# Buffer Provisioning for Large-Scale Data-Acquisition Systems

Alejandro Santos, Wainer Vandelli,
Pedro Javier García, Holger Fröning

The 12th ACM International Conference on Distributed and Event-based Systems

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

UCLM

CERN

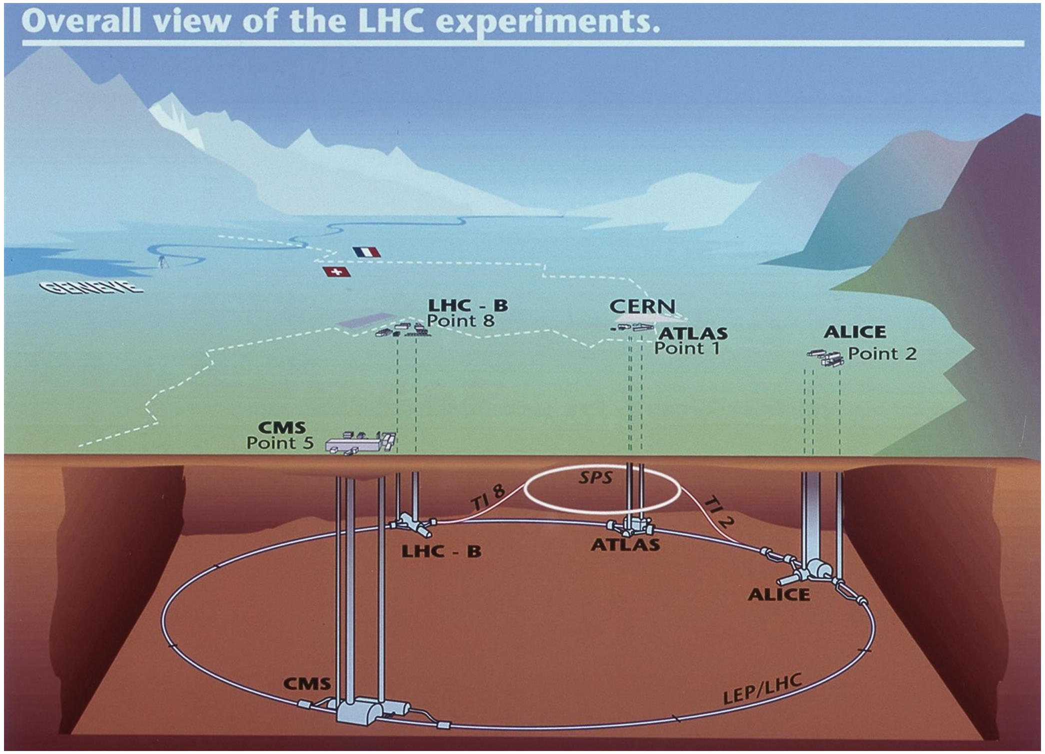
ATLAS EXPERIMENT

# Introduction

- The ATLAS experiment will go through a major upgrade in the next decade
- A very large buffering system will be put in place
- This buffer provides the possibility to trade-off computing power with storage
- In this work we analyze the consequences of this trade-off using a simulation model

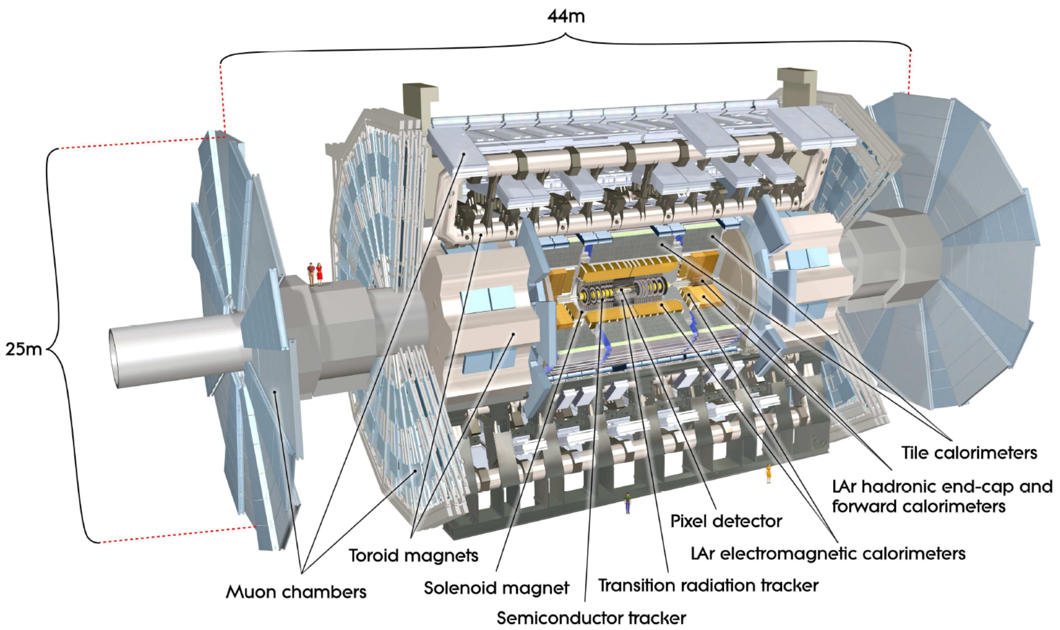
# ATLAS Experiment

- ATLAS is a general purpose experiment at the Large Hadron Collider (LHC)
  - The LHC is a 27 km particle accelerator in a tunnel 100 m underground
  - Bunches of protons are accelerated and collided 40 million times per second for periods of many hours



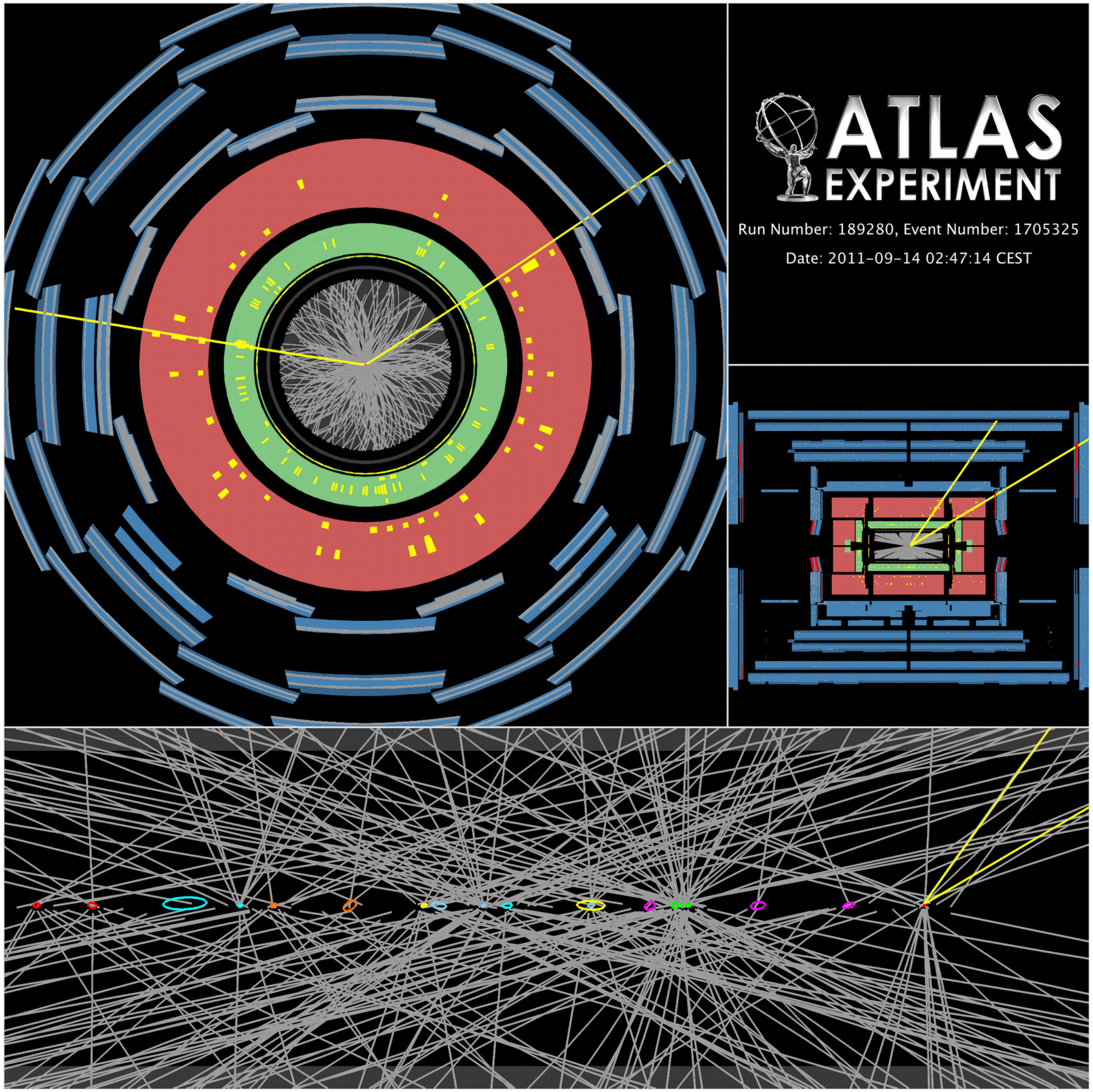
# ATLAS Detector

- ATLAS detector is designed to study subatomic particle interactions
  - Interesting collisions happening once every $10^{13}$ or more
- It consists of several different detector technologies in a layered cylindrical geometry surrounding the collision point
- Very large amounts of data produced from sensors
  - 100 million data channels
  - Today, reading out the full detector results in 2 MB of data per collision
  - Sensor data is logically grouped as “fragments” and many fragments constitute the full collision event record



# ATLAS Data

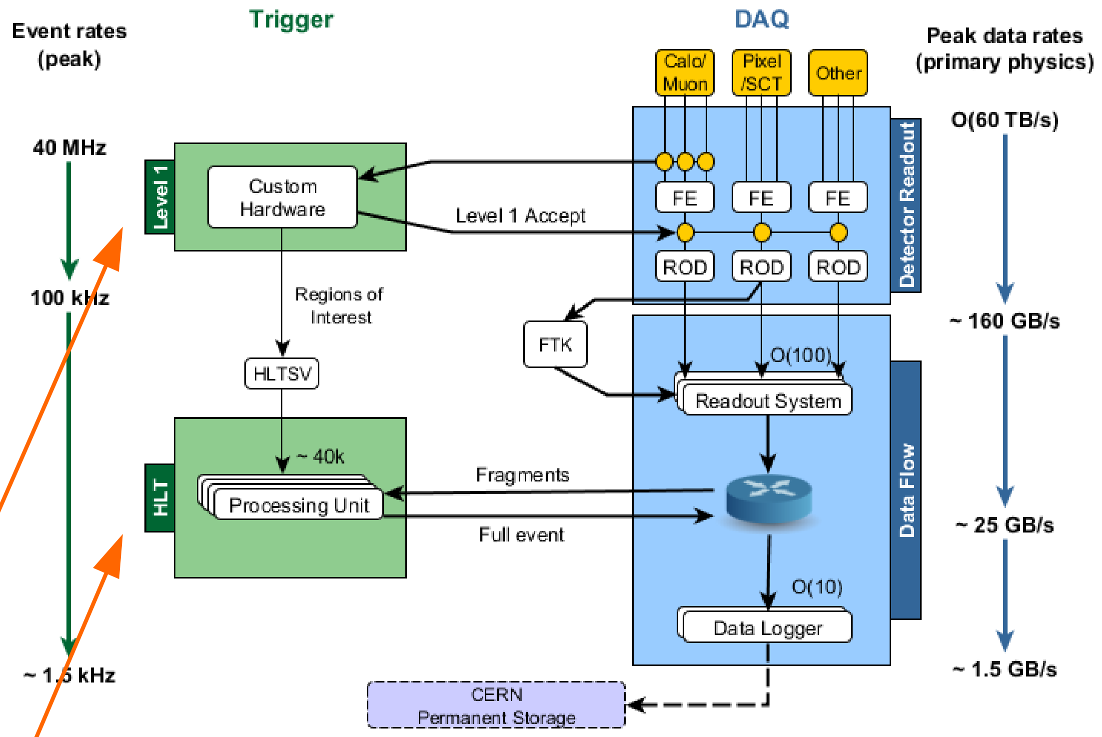
- Sensor data are used to reconstruct the collisions
  - Each is called an "Event"
- The signals from sensors are used to calculate trajectory, energy and momentum from particles
  - Custom reconstruction software processes these signals which is a very CPU intensive problem




https://twiki.cern.ch/twiki/bin/view/AtlasPublic/EventDisplayStandAlone

# Trigger

- Due to the huge amount of incoming data, not every event can be saved
- An on-line filtering system selects collision events relevant to the experiment's goals
  - Called "Trigger"
- Implemented in two levels
  - 1. Custom electronics, implemented as a real-time system with strict timings
  - 2. Processor farm connected over a packet network, with a complex processing time distribution and large variance




Functional diagram of the current ATLAS Trigger and Data Acquisition system

https://twiki.cern.ch/twiki/bin/view/AtlasPublic/ApprovedPlotsDAQ

# DAQ

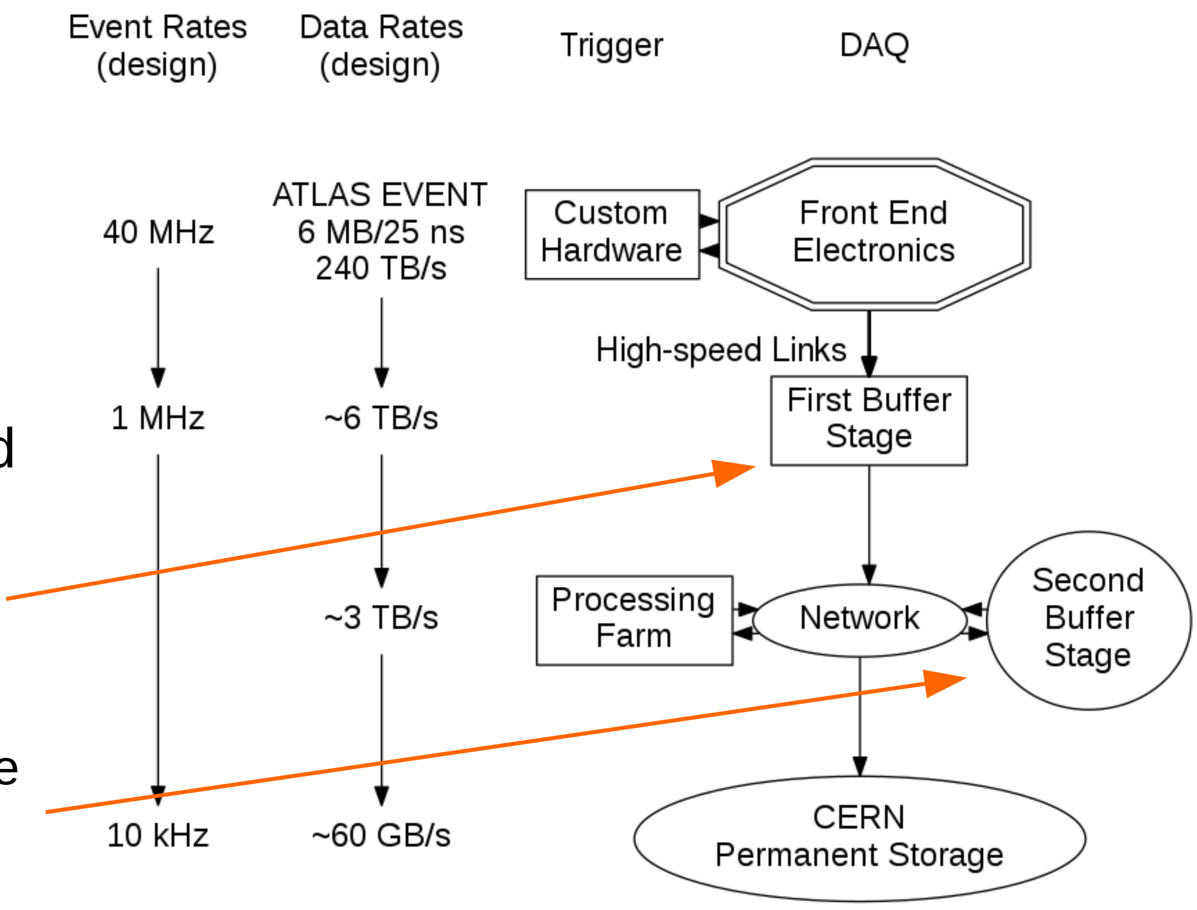
- The second level includes a buffering space
  - It gives sufficient time for collision data to be processed at this level
  - Not all data is required to be read to analyze a collision in the second trigger level
- Each level reduces the number of collision events
- Single buffering stage
  - Fragments arrive to an individual buffer machine
  - For each event, fragments are spread over different machines
  - Processors read individual fragments from the buffers

Processing farm for the ATLAS Data Acquisition System (DAQ), ~1900 machines, ~40k CPU cores (Picture from 2014)

http://atlas.cern/resources/multimedia/detector

# ATLAS Upgrade

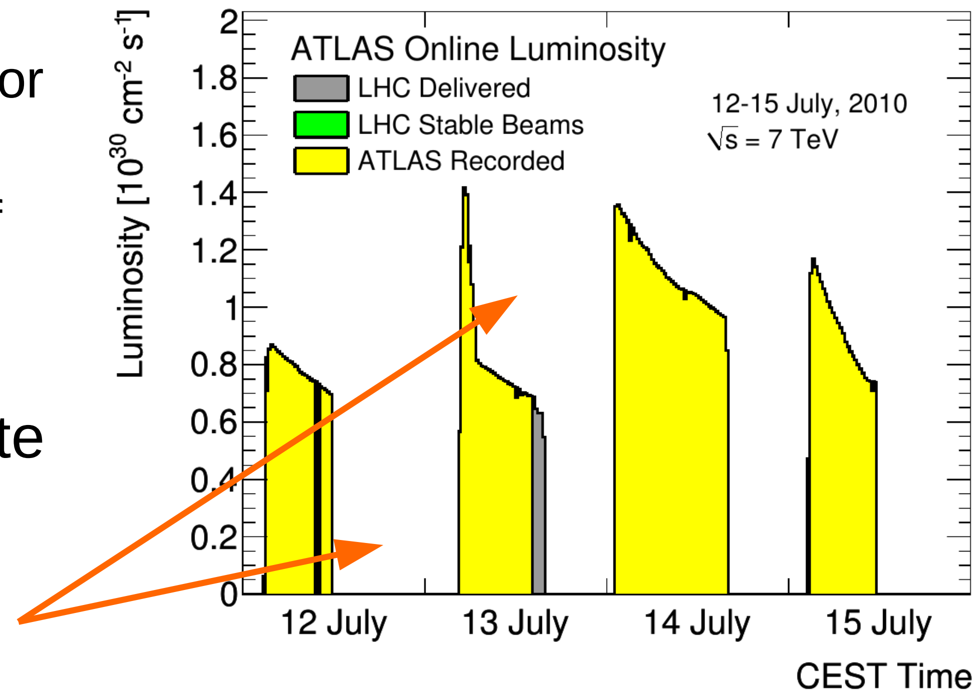
- The next upgrade to the ATLAS system will happen in ~2026
- In the design, there is a very large buffer to decouple data production and data processing between the two trigger levels
- A new buffer stage is introduced
  - The first buffer stage stores incoming data for short term, while doing submodule-specific tasks
  - The second buffer stage is a large storage space to store data for long periods of time, in the order of days
- We study this second stage buffer




General overview of ATLAS Trigger and Data-Acquistion System for the next upgrade, “Phase-2”

# Data Production Cycle

- Data production is not constant
  - Data rates change
  - Data rates show a cyclic behavior
  - In the LHC, this is needed for operations and it targets 60% of time delivery
- Simplest design of a DAQ system is to be sized to operate at peak rate
  - During the time with no peak operation, processing power is partially or totally unused
- There is an opportunity to trade-off processing power and storage

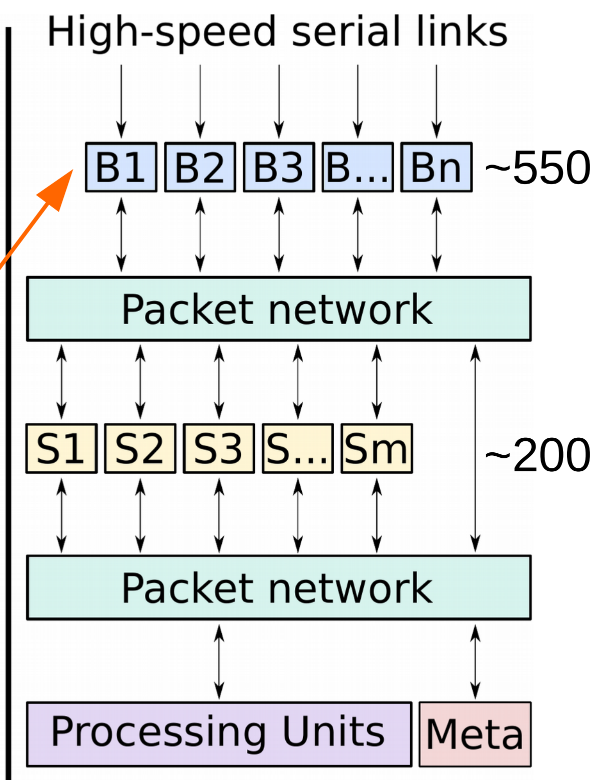



The probability of a collision is directly proportional to the luminosity

https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LuminosityPublicResults

# The split buffer model data flow

- The “split buffer” model was created to study the future ATLAS upgrade
  - Study the trade-off between processing power and storage
- Dataflow
  - Data arrives to the first buffering stage, “Bn”, synchronously
  - It is buffered for a short time period
  - It is asynchronously moved to the second buffer, “Sn”
  - Processors requests “fragments” of each “event” to analyze and filter data
  - Each “fragment” represents one sensor readout component




Split buffer model

# Simulation Model

- Follows “split buffer” model
- Implemented in OMNeT++
  - Robust and user-friendly discrete event simulation framework
- Simulation model has many input parameters, including:
  - Data rate
  - Total cores count
  - Processing time
  - Network overhead
  - Duty cycle (%) and duration

- Assumptions
  - Network overhead is added as a small fraction of the throughput
  - Network is ideal, no packet loss, infinite bandwidth*

(*) T. Colombo et al. 2016. Optimizing the data-collection time of a large-scale data-acquisition system through a simulation framework. The Journal of Supercomputing 72, 12 (2016),

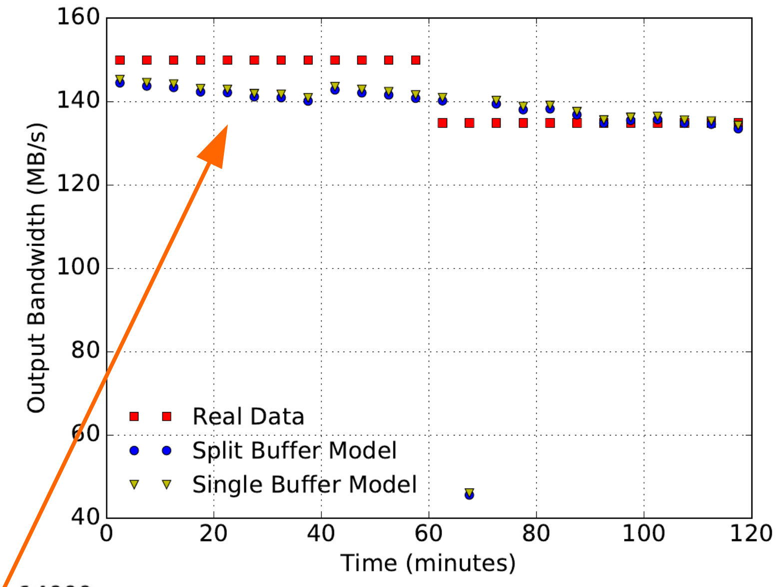
# Simulation Model Validation

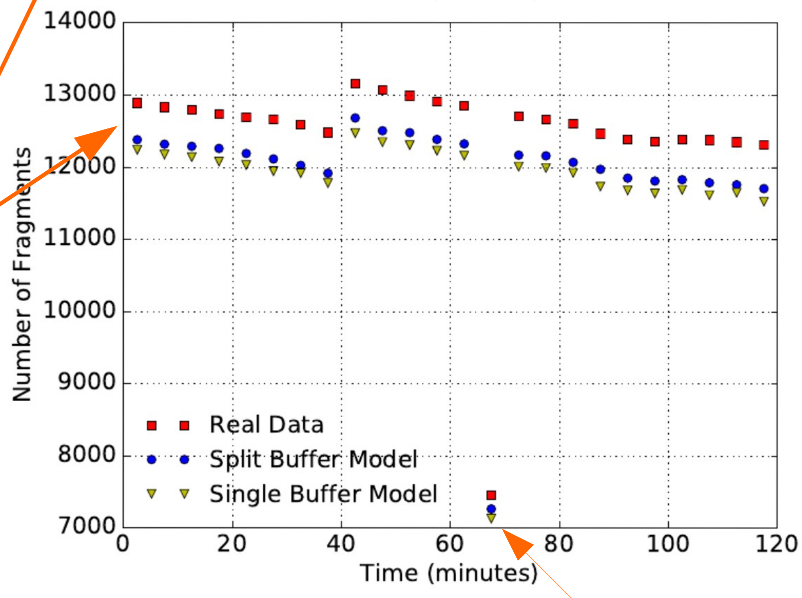
- Why → produce a model that can be used to study the buffering
- How → two separate validation schemes:
  - 1. By comparing with current system operational data
  - 2. By comparing with a small-scale emulated DAQ system software
- What → three metrics of model validation:
  - 1. Average output throughput of buffers
  - 2. Average number of fragments in buffers
  - 3. Average number of occupied processors

# Validation with Operational Data

- Compare metrics against existing operational data
  - Run the simulation using parameters from the existing system
  - Compare output metrics
- Results for running 24 simulations, 2 hours of data
  - Each point is one simulation, five minutes average
  - **There is good agreement between simulation and data**
- Simulation mismatch
  - Archived monitoring data for the output bandwidth has a 1 h resolution, data shows a step function
  - There is a bias in the number of fragments results. Simulation does not account for overhead latencies in the system

| | Mismatch |
|---|---|
| Output bandwidth | 4.8 % |
| Number of fragments | 4.2 % |
| Occupied processors | < 0.1% |

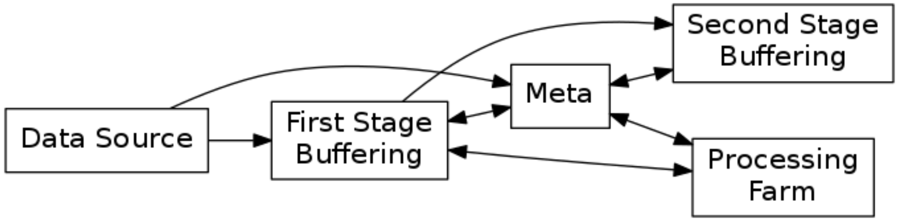



External conditions changed data rates of the real system

# DAQ Emulator

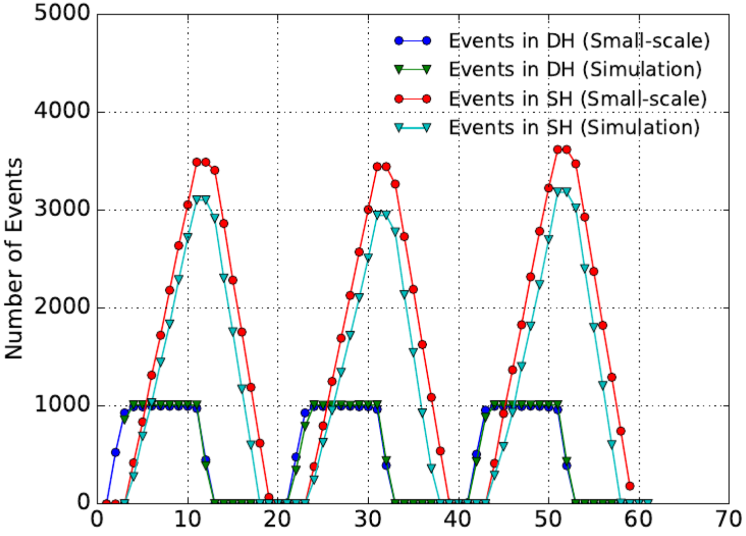
- A small-scale software DAQ emulator was written to test in a real context the split buffer design
- It follows the design of a DAQ system, without processing any real data
  - Implemented as a distributed system in Python
  - Configurable number of application instances running

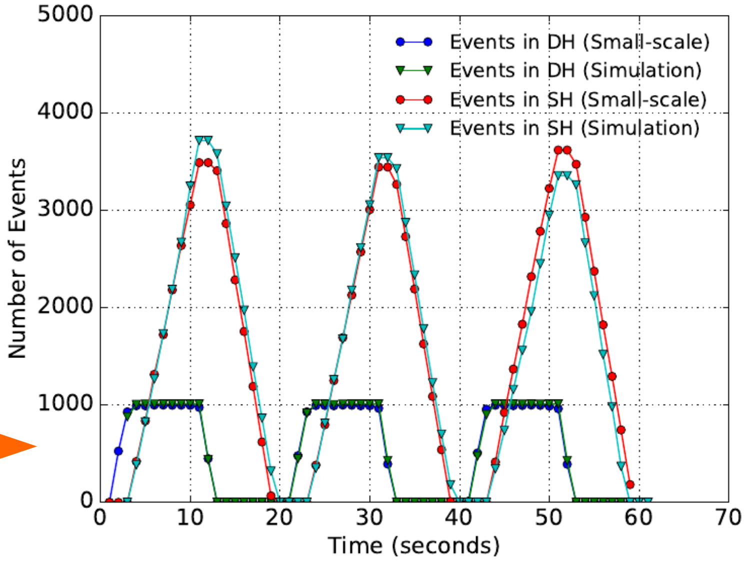



| | Applications in the Small-Scale | Elements in the ATLAS upgrade |
|---|---|---|
| First buffer stage | 10 | 550 |
| Second buffer stage | 1 | 200 |
| Processing Farm | 12 | Very large |

# Validation with Emulator

- Comparing metrics against the DAQ emulator
  - Run both emulator and simulation using common parameters
  - Compare output metrics
- Results for running a single execution for 60 seconds
  - Each point is a sample, one each second
  - **There is good agreement between simulation and data**
- Simulation mismatch
  - Initial simulation does not account for overhead latencies in the system
  - Measuring and including calibration latencies in new simulation produces very accurate results



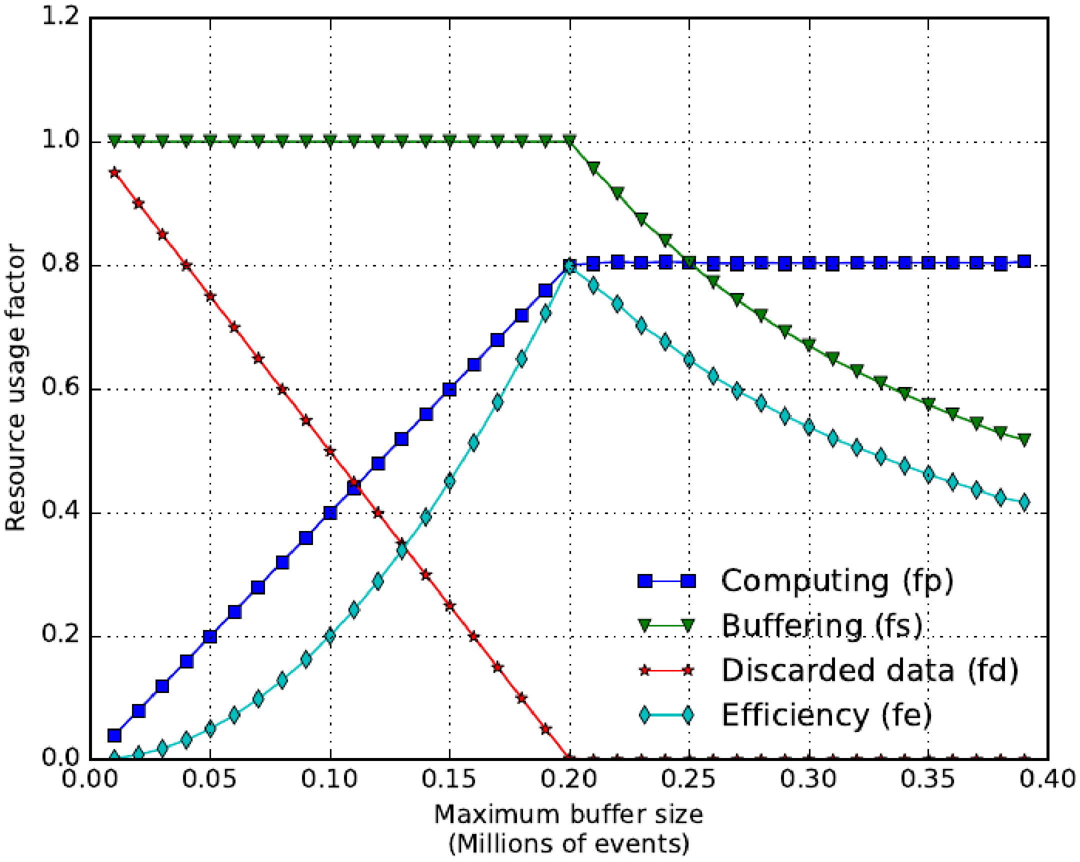

DH is the first buffer stage
SH is the second buffer stage

# Operational Envelope

- The "split buffer" model is used to study the efficiency of the system
  - Called "Operational Envelope"
- Function of resource utilization of:
  - Processing power usage
  - Buffer space usage
  - Discarded events due to lack of buffer space
- Efficiency:
  - Fe = Fp * Fs * (1 – Fd)
- Where
  - Fe is the utilization efficiency of the system
  - Fs is the storage utilization of the system
  - Fp is the processing power utilization of the system
  - Fd is the discarded events of the system
- Discarded Events
  - With limited buffering space, collision data can be lost
  - Fundamental to physics experiments to minimize the amount of discarded data
- Simulation experiments: three experiments are run to find the best buffer size

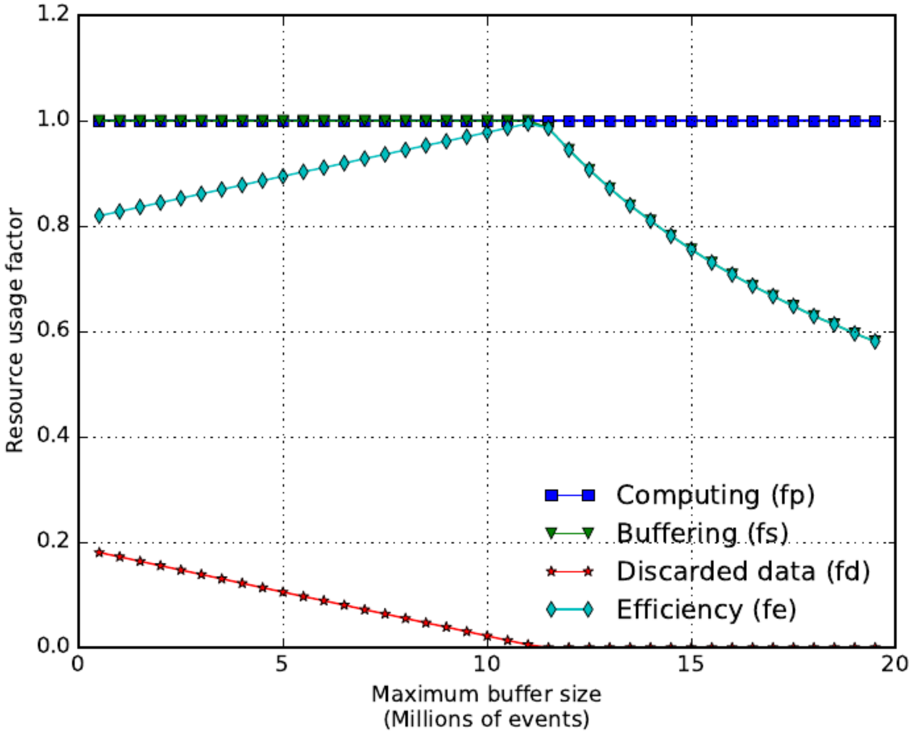
# Data production at constant rate

- Conditions:
  - At 1 million events per second, with a processing time of 200 ms and 250,000 cores
  - Over-provisioning the processing system, shows peak efficiency of the system is at ~80 %
- Results
  - The efficiency “Fe” is the product of Computing, Buffering and one minus Discarded factors



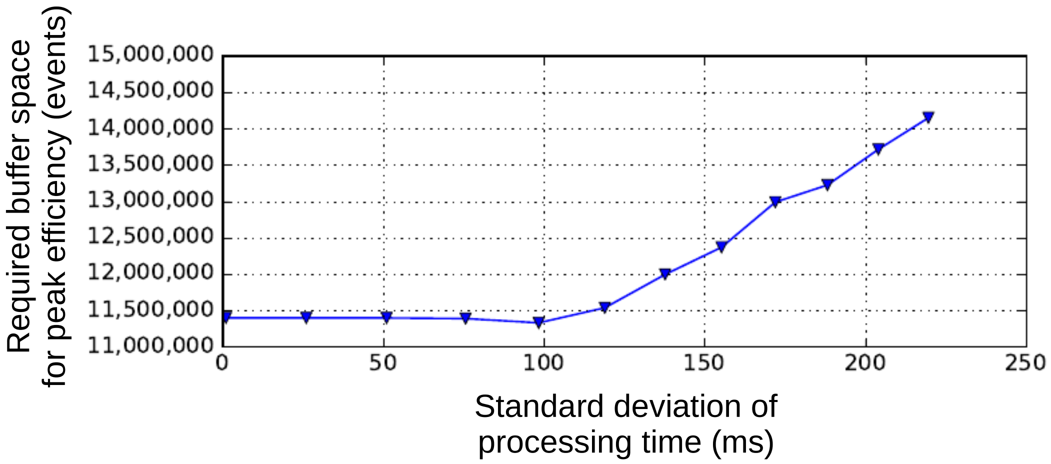
# Data production as a cycle

- Conditions:
  - 1 million events per second, with a processing time of 200 ms and 150,000 processors
  - Run the simulation with a 75% duty cycle: during 60 simulated seconds, data arrives for 45 seconds
- Results:
  - The buffering system absorbs the incoming and processing rates differences



# Data processing variance

- Conditions:
  - 1 million events per second, with an average processing time of 200 ms and 150,000 processors
  - Explore different variance values for the processing time using a normal distribution from 1 to 300 ms in variance
  - For each variance value, the buffer space for the peak efficiency is reported
- Results
  - Large variance shows there is a meaningful impact on storage requirements



# Conclusions

- We introduce a discrete event-based simulation model for high-energy physics experiments
  - Accurate results, validated with both real data and with a software DAQ emulator
  - Simulations agree to within 5% of the real system
- Data production as a cycle allows to have a trade-off between storage and processing power
  - When there is no new data, the system can continue processing buffered data
- A simulation model allows to study complex interactions between system components
  - The operational envelope represents a high-level overview of the resource utilization
  - It also enables the exploration of different scenarios in order to understand tolerance margins to use in building a real system
- Next steps include the study of more specific storage technologies

# Backup Slides

# The Two Models

- Two distinct DAQ architectures are studied
- The “single buffer” model follows the design of the current ATLAS system
  - Previous work where the current ATLAS buffering system was studied
    - A Santos, W Vandelli, P Garcia, H Fröning. Modeling and Validating Time, Buffering, and Utilization of a Large-Scale, Real-Time Data Acquisition System. MSPDS Workshop 2017.
- The “split buffer” model was created to study the future ATLAS upgrade
  - Study the trade-off between processing power and storage
- This work is about the “split buffer” model
  - The “single buffer” model is used in the validation of the model