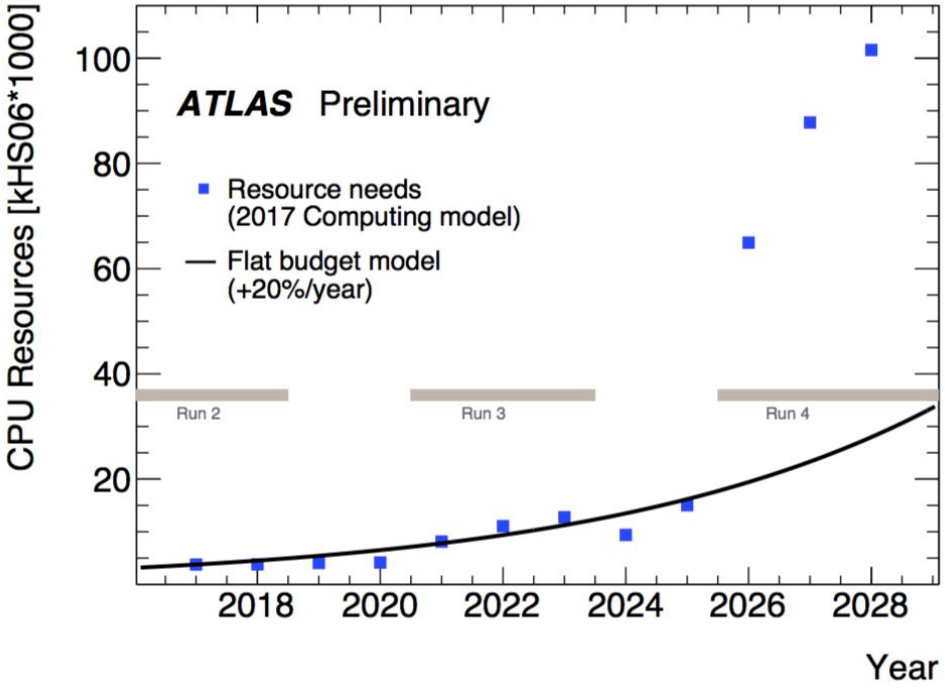
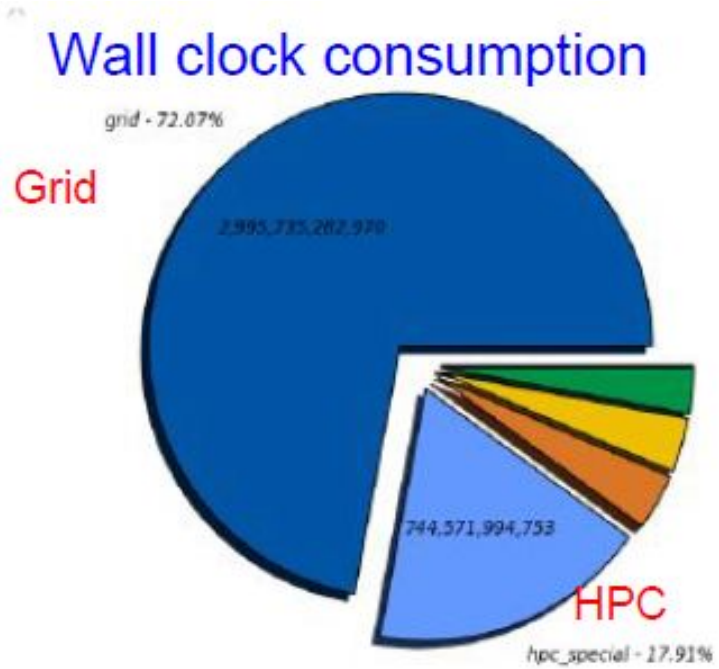
Towards an Event Streaming Service for ATLAS data processing

Nicolò Magini
on behalf of the ATLAS Collaboration



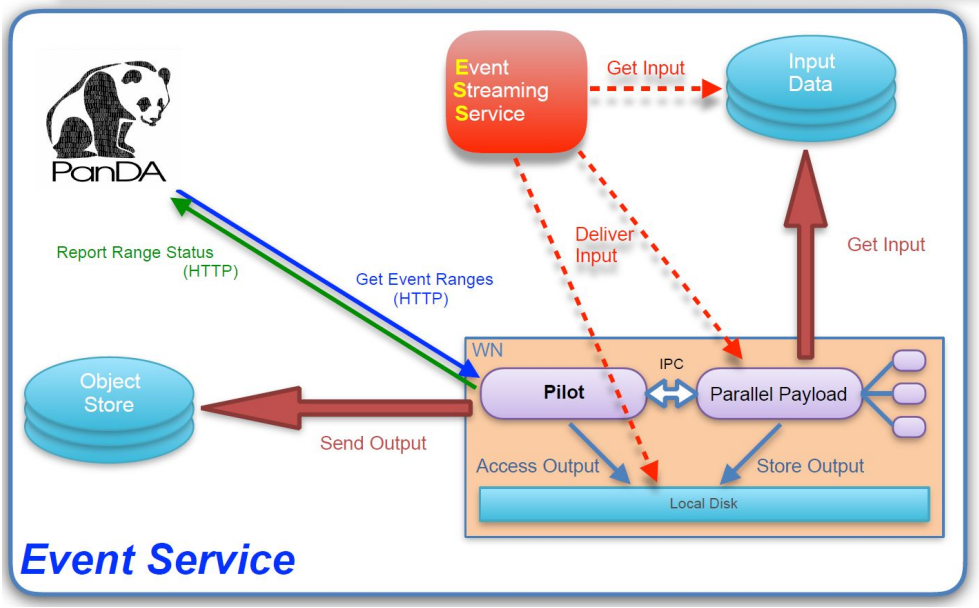


As resource needs are going to rise rapidly…

… we are expanding to a diverse set of extra resources

Event-level processing

- **ATLAS Event Service (AES)** deployed to exploit diverse resources that can disappear on short notice
- Assigns event-level granularity workloads to running application
- Streams outputs away almost continuously



- See [Poster CHEP2018-155]

Event Streaming Service

- **ATLAS Event Service** currently requires the whole input file to be accessible (locally or over WAN)

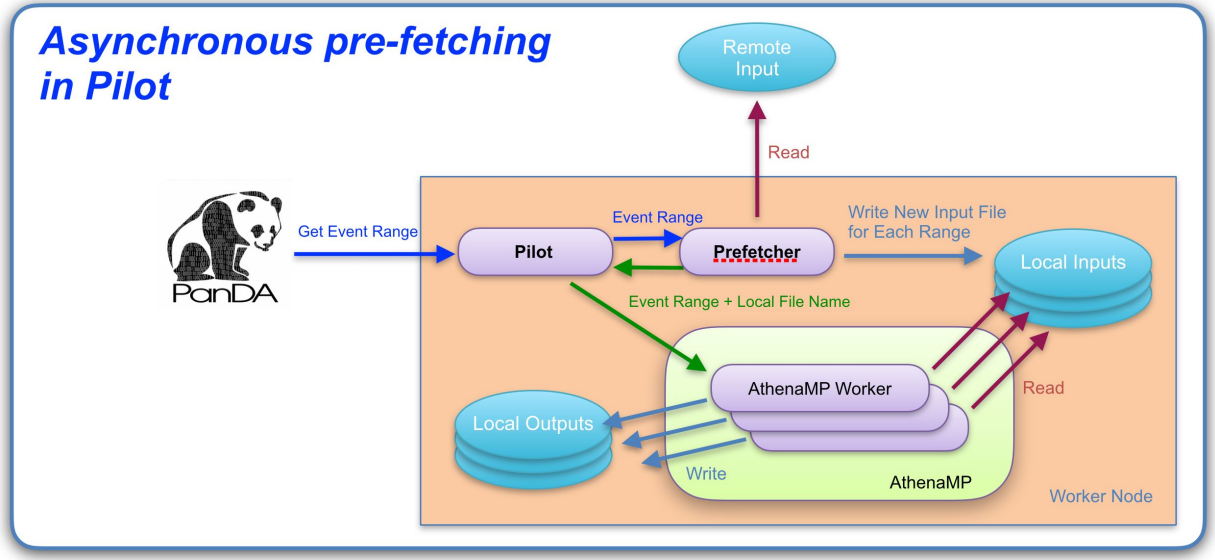
- **Event Streaming Service (ESS)**: deliver input data to compute nodes over the WAN at fine granularity
  - e.g. only the event ranges that will be processed, or even only specific objects within the events

- Reduce the need for local storage and minimize the impact of high WAN latency

ESSv1 prototype

- Implemented as a **Prefetcher** process started by the pilot on the worker node in parallel to the main AthenaMP application

- Reads the large remote input file, and duplicates each event range to a small local input file



Testing remote access with Prefetcher

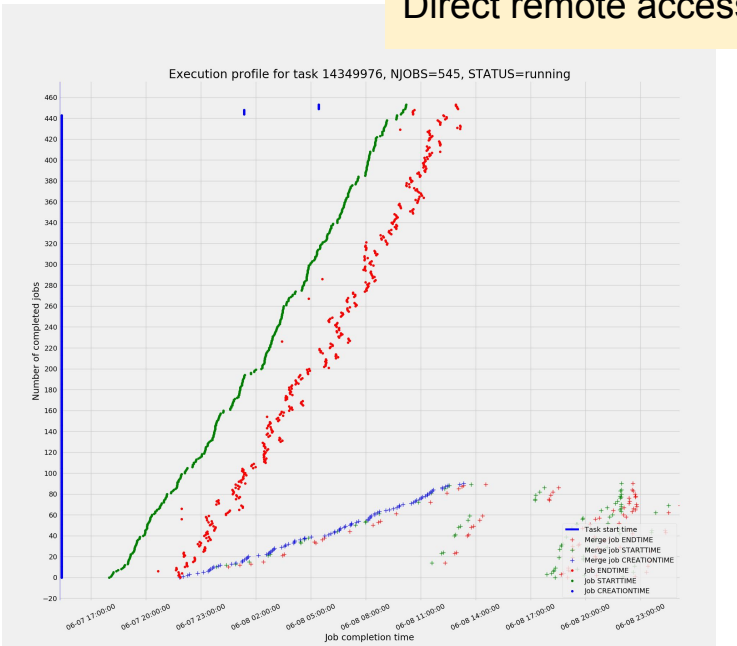
- Prefetcher used to study feasibility of remote streaming of input events
- Started validation with G4 simulation tasks
  - Not I/O bound, so not the main use case for ESS in the long run - but it is a tried-and-tested application for comparison with “standard” AES
- Enabled remote access in ATLAS production system, discovering a few issues affecting remote I/O
  - Occasionally stuck/slow opening files over WAN
    - Can work around restarting the application
  - Repeated failures if the chosen remote replica is corrupted
    - Planning to use Rucio redirector for automated fallback

Prefetcher commissioning

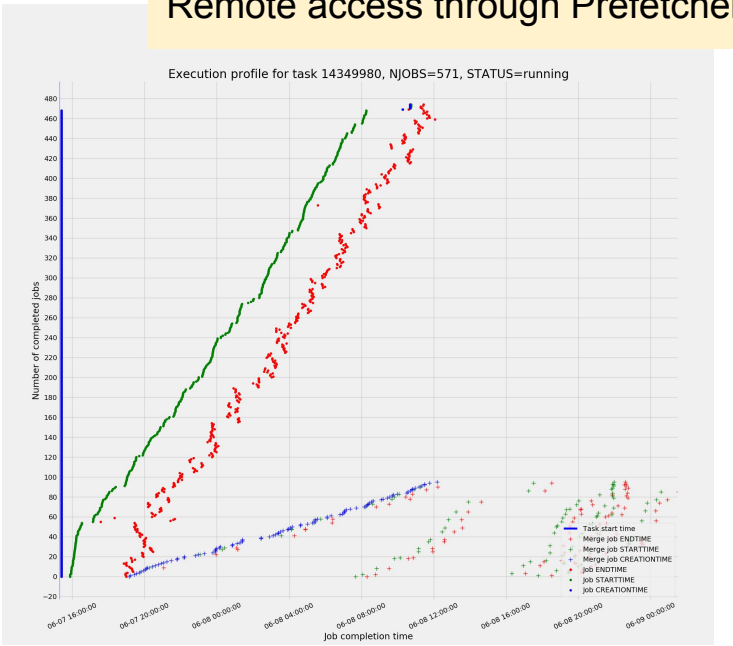
- Measuring performance of AES tasks doing direct remote access vs remote access through Prefetcher
- Created 100k event validation tasks and ran them on selected sites
  - e.g. read input from BNL or NL T1s, run on Great Lakes T2
- Compared metrics for task completion time, success rate, etc.

Prefetcher task performance

Direct remote access



Remote access through Prefetcher

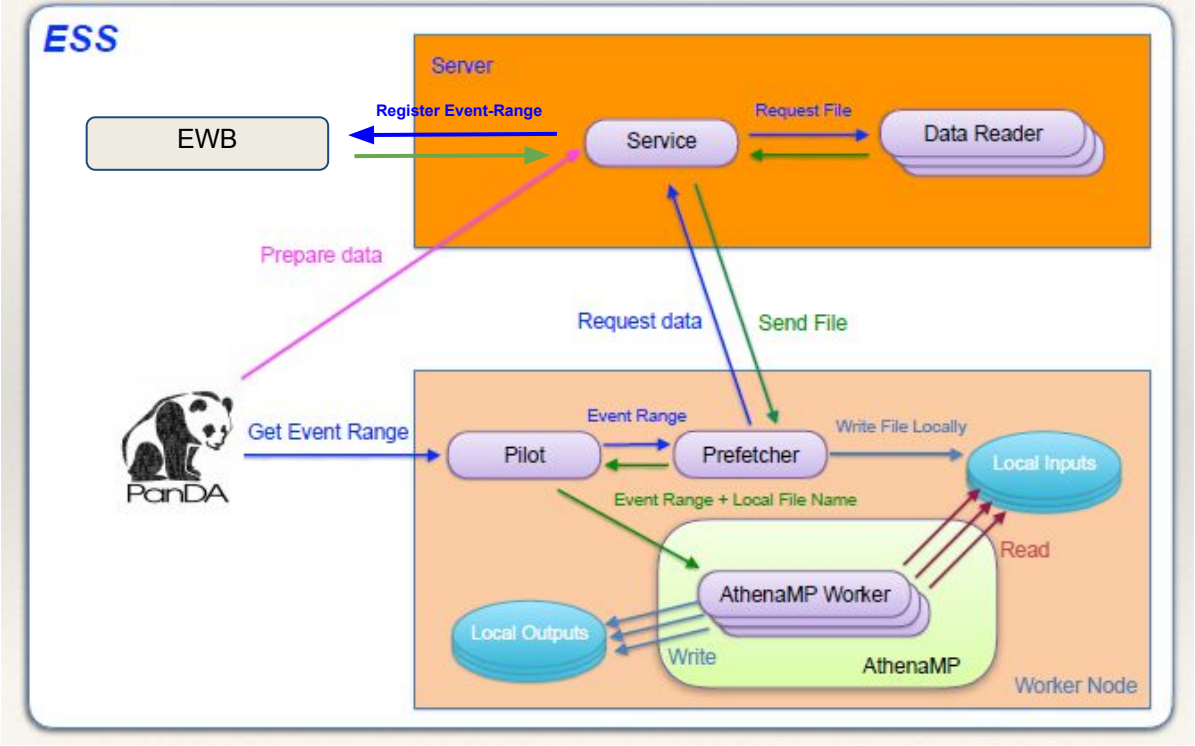


- No significant difference observed in task running time (not expected for SIM)

Next steps for ESSv1

- Extend Prefetcher to I/O-heavy workflows that could benefit more from ESS, e.g. AOD Derivation
  - Many inputs, potentially distributed, expensive in (re)brokering
  - Instrumenting production system to support AES/ESS for Derivation
- Optimize number of events to prefetch in the background
- Experiment with caching of prefetched events
  - e.g. with XCache

ESSv2



- ESS becomes a server close to the data
  - prepares the event-range files to be delivered to the client, possibly filtering the event contents
  - records them in a DB for bookkeeping
- Prefetcher process in pilot becomes a client asking for event-range files from the Server

Bookkeeping for ESS

- Client will need to be able to discover which event-range files are available and where; eventually also which objects are contained in the files

- Depending on the granularity of the event-range files and their lifetime, ESS could produce O(10 Hz) files and store up to O(100M) of them

- Working to implement efficient bookkeeping for these transient data in Event WhiteBoard - a new ATLAS service to track metadata for event collections

Conclusions

- Developed a working ESS prototype
- Along the way, improved support for remote access in ATLAS production system
- Measuring the ESS performance and planning to use it also for other workflows i.e. Derivation
- Started design of ESSv2

Backup Slides

Prefetcher commissioning

Prefetcher over WAN				
Task ID	Done evts	Failed jobs	Completion time - h	Completion time (no queue) - h
12858157	100%	70	6:00	4:30
12868946	100%	56	8:30	7:45
12944727	100%	45	13:45	10:45
12959866	98%	196	9:00	8:30
13048011	100%	22	6:45	6:45

Prefetcher commissioning

Direct access over WAN				
Task ID	Done evts	Failed jobs	Completion time - h	Completion time (no queue) - h
12858152	100%	36	4:15	4:15
12868956	100%	8	8:00	8:00
12944729	51%	3091	16:06	7:00
13048014	100%	3	7:13	5:15
Default (Rucio transfer + copy-to-scratch)				
Task ID	Done evts	Failed jobs	Completion time - h	Completion time (no queue) - h
12935663	100%	5	7:00	5:00