

Pilot Software in the SPD Online Filter

Romanychev Leonid
JINR MLIT
romanychev@jinr.ru

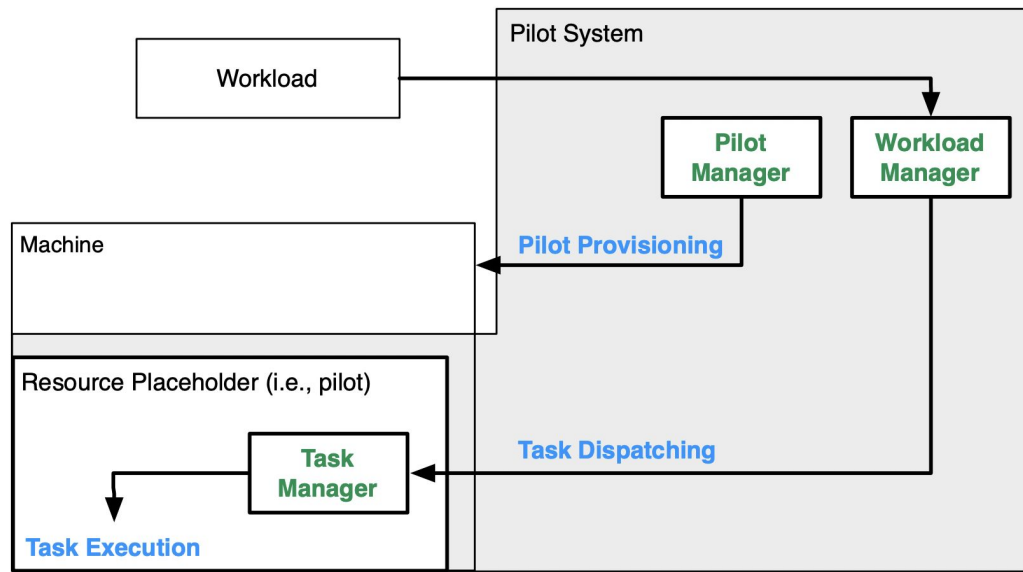
JINR AYSS Conference "Alushta-2025"
11.06.2025

Introduction: Role of Pilot Applications

- Provide a flexible mechanism for execution of computational tasks.
- Widely used in high-throughput computing (HTC) systems for scientific data processing (ex. LHC computing).
- Issue: lack of unified abstraction and best practices leads to a variety of implementations.

Core Components of Pilot Software

- **Pilot Manager:** Launches pilots (resource placeholders) on computing resource; Interfaces with SLURM, HTCondor, etc.
- **Workload Manager:** Organizes task queue (dependencies, priorities, resource readiness).
- **Task Manager:** Executes tasks on pilot-reserved resources; Manages task lifecycle (launch, restart, monitor, error handling).



Functionality & Architecture

Functional Stages:

- Provisioning (Acquiring & deploying resources)
- Dispatching (Assigning tasks to pilots)
- Execution (Running tasks on resources)

Architectural Features:

- Multi-level scheduling
- Communication Models (Master-worker, Broker-oriented)
- Flexibility (integrates with various DCRs: clusters, grids, clouds)

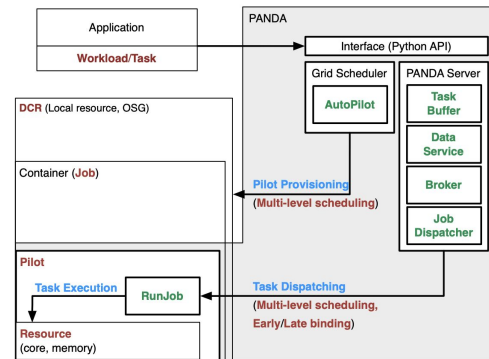
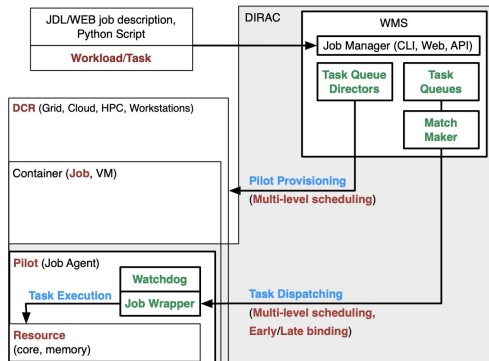
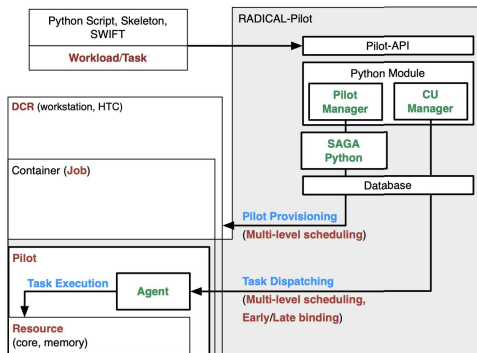
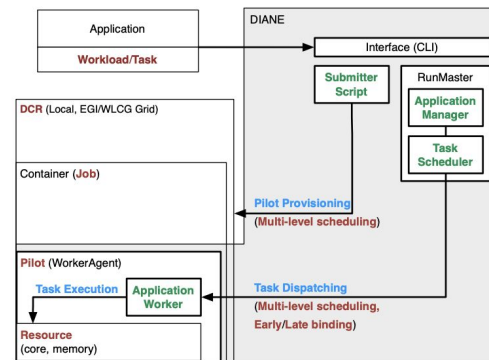
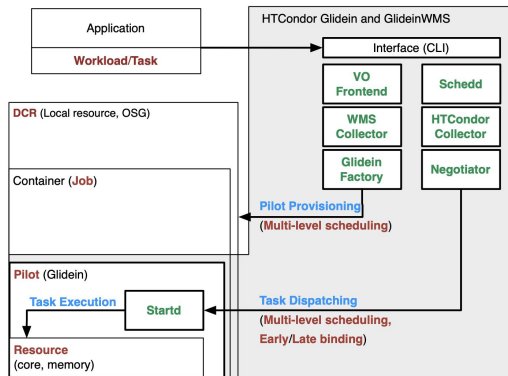
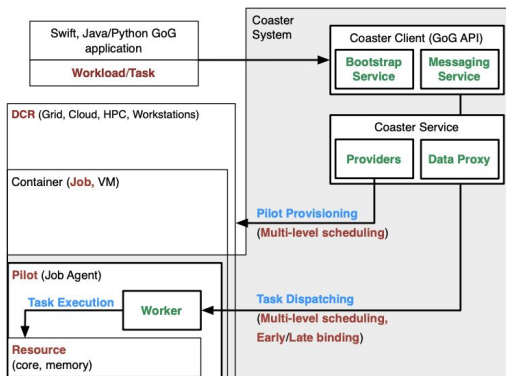
Late Binding Mechanism

Definition:

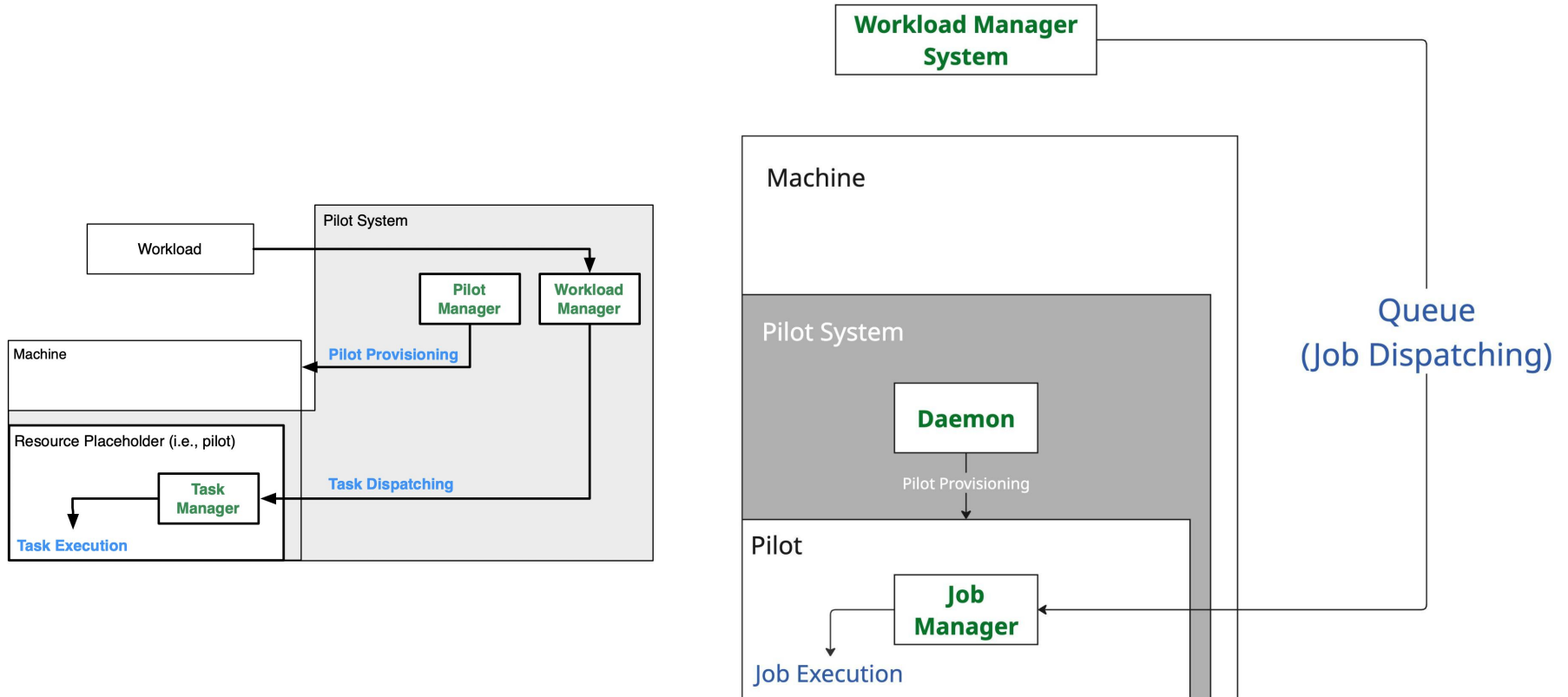
Late binding is the process of assigning tasks to active pilots at the moment of availability, unlike early binding, where tasks are tied to inactive pilots.

Benefits:

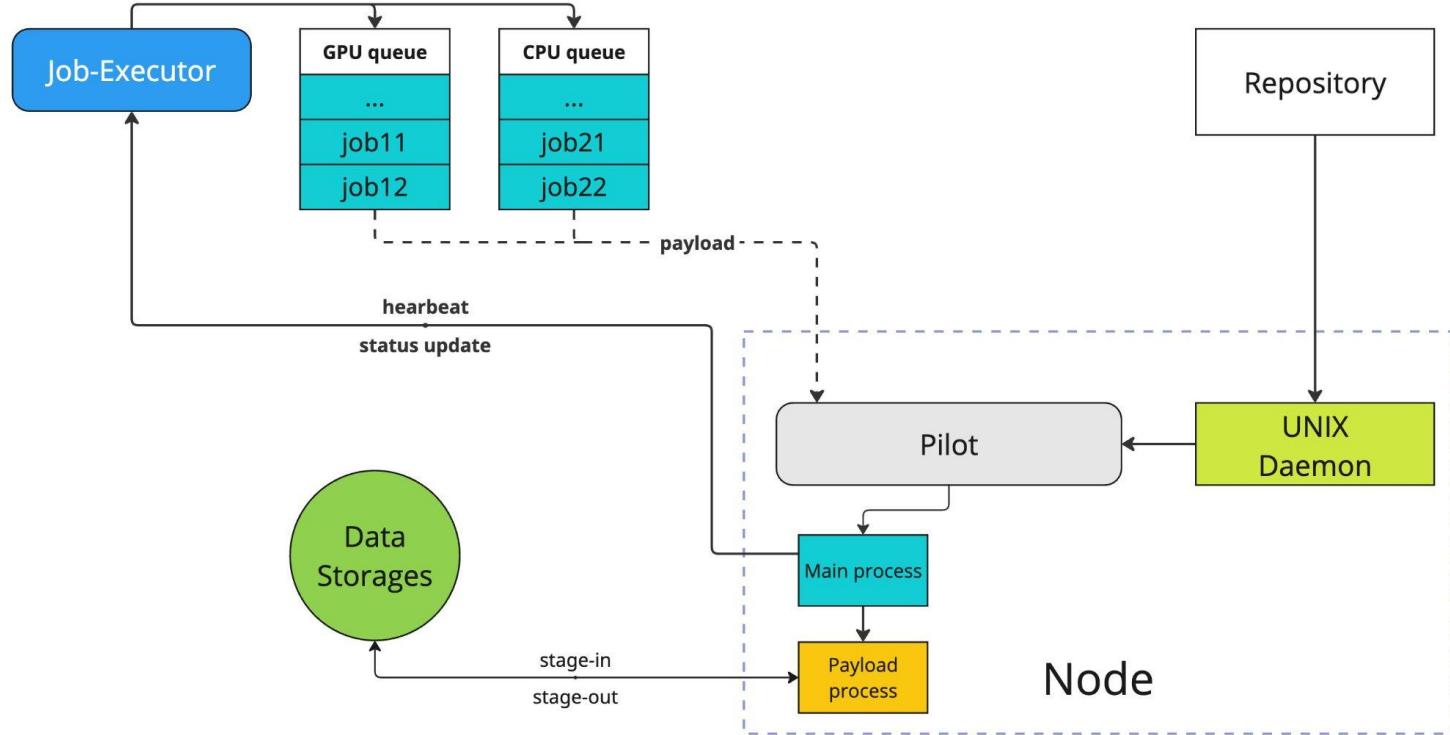
- Dynamic task allocation improves resource utilization efficiency.
- Reduces queue wait times, critical for high-performance systems.
- Enables high throughput (e.g., up to 1 million tasks per day for ATLAS).



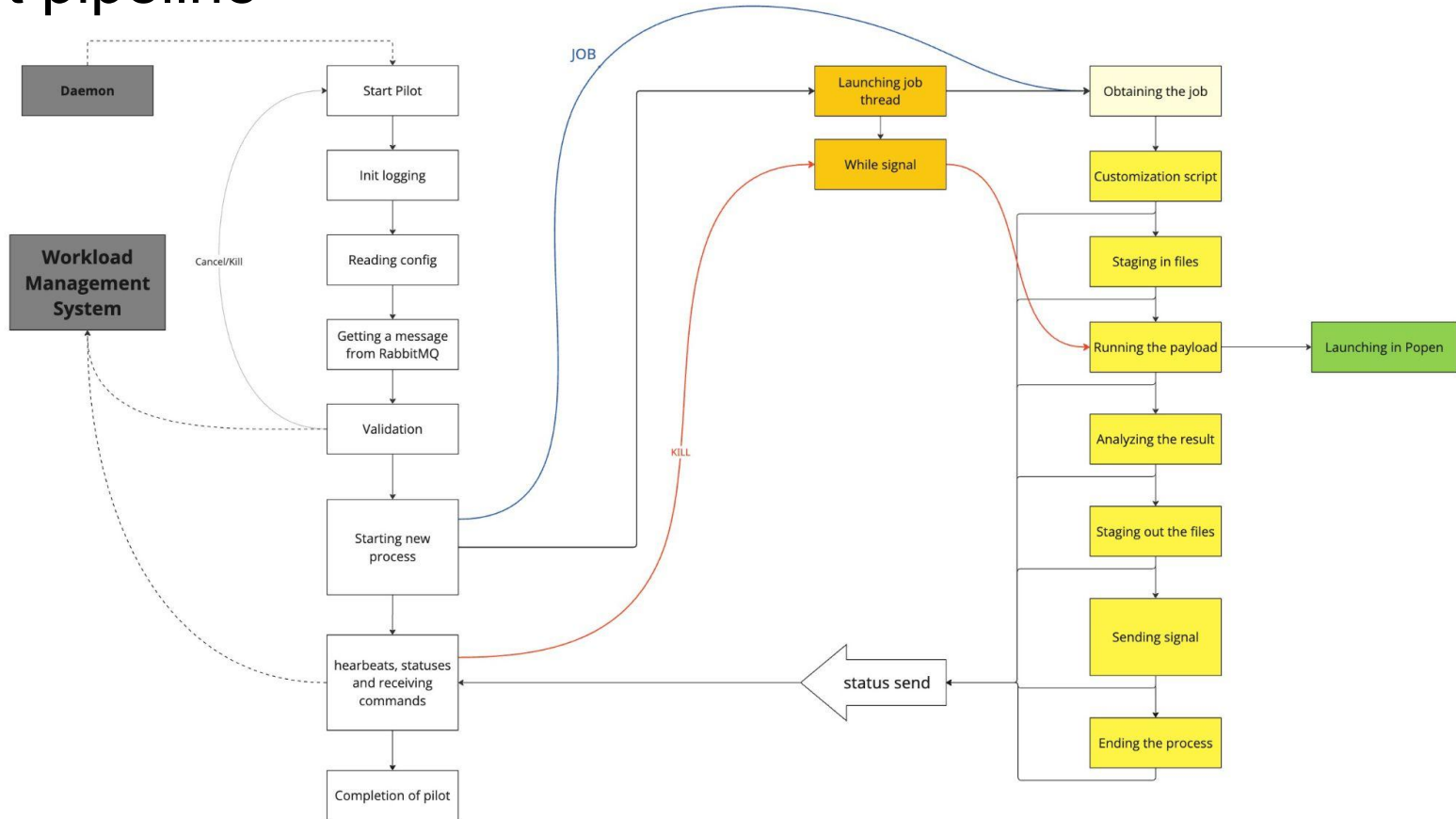
SPD Online Filter Pilot Software



SPD Online Filter Pilot Software



Pilot pipeline



“DAQ emulator”

1. Using SPD DAQ emulator, we've generated 50 files, each ~2Gb;
2. Input dataset has been registered with these files;
3. Task has been processed (or 50 jobs);
4. The payload for Pilot is simple: compute the MD5/BLAKE3 hash, as there is no actual computation involved at this stage.;
5. Generation of one files takes around ~7 min, using JINR Cloud VM: 12x 1-core Intel Xeon E5-2650
6. Registration of the entire dataset: ~10 sec

```
# Configuration file for SPD DAQ data generator  
# 2023/03/01
```

```
#Data file name format: run-<run number>-<chunk  
number>-<builder id>.spd  
DataFileNameFormat = run-%06u-%05u-%02u.spd
```

```
#RND generator seed:  
RandomSeed = 12345
```

```
#The size limit of the output data file in bytes:  
DataFileSizeLimit = 2147483648
```

```
#debug mode for debugging front-end card. If it is 1  
then generator will  
#produce all data words (headers and trailers) even  
if there are no hits,  
#otherwise all empty data blocks are removing  
DebugMode = 0
```

```
#Source ID(s) of the clock module(s) for  
measurement start of frame time:  
FrameClockID = 1000,1001
```

```
#Source ID(s) of the TDC module(s) for measurement  
of the bunch crossing time:  
BunchCrossingID = 1004
```

```
#Slice length in ns (must be less than smallest TDC  
over-roll time (4.5 ms for RS)):  
SliceLength = 10000
```

```
#Number of slices in a frame:  
FrameLength = 100000
```

Conclusion

- Unified task execution interface
- Adaptability across platforms
- Reduced overhead (scheduling & execution)
- Scalability (to millions of concurrent tasks)

Thanks for your attention