



LHCC Poster Session - CERN, 21 March 2012

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

The ATLAS detector requires a very large trigger and data acquisition system (TDAQ^[1]). This consists of approximately 20.000 applications running on 2400 interconnected computers and is foreseen to grow further in the near future.

The monitoring of these applications is performed by several sub-systems. 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.

The Information Service was not built for persistency. Due to this fact data is transient which means that experts cannot investigate historical IS information. P-BEAST is meant to provide this service. It will store data coming from over 5 million independent streams. Making this accessible facilitates long term analysis and faster debugging.

The novelty of this research consists of using a novel key value distributed structured storage technology called Cassandra to satisfy the massive time series data rates, flexibility and scalability requirements.

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.

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 integration of the P-BEAST insertion instances with TDAQ infrastructure and development of the retrieval mechanisms.

2. What is P-BEAST?

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.

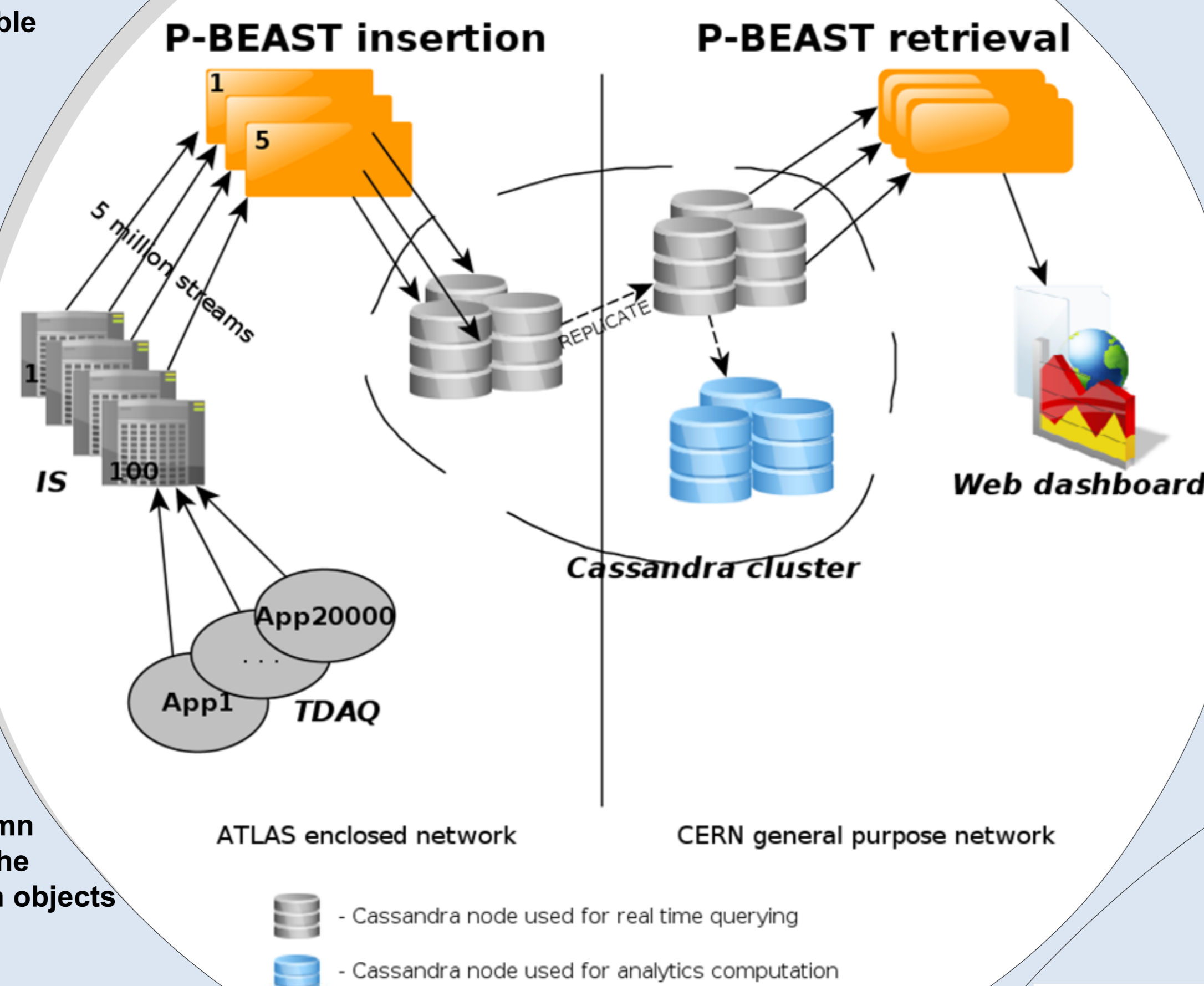
The project has thus two major parts which are reflected in its architecture.

► 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.
- filtering 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.

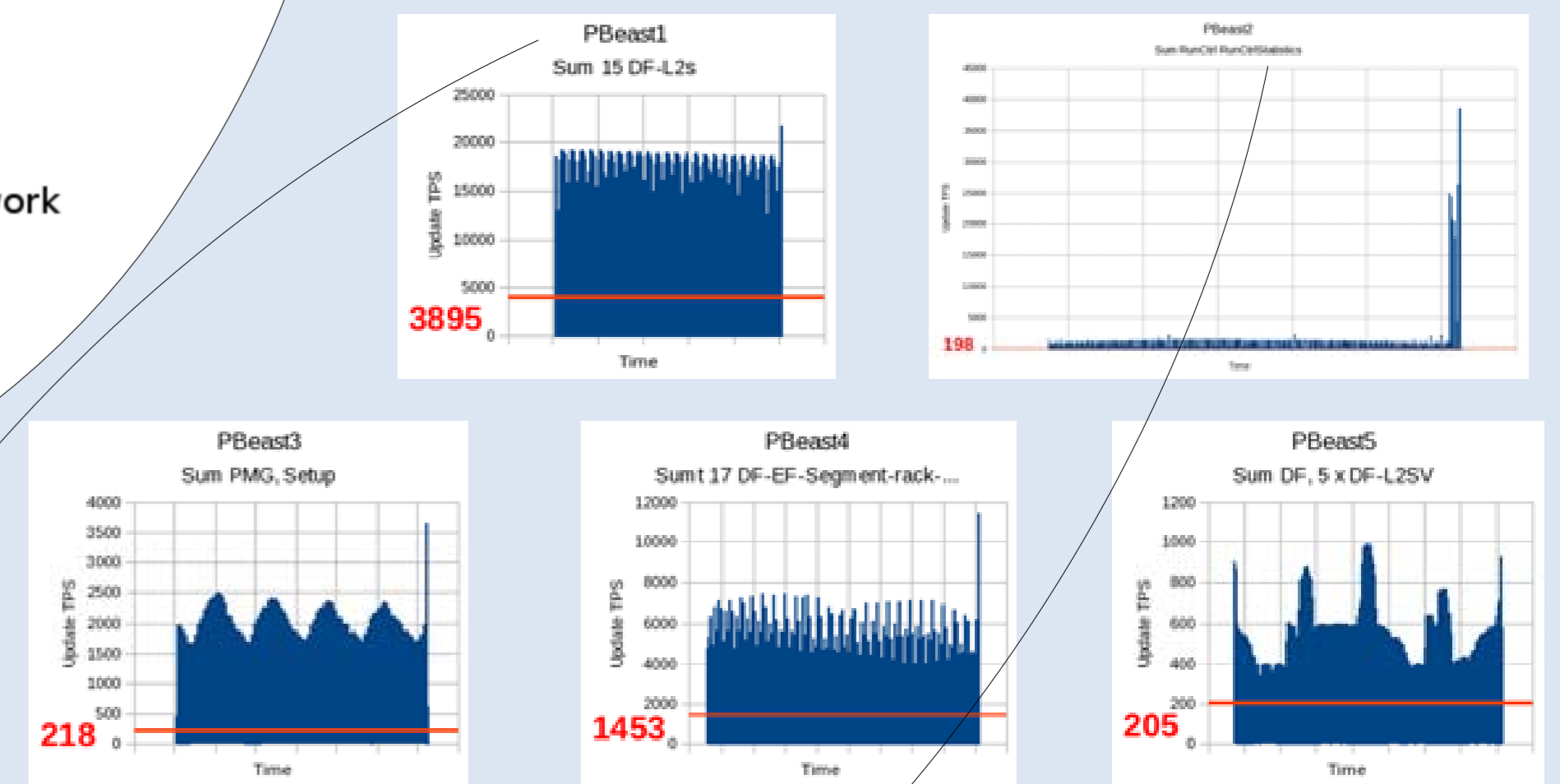
► The retrieval path:

- a REST interface shall be exposed by P-BEAST retrieval instances to any client application that wants to access the stored data over HTTP (web based dashboard).
- only time slice queries will be supported initially (value based queries will follow).
- the Cassandra cluster nodes in the general purpose network will be separated in nodes used for serving client queries (real time query serving) or computing statistics such as aggregates (offline analytics).

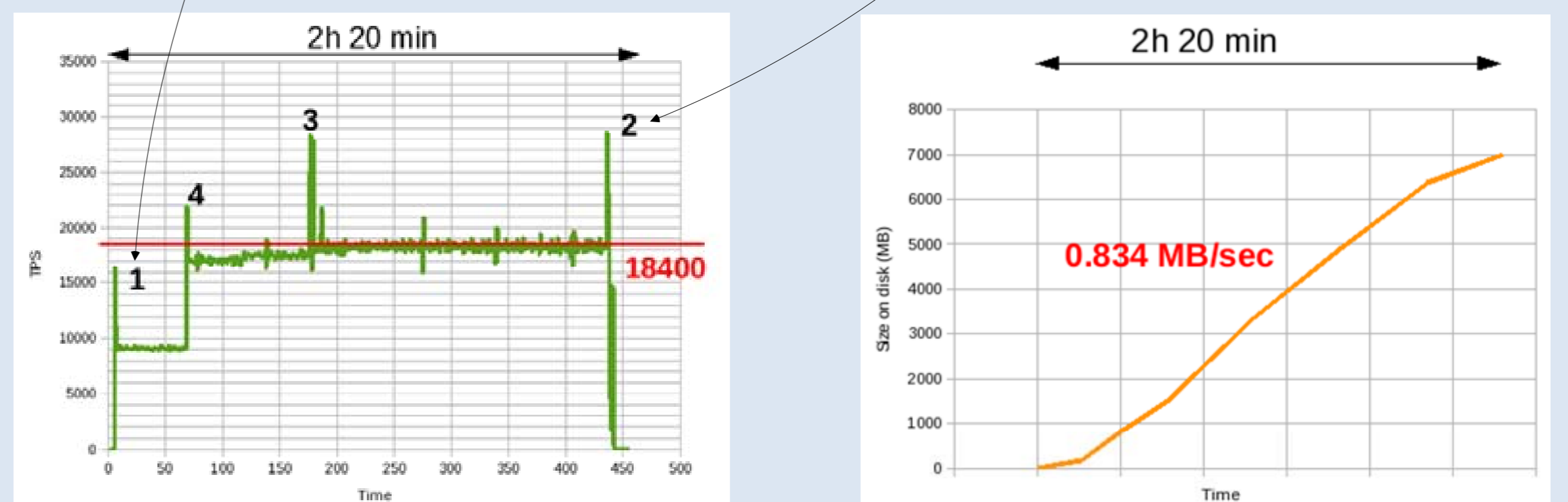


4. Insertion results

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



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



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)



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