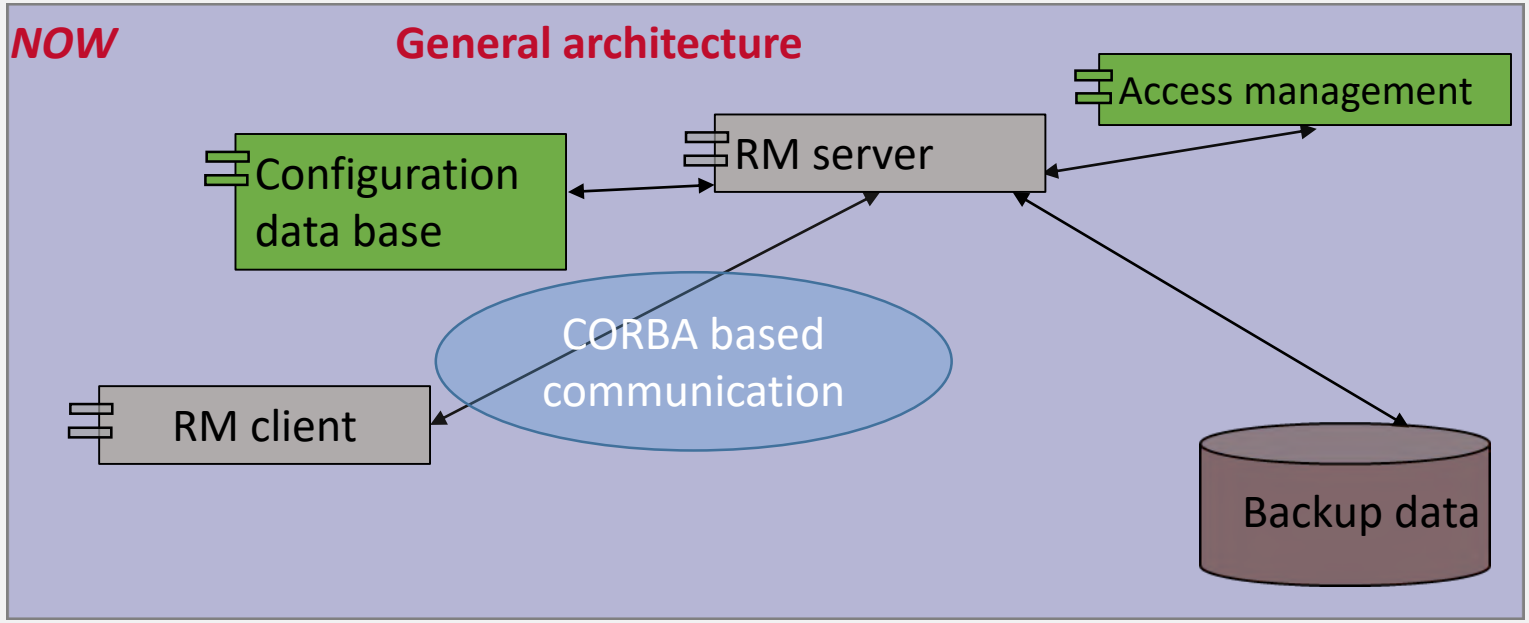
Use case 2

- Process asks the RM if its resources are available
- RM allocated resources if they are available
- Process free its resources before stop

Resource Manager Architecture



The architecture has evolved with DAQ operations needs.



New RM requirements

- The requirements for the Resource Manager were updated several times during Run 2 and following long shutdown.
  - Recovery rules if client-server communication fail
  - Server startup without automatic configuration loading
  - Use DAQ tokens service for client authentication and the Access Manager services for client authorization
  - Move configuration load from client to server

Resource Manager evolution trends

- The RM service review was initiated by the operations experience and changes in the underlying DAQ system
  - Code optimization, remove unused parameters and methods
    - clean IDL unused parameters and methods
    - separate client part interface and implementation part, remove unused methods
  - Improve RM server info messages, use appropriate severity of the Error Reporting System (waring, error, fatal)
- Use modern C++ 17 standards, replace boost classes by standard C++ 17 library classes
- Improve error reporting using nested exceptions (based on standard DAQ error reporting facilities)
- Replace obsolete classes from TDAQ packages by standard C++ ones if available
- Expand and detail functional tests
- Clarify new requirements and ensure their implementation

Conclusion

- Motivation of development, use cases, context, types and scope of resources and architecture of the **Resource Manager** service was introduced
- Evolution of the Resource manager shown
  - The architecture has evolved with DAQ operations needs
  - New requirements were satisfied
  - The RM service has been updated based on the results of the review
    - Code optimization, remove unused parameters and methods, use modern C++ 17 standards
    - Improve error reporting using nested exceptions
    - Expand functional tests

Thank you for attention!