A Persistent Back-End for the ATLAS Online Information Service (P-BEAST)

Author:

SICOE, Alexandru Dan
(Marie Curie ACEOLE Fellow, CERN,
asicoe@cern.ch, tel: +41 2276 71113)

Contributors:

LEHMANN MIOTTO, Giovanna (CERN)
KOLOS, Serguei (University of California Irvine (US))
MAGNONI, Luca (CERN)
SOLOVIEV, Igor (University of California Irvine (US))

1.Introduction

ATLAS is the largest of several detectors built along the Large Hadron Collider at CERN. Its aim is to measure particle production when protons collide at a very high center of mass energy, thus reproducing the behavior of matter a few instants after the Big Bang. The detecting techniques used for this purpose are very sophisticated and the amount of digitized data created by the sensing elements requires a very large trigger and data acquisition system (TDAQ[1]). This consists of approximately 30.000 applications running on 2000 interconnected computers.

There are several sub-systems responsible for facilitating information exchange between these applications and for monitoring their health. One of these is called the Information Service[1] (IS). It consists of a multitude of server applications running on dedicated machines. Any TDAQ application can be an IS client and can publish information objects of various types or it can subscribe to receive information objects from a specified source. The publishing rates vary widely and give a bursty nature to the traffic that IS is capable of generating.

During normal operation the rates have relatively steady levels. Peaks in the rates appear however when many applications publish data at the same time. This can happen when the state of the ATLAS infrastructure changes. For example, during a starting transition a lot of applications come alive and as soon as they do, they start publishing information about themselves.

2.What is P-BEAST?

P-BEAST wants to offer persistency to a large part of the information published in IS, denoted by the term "Operational Information". IS already has a mechanism that buffers a certain amount of values in memory but this is not sufficient for offline data analysis. What is needed is a system that stores the time series data on disk such that it can be retrieved at any point by data flow experts who will visualize it with the help of specialized dashboards. Such functionality is useful for:

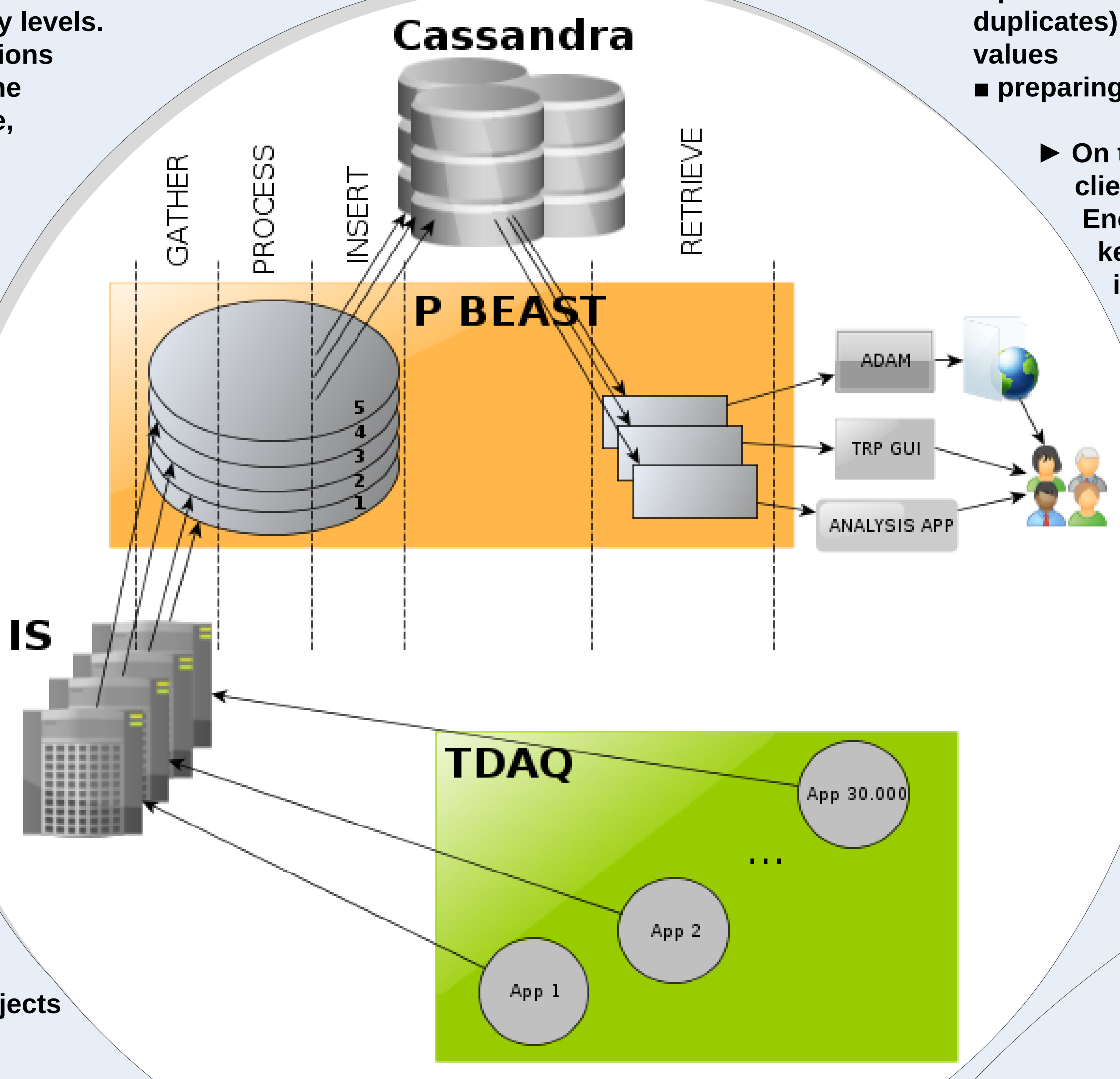
- **understanding short/long term past behavior of different components of the ATLAS TDAQ**
- **comparing between physics data taking sessions of the detector**
- **investigating problems that occurred during a certain data taking session**

The project has thus two major parts which are reflected in its architecture.

► The insertion path involves:

- **gathering the required information by subscribing to IS and receiving callbacks whenever an information object is created or updated by the source application**
- **processing the information by applying configurable filters (smoothing, duplicates) to reduce unnecessary storage of unimportant or repeating values**
- **preparing the accepted values for insertion in a database**

► On the retrieval side, a programmatic API shall be offered to any client application that wants to access the raw stored data. Enough metadata will be accessible in order for the clients to keep track of changes made in the structuring of IS information in time. A special type of client will be a driver that implements a general retrieval protocol on top of HTTP. Supporting this will allow the data stored in P-BEAST to feed into a web based visualization tool that displays data from several different sources of information within the ATLAS TDAQ (ADAM).



3.Why Cassandra?

The database technology of choice is a key-value distributed storage system called Cassandra. The main reasons for adoption of this technology are:

► built to sustain massive insertion data rates presented in an irregular fashion.

► within a top level logical partitioning of data (column family) Cassandra is schemaless which means that the stored data can follow the evolution of IS information objects over time in a seamless fashion

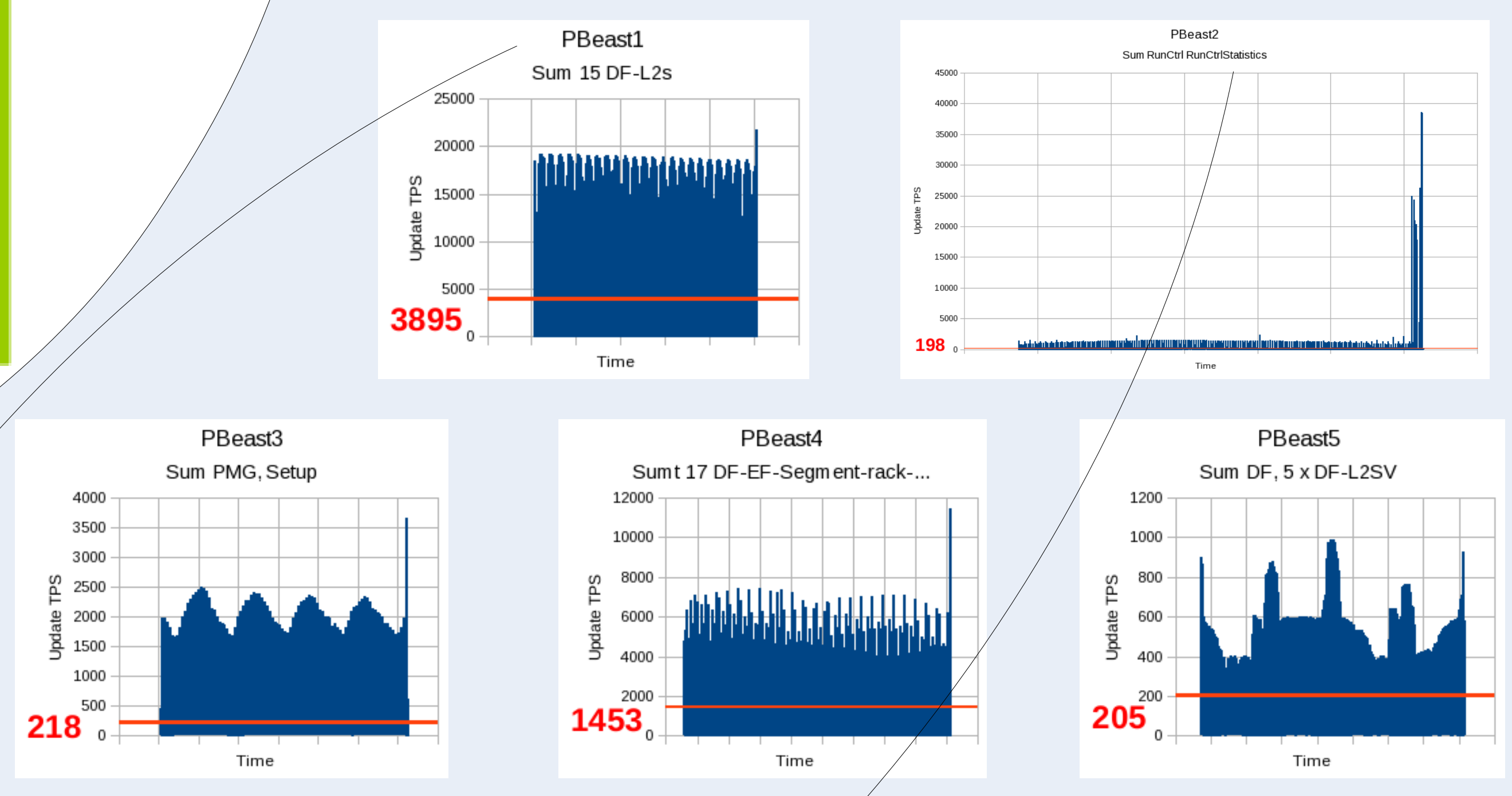
► easy to scale horizontally and configure a cluster to balance the load amongst its nodes

► data is arranged in rows of key-value pairs making it ideal to store time series data (timestamp as key).

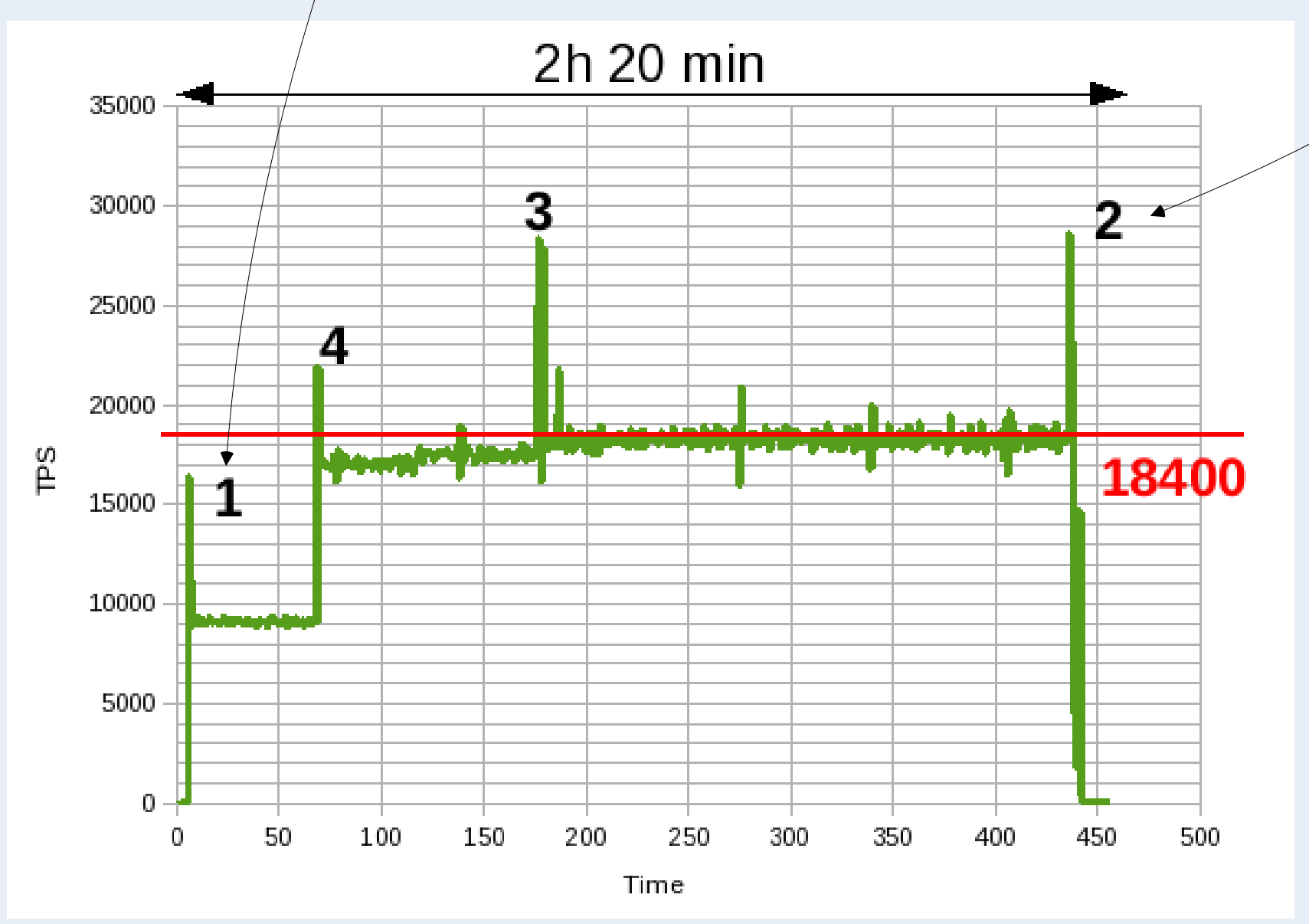
► lots of sources of information: the Apache project homepage[2], the online community or the books[3][4] written about this technology

4.Results

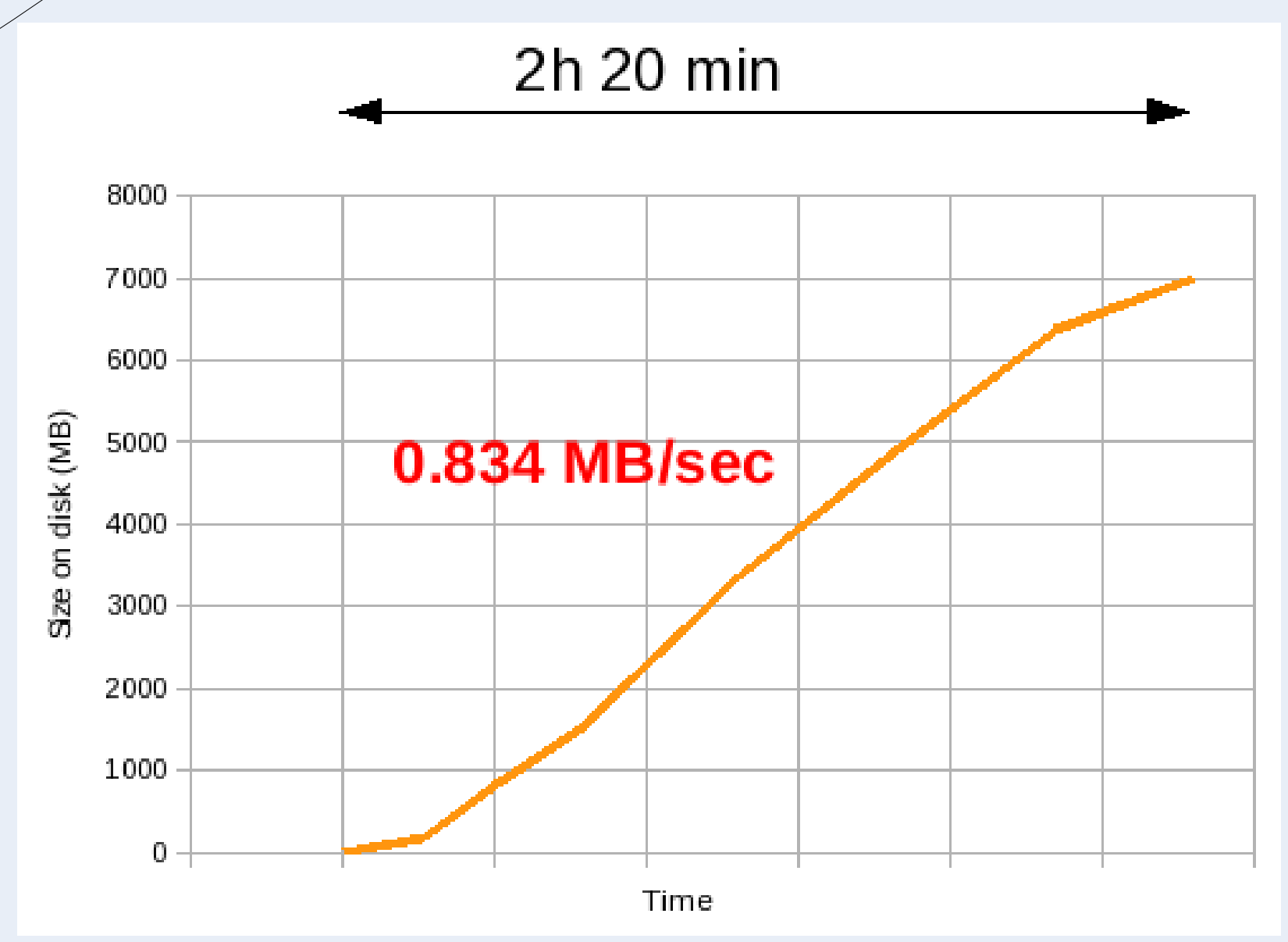
►Transactions per second (TPS) performed on each of the 5 P-BEAST instances used:



►Aggregate transactions per second (TPS) performed on the 3 Cassandra nodes in the cluster:



► Cassandra storage space size:



5.Conclusions

► the results are a good indication that P-BEAST can sustain the data rate generated by the ATLAS Online Information Service running within the TDAQ infrastructure

►measurements of the update rates confirm the varied behavior of different classes of IS servers with respect to the information rates they provide

►intermediate buffering in the P-BEAST gathering instances as well as Cassandra's insertion mechanism account for the spikes in the information rate

►the storage space required is significant due to the fact that the results shown were taken with only the mildest form of filtering applied to the incoming data (duplicates filtering). It is expected that further smoothing filters would further reduce the amount of stored data

►further work entails more testing for refining the insertion path and tuning filtering parameters, integration with TDAQ infrastructure and development of the retrieval mechanism

ACEOLE
www.cern.ch/aceole

This research project has been supported by a Marie Curie Initial Training Network Fellowship of the European Community's & Seventh Framework Programme under contract number (PITN-GA-2008-211801-ACEOLE) and by the European Community's Seventh Framework Programme under the Grant Agreement no 212114 (SLHC-PP)

MARIE CURIE ACTIONS

References:

[1] ATLAS HLT/DAQ/DCS Group. (2003, Jul.), ATLAS high-level trigger, data acquisition and controls : Technical Design Report. CERN, Sw [Online]. Available: http://cdsweb.cern.ch/record/616089

[2] The Apache Software Foundation. Apache Cassandra Home Page. [Online]. Viewed 2011 February. Available: http://cassandra.apache.org/

[3] E. Hewitt, Cassandra: The Definitive Guide, O'Reilly, ISBN: 978-1-449-39041-9

[4] E. Capriolo, Cassandra High Performance Cookbook, PACKT Publishing, ISBN: 978-1-849515-12-2