Evaluation of Apache Hadoop for parallel data analysis with ROOT

Sebastian Lehrack, LMU München, Germany
Johannes Ebke, Günter Duckeck, LMU München

Aims

The Apache Hadoop software is a Java based framework for the processing of large data sets distributed across clusters of computers. It uses the Hadoop file system (HDFS) as data storage and backup as well as MapReduce as a processing platform.

We tested Hadoop on our local workstations as an alternative to PROOF for distributed ROOT data analysis. The main challenge of this combination is ROOT's binary file format, which must not be splitted.

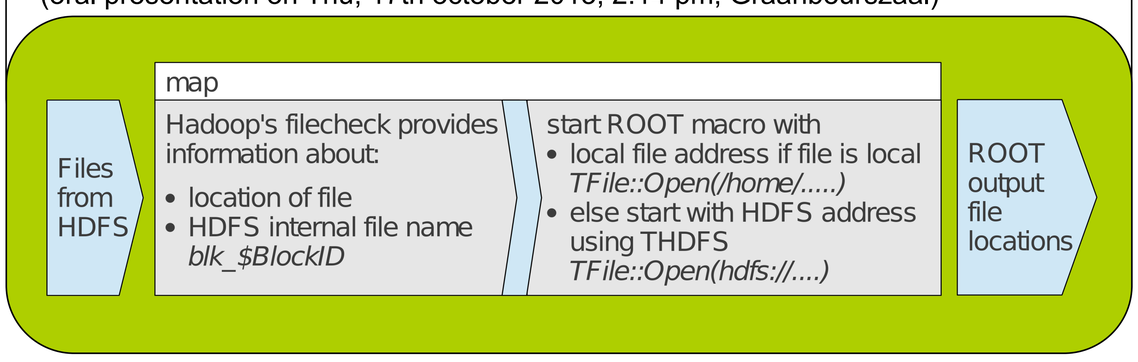
HDFS and MapReduce

The Hadoop file system is an open source derivation of the Google file system (GFS). Its intention is to achieve high data availability and to reduce costs by replacing server hard disks with ordinary hard disks. The data is therefore distributed over several data nodes and each file is replicated several times.
Data stored on the HDFS can be processed with MapReduce (this term derives from the known map- and reduce functions in functional programming). Data is read from HDFS, splitted, individually analyzed and equally distributed for a final merge.

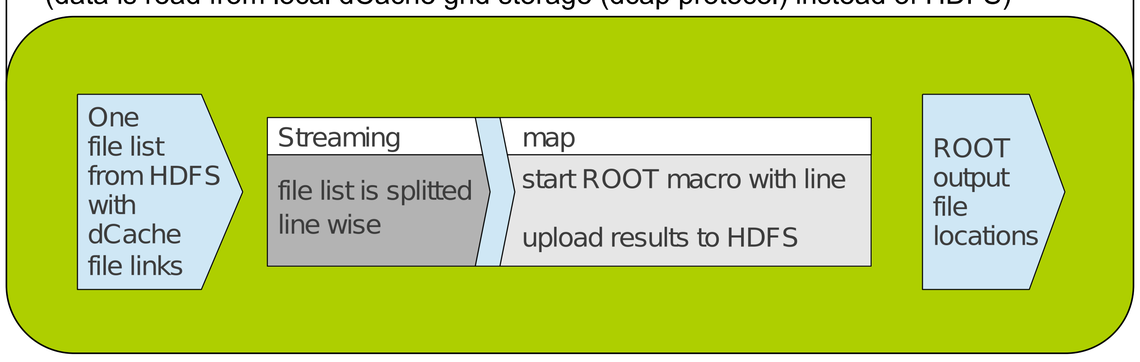
ROOT on Hadoop

In order to use ROOT's non-splittable binary file format for a Hadoop analysis, we developed and tested two different approaches on our Hadoop workstation cluster.

RootOnHadoop, Java Code by Stefano A. Russo
(oral presentation on Thu, 17th october 2013, 2:14 pm, Graanbeurszaal)



Hadoop Streaming
(data is read from local dCache grid storage (dcap protocol) instead of HDFS)



ETP local workstation cluster

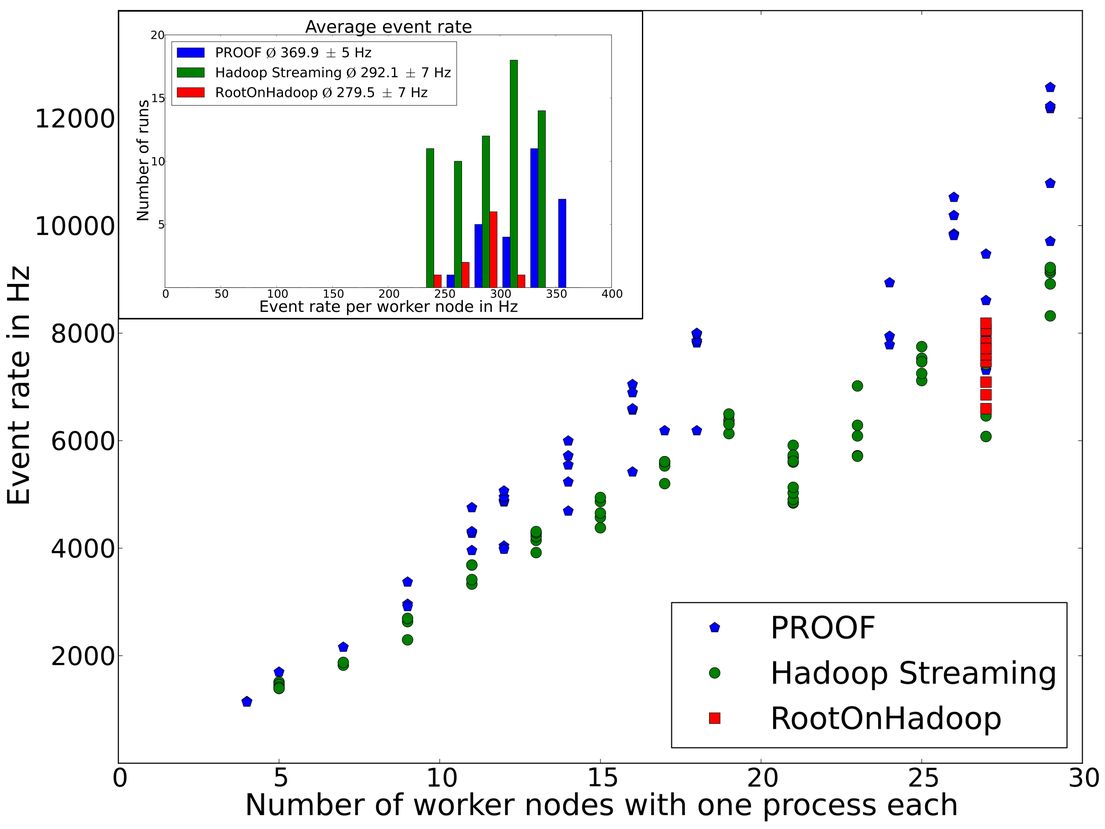
Our local workstation cluster consists of approx. 30 desktop PCs with the following configuration:

- mostly dual core processors, some quad core
- 2 GB RAM per core
- local HDD space ranges from 80 GB to 300 GB, some with SSD
- Ubuntu 12.04 OS, ROOT Version 5.34
- 1 GBits/s network connection for each PC
- 10 GBits/s network link to dCache storage

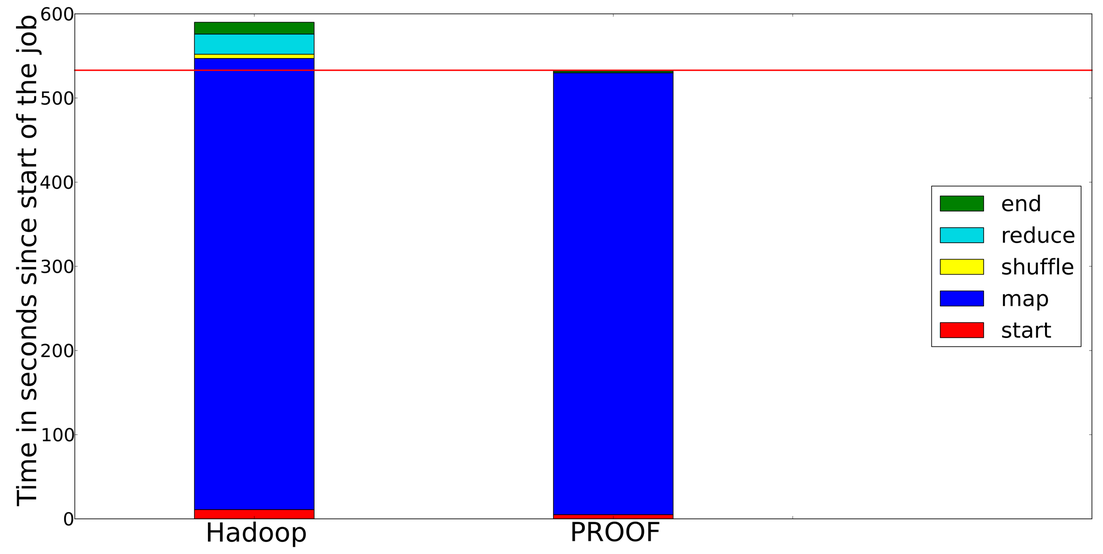
Setup

We installed Apache Hadoop on our workstation cluster consisting of 30 local workstations.
The measurements of the event rates were taken over night, analyzing the data with PROOF and with the Hadoop Streaming schema (described above). Both were alternately and consecutively used. The required time was measured with the Linux time command. To evaluate the scaling behaviour, we repeated these measurements over a wide range of cluster sizes.
In order to make the measured event rates comparable, we started the Hadoop cluster with the same worker nodes as PROOF. In contrast, we restricted to a fixed number of worker nodes for the evaluation of the RootOnHadoop schema to ensure the same data set.

Event rates Hadoop vs. PROOF



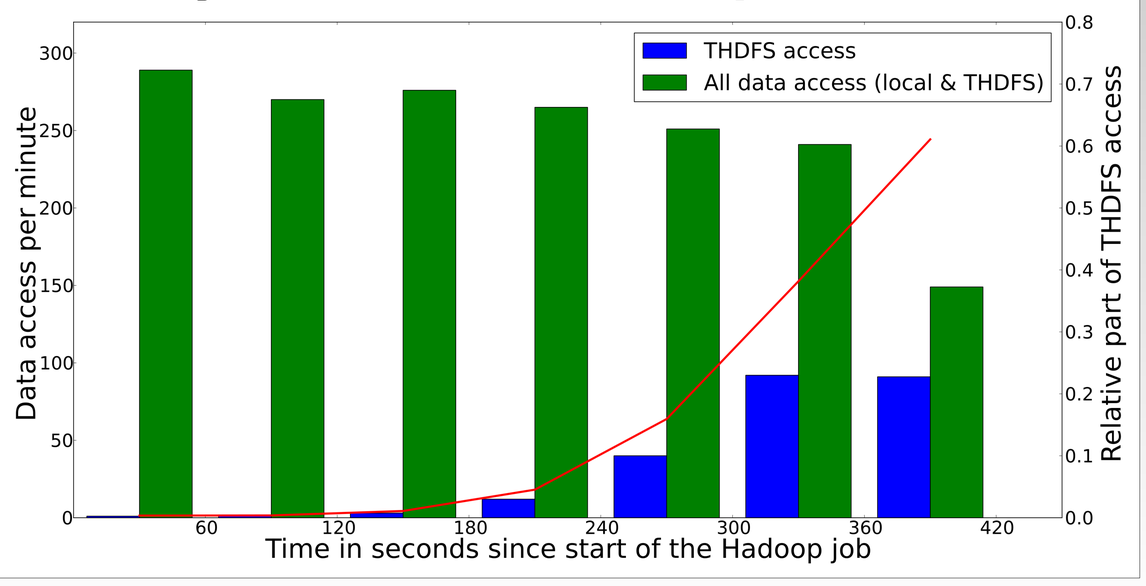
Time per analysis step Hadoop vs. PROOF



Reference analysis

A typical N-tuple analysis from LMU's ATLAS H→WW analysis group was applied. As input data, we used ATLAS 'SMWZ D3PD' files. The ROOT analysis performs a simple skimming of the data and fills the histograms for a cut-flow analysis.

Locality for RootOnHadoop



Conclusion

- processing binary data for ROOT analysis with Hadoop is possible
- it offers good job management via web interface
- it is easy to use in installation and handling
- failed tasks during a running job are well compensated and rescheduled
- event rates show linear scaling up to 30 nodes
- Hadoop is approx. 20% slower regarding event rates compared to PROOF on the same cluster
- one possible explanation: each file requires a new ROOT process

Acknowledgement

I would like to thank the Helmholtz Alliance 'Physics at the Terascale' for financing this evaluation and my appointment.