Event Streaming Service for ATLAS Event Processing

CHEP 2019(24th International Conference on Computing in High Energy & Nuclear Physics)

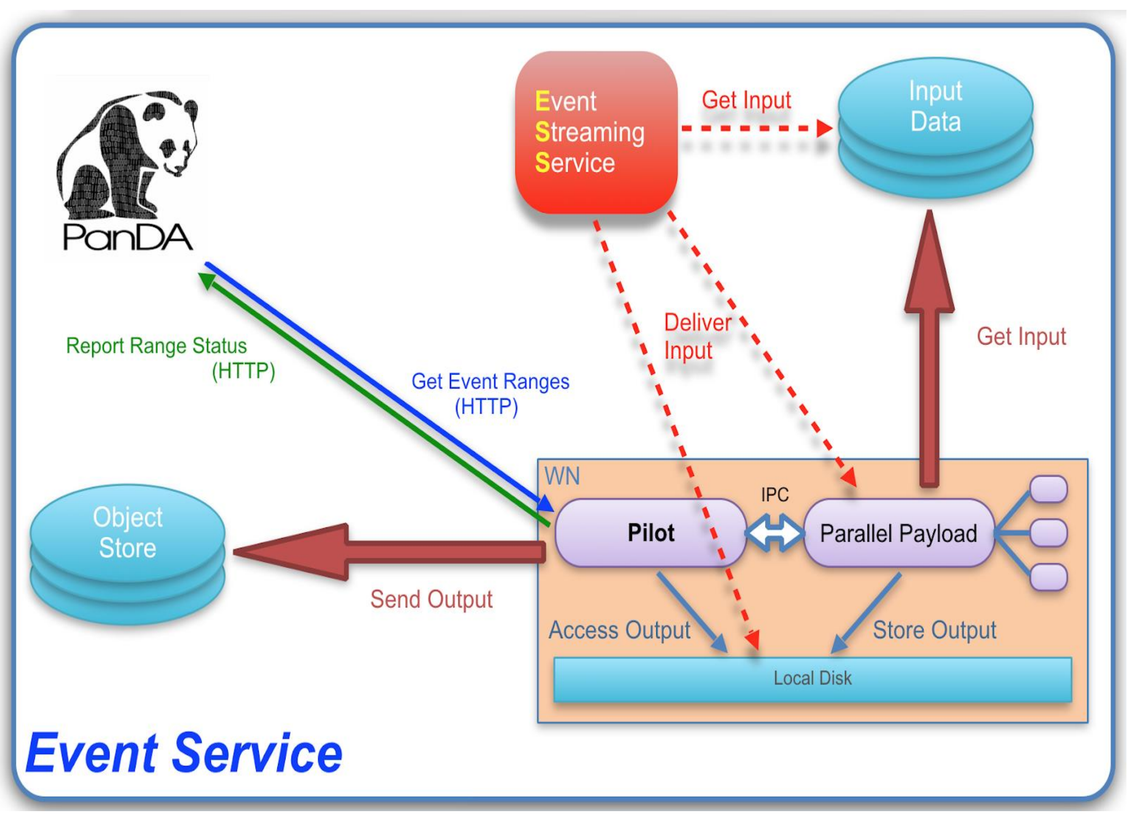
Wen Guan[1], Tadashi Maeno[2], Gancho Dimitrov[3], Brian Paul Bockelman[4], Torre Wenaus[2], Vakhtang Tsulaia[5], Nicolo Magini[6]

[1]University of Wisconsin-Madison; [2]Brookhaven National Laboratory (BNL); [3]European Laboratory for Particle Physics, CERN; [4]University of Nebraska Lincoln,; [5]Lawrence Berkeley National Laboratory; [6]Iowa State University

ATLAS Event Streaming Service & iDDS

- **The ATLAS Event Streaming Service (ESS) is an approach to preprocess and deliver data for Event Service (ES) that has implemented a fine-grained approach for ATLAS event processing.**
- **The ESS allows one to asynchronously deliver only the input events required by ES processing, with the aim to decrease data traffic over WAN and improve overall data processing throughput.**
- **iDDS (intelligent Data Delivery Service)**
  - **A new service under development to serve Event Streaming Service and beyond.**
  - **Intelligent to transform and deliver needed data to consumers**
  - **Orchestration of WFMS(Workflow Management Service) and DDM(Data Management Service)**
  - **Beyond**
    - **On-demand production of analysis format data**
    - **Fine grained tape carousel**
    - **etc**

Event Streaming Service

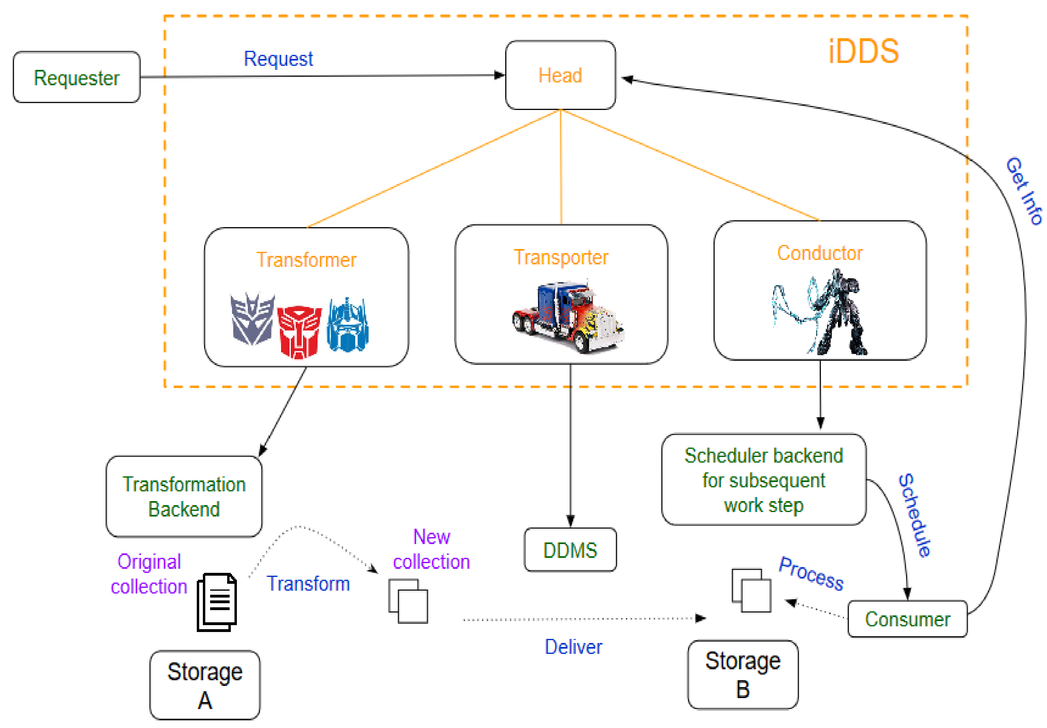


- **Event Streaming Service Requirements:**
  - **Intelligently deliver fine-grained input events to Event Service based on the payload requirements.**
  - **Trigger to get input data with DDM**
  - **Transform input data to fine-grained events**
  - **Trigger to activate jobs in WFMS**
  - **Integrate with PanDA Pilot.**

ESS -> iDDS

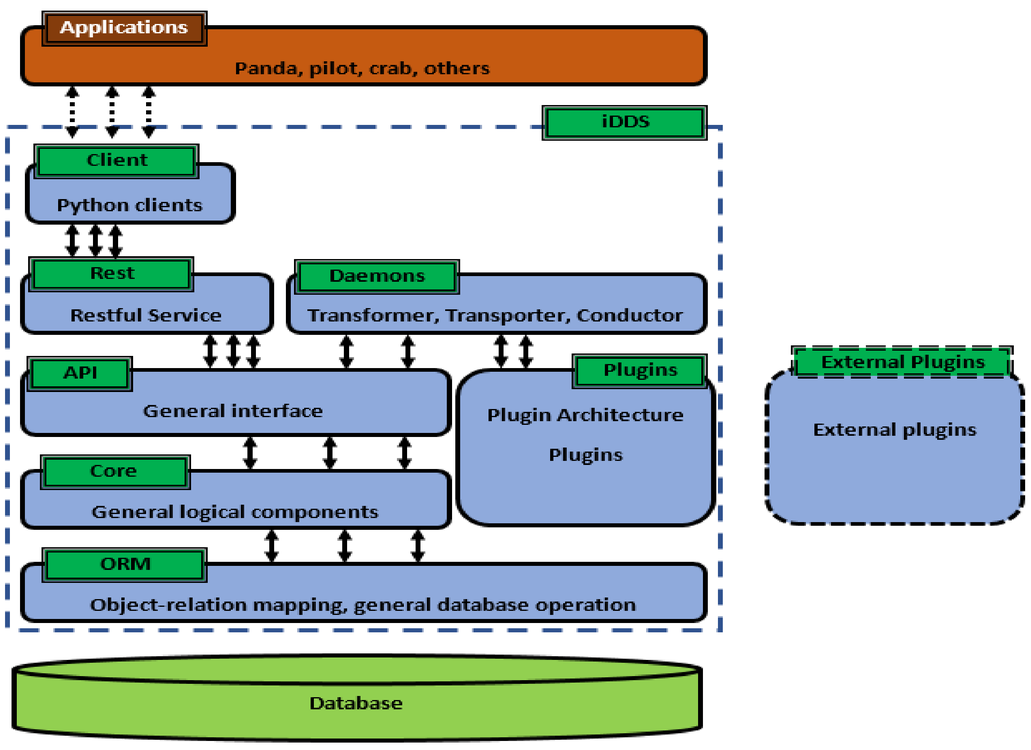
- **Beyond ESS, similar new workflows emerges**
  - **Remote data transformation/reduction**
  - **On-demand production of analysis format data**
  - **Fine grained tape carousel**
  - **etc**
- **Orchestration of WFMS and DDMS with generalized workflows.**
  - **Because of accumulating requirements, functions for orchestration have been added to existing services without generalization.**
  - **Extraction and abstraction of these functions**
- **These two requirements trigger us to think of a new service: iDDS**
  - **iDDS is also designed with plugin architecture to support more new workflows.**

Intelligent Data Delivery Service



- **HEAD: Restful interface**
  - **To register and query requests**
  - **To provide catalog service for consumers to get/list required collections and contents(files or events).**
- **Transformer**
  - **To transform input data from one collection to another collection**
  - **With plugin structure to support different transform type and different backends.**
    - **For ESS, the transformer plugin is splitting**
- **Transporter**
  - **To manage collections with DDM backends**
- **Conductor**
  - **To notify/schedule consumers to consume new transformed data in a fine-grained granularity**
- **Others**
  - **Client: To communicate with iDDS Restful service.**

iDDS Architecture



- **Abstract Layers**
  - **Hide the complexity of different logics to implement general API.**
- **Plugin Architecture**
  - **To support different transform types**
  - **To support different backends**

Future and Plans

- **Primitive version (~ early 2020)**
  - **The requester defines everything, such as workflow and input in addition to granularity and destination of output data collection and how the subsequent step is triggered**
- **Advanced version (~ by the end of 2020)**
  - **The requester defines only workflow and input, while iDDS dynamically optimizes granularity and destination of output collection and triggers the subsequent step by using own decision making engine**