!CHAOS

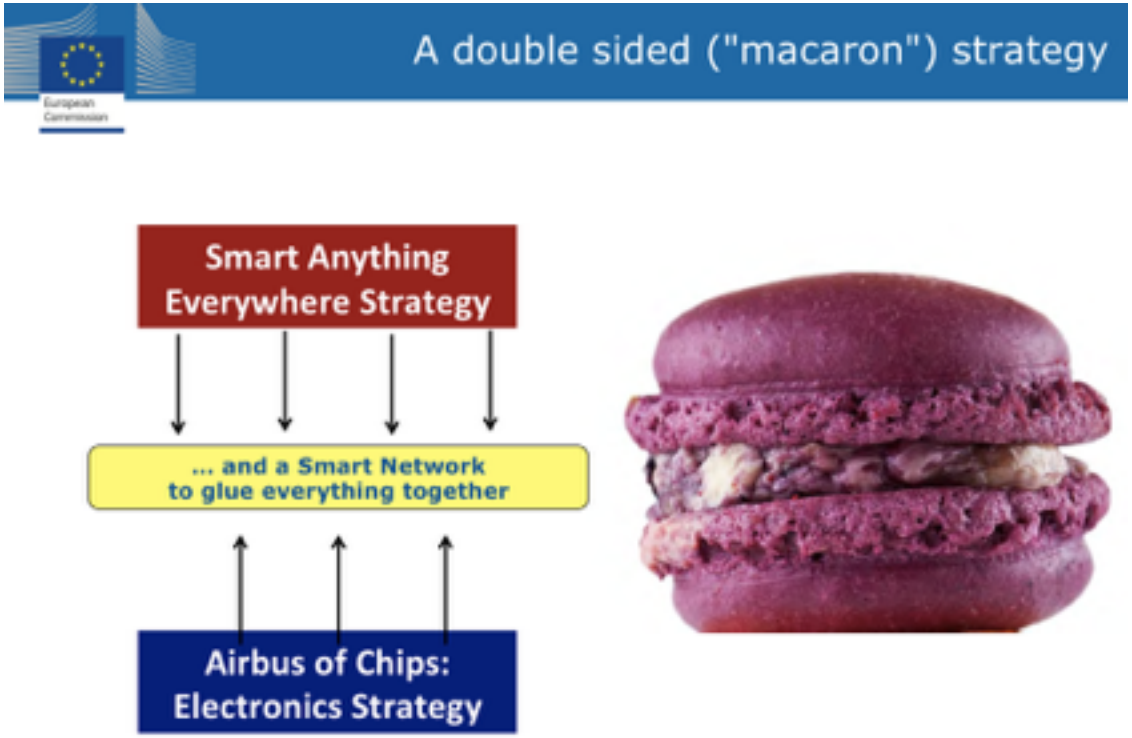
Performance e Benchmark

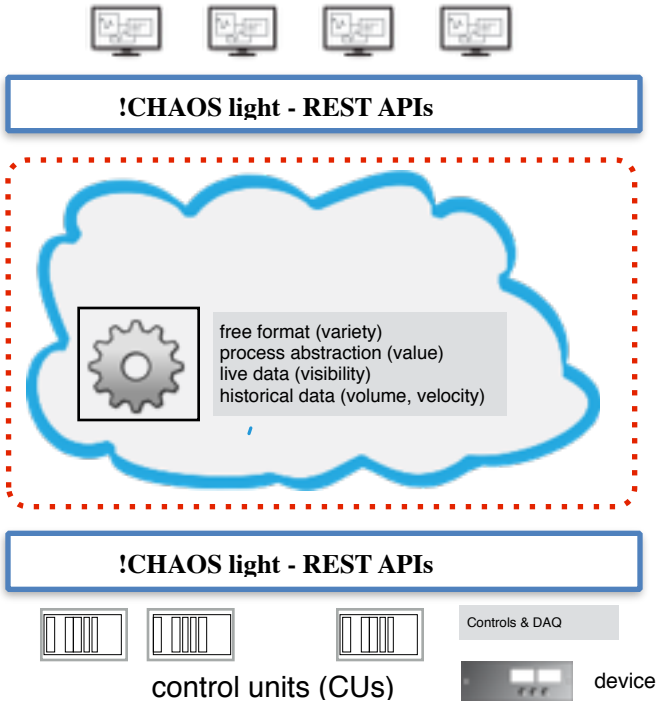
!CHAOS: a cloud of controls - G. Mazzitelli
14 Maggio 2015

INFN
Istituto Nazionale
di Fisica Nucleare

Internet of Things

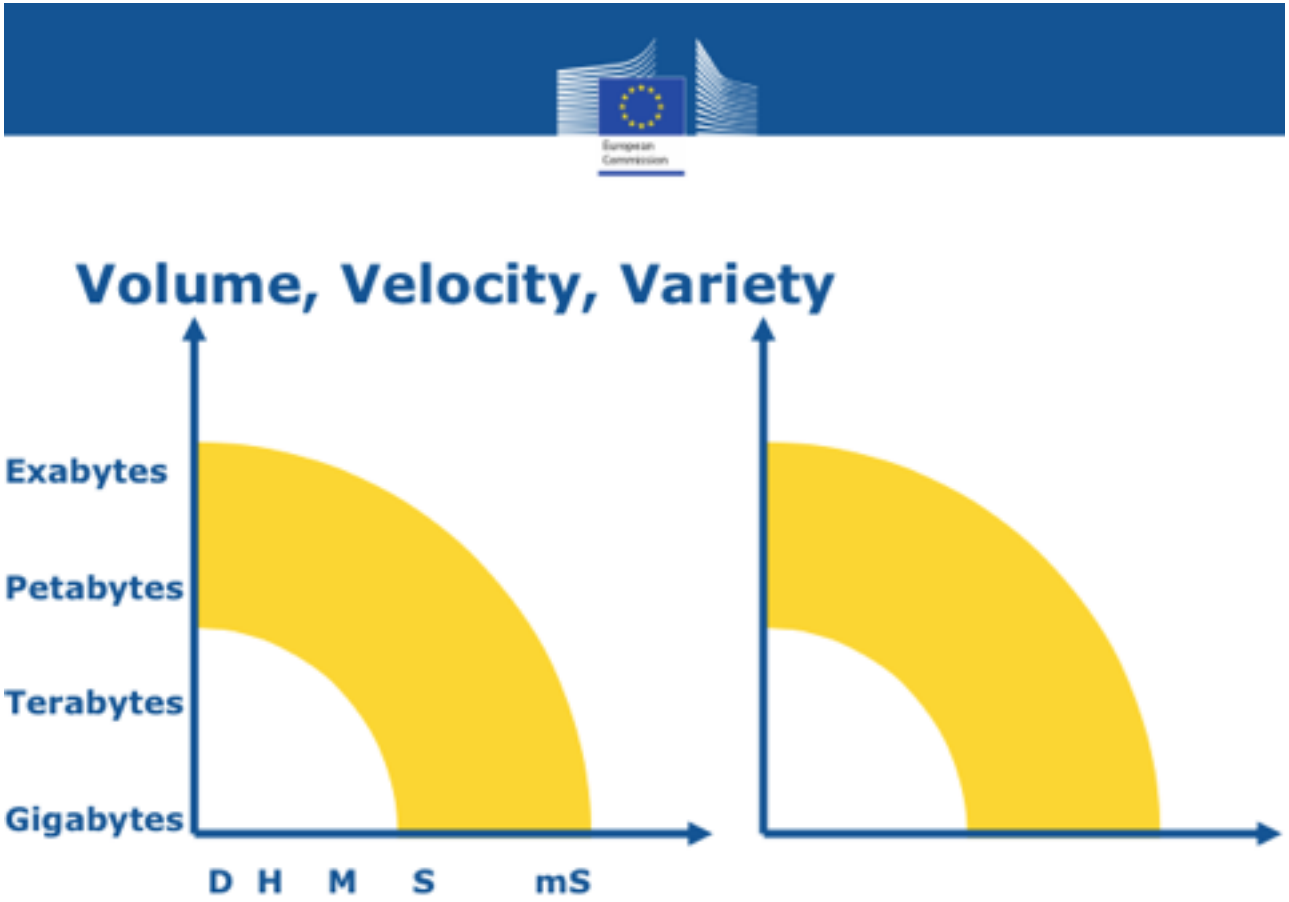
Internet of Things and Platforms for Connected Smart Objects cuts across several LEIT-ICT technological areas (**smart systems integration, cyber-physical systems, smart networks, big data**) and brings together different generic ICT technologies and their stakeholder constituencies to develop technological platforms which will have a strong influence on the way in which we live and work.





Big Data

EU support the Big Data challenge by addressing fundamental research problems related to the **scalability and responsiveness of analytics capabilities** (such as privacy-aware machine learning, language understanding, data mining and visualization)



benchmarks & performances (WIKI)

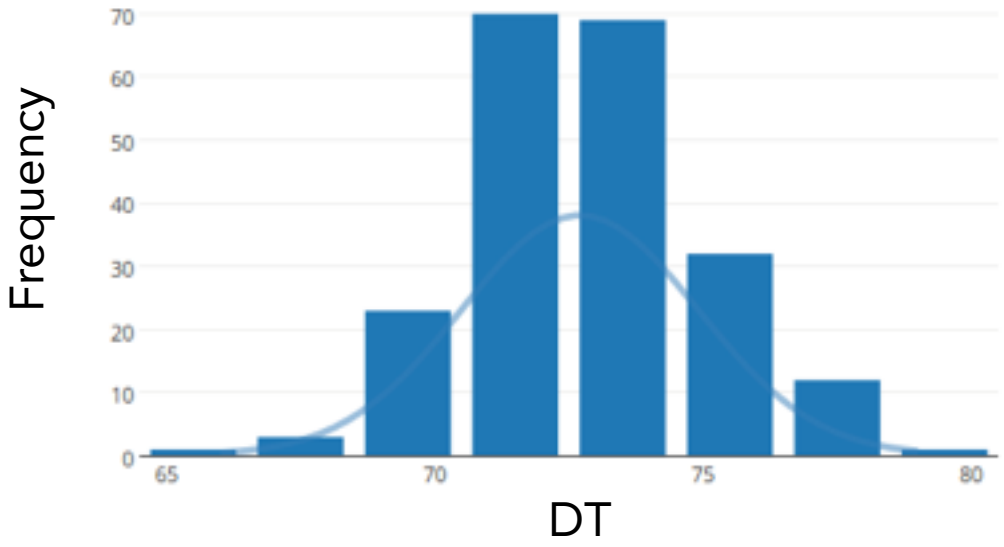
- In computing, a **benchmark** is the act of running a computer program, a set of programs, or other operations, in order to assess the **relative performance** of an object, normally by **running a number of standard tests and trials against it**. The term 'benchmark' is also mostly utilized for the purposes of elaborately-designed benchmarking programs themselves.
- **Performance** measurement is the process of **collecting, analyzing and/or reporting information** regarding the performance of an individual, group, organization, system or component. Computer performance is c**haracterized by the amount of useful work accomplished by a computer system or computer network compared to the time and resources used**. Depending on the context, high computer performance may involve **one or more of the following**:
    - Short response time for a given piece of work
    - High throughput (rate of processing work)
    - Low utilization of computing resource(s)
    - High availability of the computing system or application
    - Fast (or highly compact) data compression and decompression
    - High bandwidth
    - Short data transmission time

why? performance measurements

- **external and internal** motivations:
  - showing the performances of the system (framework) to competitors and stakeholders
  - demonstrating the project milestones & deliverables
  - checking the theoretical designs in a real environment
  - confirming e/o rejecting branch choices

DT distribution (main & SD)

- **Latency** is a time **interval between the stimulation and response**, or, from a more general point of view, as a time delay **between the cause and the effect** of some physical change in the system being observed.
- **Jitter** is the deviation from true periodicity of a presumed periodic signal in electronics and telecommunications, often in relation to a reference clock source.



what? benchmark & performance

1. scalability of performances and size
2. integration of all functionalities ?
3. abstraction of services, devices and data
4. easy and modular customisation ?
5. extensive data catching for performance boosting
6. use of high-performance internet software technologies

- key/value database (KVDB) implemented as distributed memory object caching systems (DOC).
- using non-relational databases, in a specific document database.
- optimizing and embedding high performance e/o standard inter process communication and handling (RPC/direct I-O/Events).
- abstracting data structure through binary data serialization (BSON)
- embedding COTS and, in general, open hardware, in order to minimize costs and integration time
- making “Controls as a services” to be responsive to new trends of IT technologies and the request of industries and society.

from Qualitative to Quantitative

Concepts & Solution: Big Data, IoT, Cloud, etc

Quality: Variety, Volume, Velocity, Value and Visibility

Quantity to mesure or qualify: Capacity, Scalability, Reliability, Flexibility, Accessibility, Analytics, Responsiveness, Versatility

methodology

- ideated and performed by other person than designer (ex AB @ BTF for REST API C.lib), in general based on accredited and third party software
- differential measurements
- modelling
- …

es TANGO

Table 1: Performance - *performance figures measured on Linux on a Pentium @ 200MHz, Windows/NT on a Pentium III @ 450 MHz, Solaris on an UltraSparc 1, HP-UX on a HP9000/735, network was Ethernet 10baseT. Note the times presented here represent the minimum overhead to trigger an action, the time to execute the action in the server has to be added to this.*

from - to	platform	transferred	time
client - device	Linux	8 bytes	1.7 ms
client - device	Win/NT	8 bytes	0.9 ms
client - device	Solaris	8 bytes	3.7 ms
client - device	HP-UX	8 bytes	3.0 ms
client - device	Linux	1 Mbyte	1.5 s
build connection	Linux	1 device	10.0 ms

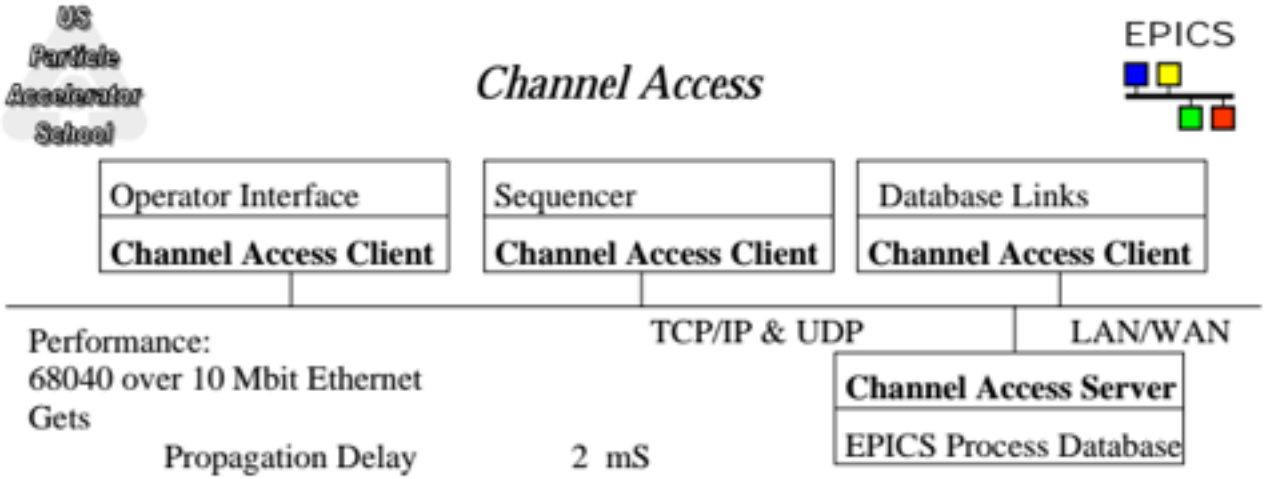
Performance measurements for event distribution have been carried out using the Data Distribution Service (DDS) [10] implementation OpenSplice [11] and the publisher/subscriber pattern of the ØMQ [12] Socket API.

The measured performance values are in received events per second between two machines (P4, 2.5GHz, Ubuntu 9.04 – Core 2 Duo, 2.6GHz, Ubuntu 9.04) on a 100 Mbit network.

Table 2 : Event System Performance Tests

Sub	1 int (32bits)			1024 int		
	Tango	**DDS**	**ØMQ**	**Tango**	**DDS**	**ØMQ**
1	770	12500	45000	650	1850	2400
5	400	7900	14000	200	1800	500
10	220	6500	7300	100	1700	230

es EPICS


US Particle Accelerator School
Channel Access
EPICS
Operator Interface
Channel Access Client
Sequencer
Channel Access Client
Database Links
Channel Access Client
TCP/IP & UDP
LAN/WAN
Channel Access Server
EPICS Process Database
Performance:
68040 over 10 Mbit Ethernet
Gets
Propagation Delay 2 mS
Throughput 7.8K /sec
Puts
Propagation Delay 1 mS
Throughput 17K /sec
Monitors
Propagation Delay Dependent
Throughput 10K / sec
(Typically 10% channels have monitors)
(memory use in IOC - 2 Meg / 60 connections)
(30% network load for 10K monitors / second)
Increase bandwidth with Routers, Bridges, Higher speed networks and EPICS gateway
1999/Ph 514: EPICS Overview
16

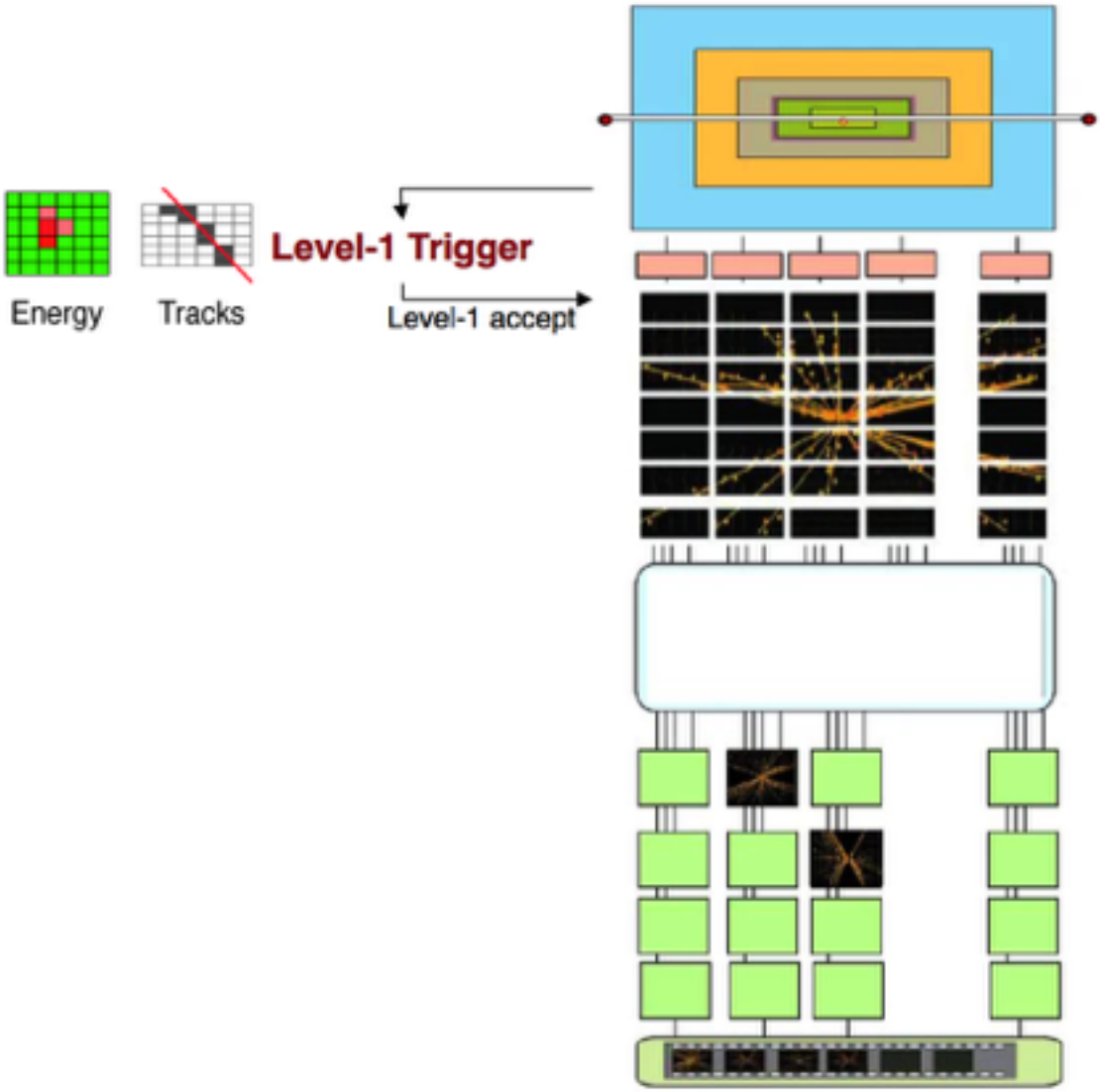
EPICS example (performance & benchmark)

- EPICS Input/Output Controller core (IOCcore) software has been ported to several different operating systems (OSs) and many hardware platforms. This module **compares the EPICS IOCcore runtime performance on one hardware platform** (MVME2100 PowerPC) and three popular Operating Systems
- Three real-time parameters are measured: **interrupt**, **context switch**, and **total response latency**. On Linux, more detailed interrupt latencies are measured: interrupt top half to bottom half, and interrupt bottom half to user space interrupt service routine.

OS		Minimum	Median	Maximum	>100 μs(%)
Private Network	Linux 2.4.30	9	12	72	0.000
	Linux 2.4.2hhl2.0	8	12	62	0.000
	Linux 2.6.13 non preemptive	9	13	80	0.000
	Linux 2.6.13 preemptive	8	13	71	0.000
	Linux 2.6.13 preemptive with user level ISR	15	21	77	0.000
	vxWorks 5.5 net task has higher priority	6	9	53	0.000
	vxWorks 5.5 net task has lower priority	6	8	56	0.000
	RTEMS 4.6	N/A	N/A	N/A	N/A
Public Network	Linux 2.6.13 preemptive	8	13	78	0.000
	vxWorks 5.5 net task has higher priority	6	9	60	0.000
	vxWorks 5.5 net task has lower priority	6	9	59	0.000
	RTEMS 4.6	N/A	N/A	N/A	N/A

Table 1: Interrupt Latency

A typical DAQ system



0) Detectors

1) Front-end systems
Analog-digital conversion
Detector readout link
Feature extraction

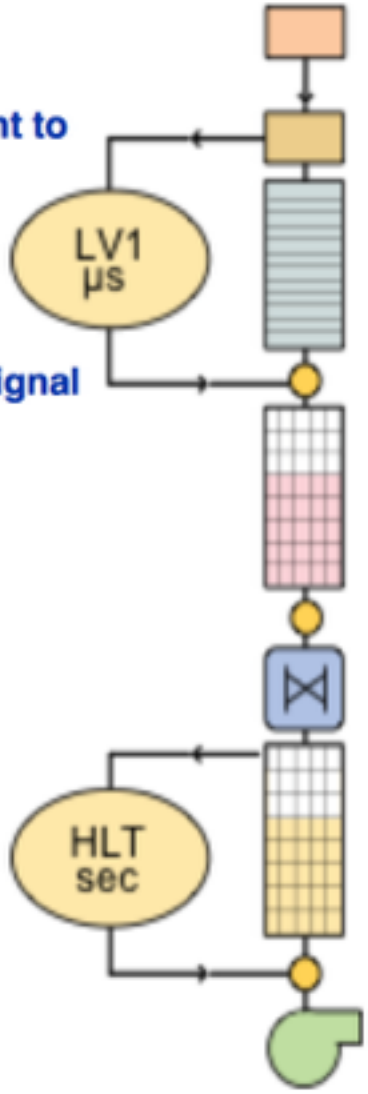
2) DAQ readout Links
Interface from custom to commercial electronics

3) Event Building
Network

4) Event Filter (High Level Trigger)
Thousands of CPU cores

5) Local mass storage

Trigger and DAQ interaction



Coarse data sent to trigger

Pipelined electronics process every bx

Level-1 accept signal

High Level trigger processing (software)

At beam crossing (bx) frequency
For the LHC: 40 MHz

Full data stored in a pipeline

Accepted event fragments stored in de-randomizer buffer

Central DAQ reads accepted fragments ..

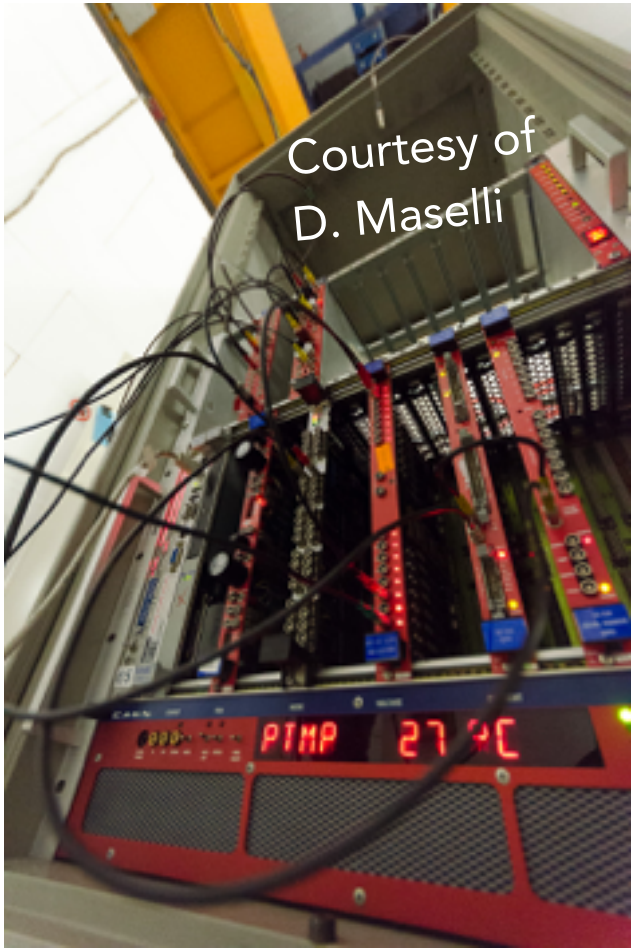
.... and builds events

Level-1 accept rate
= event building rate
CMS: 100 kHz
LHCb: 1MHz

Full events buffer during high-level trigger processing

Rate to disk:
few 100 Hz
up to kHz

from micro to macro DAQ - same logic



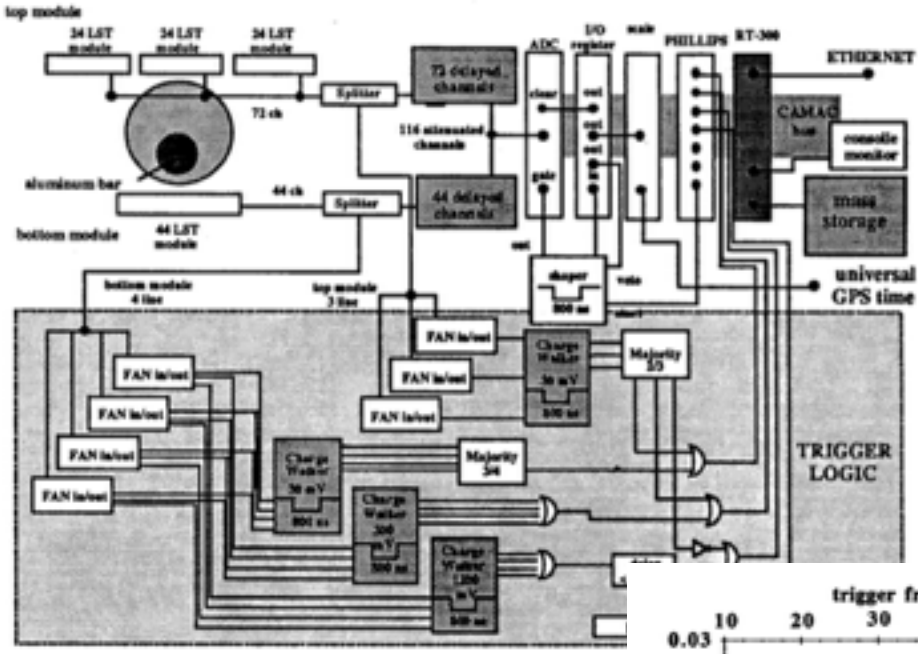
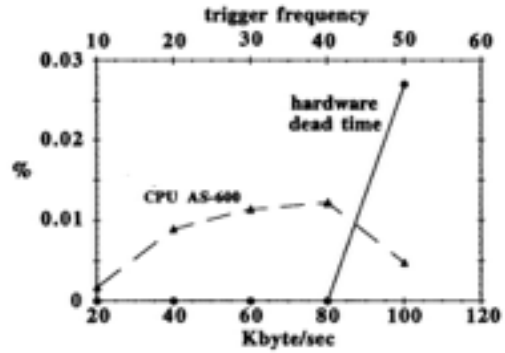
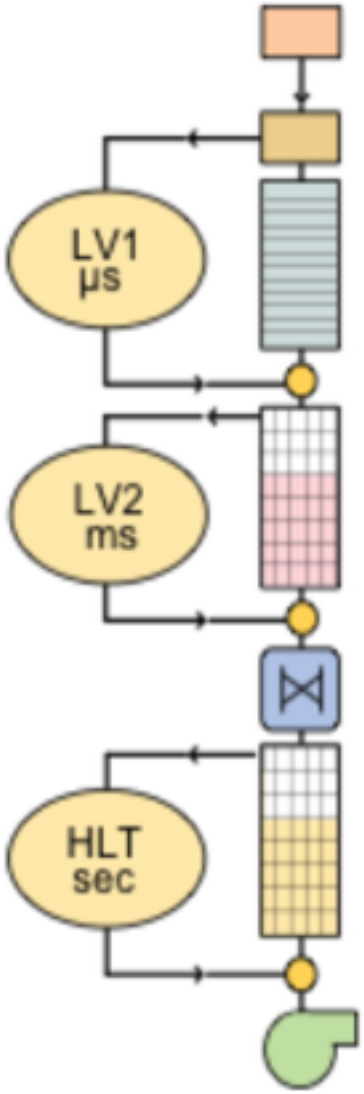


Fig. 2. Readout system.



Trigger and DAQ interaction



Beam crossing (bx) frequency
For the LHC: 40 MHz

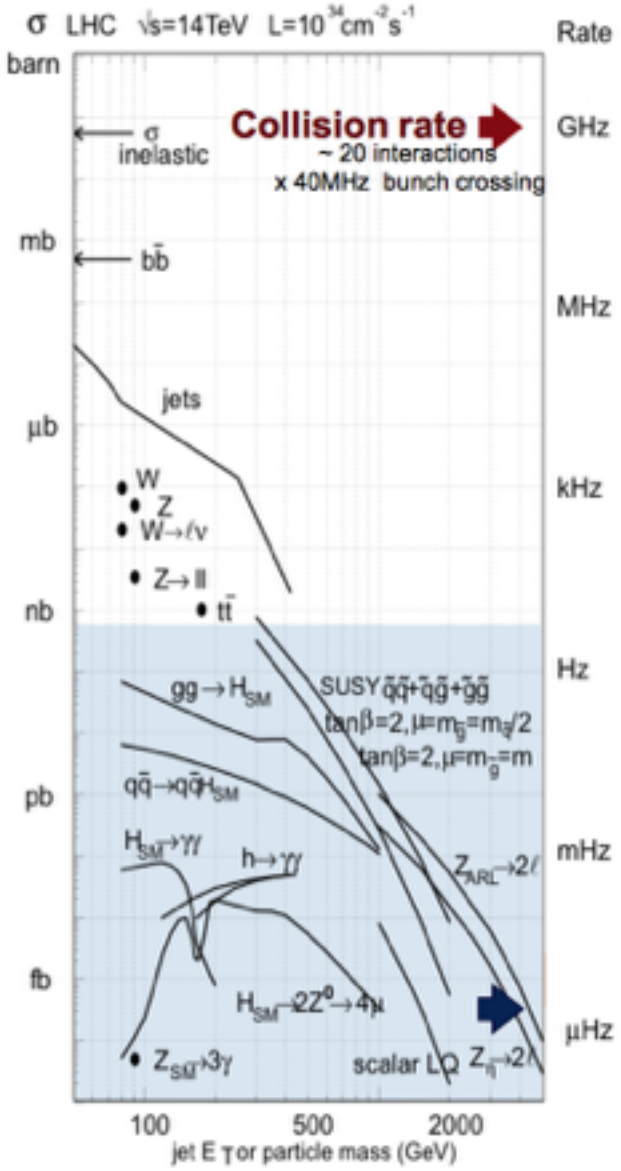
Level-1 accept rate
ATLAS: 65 KHz

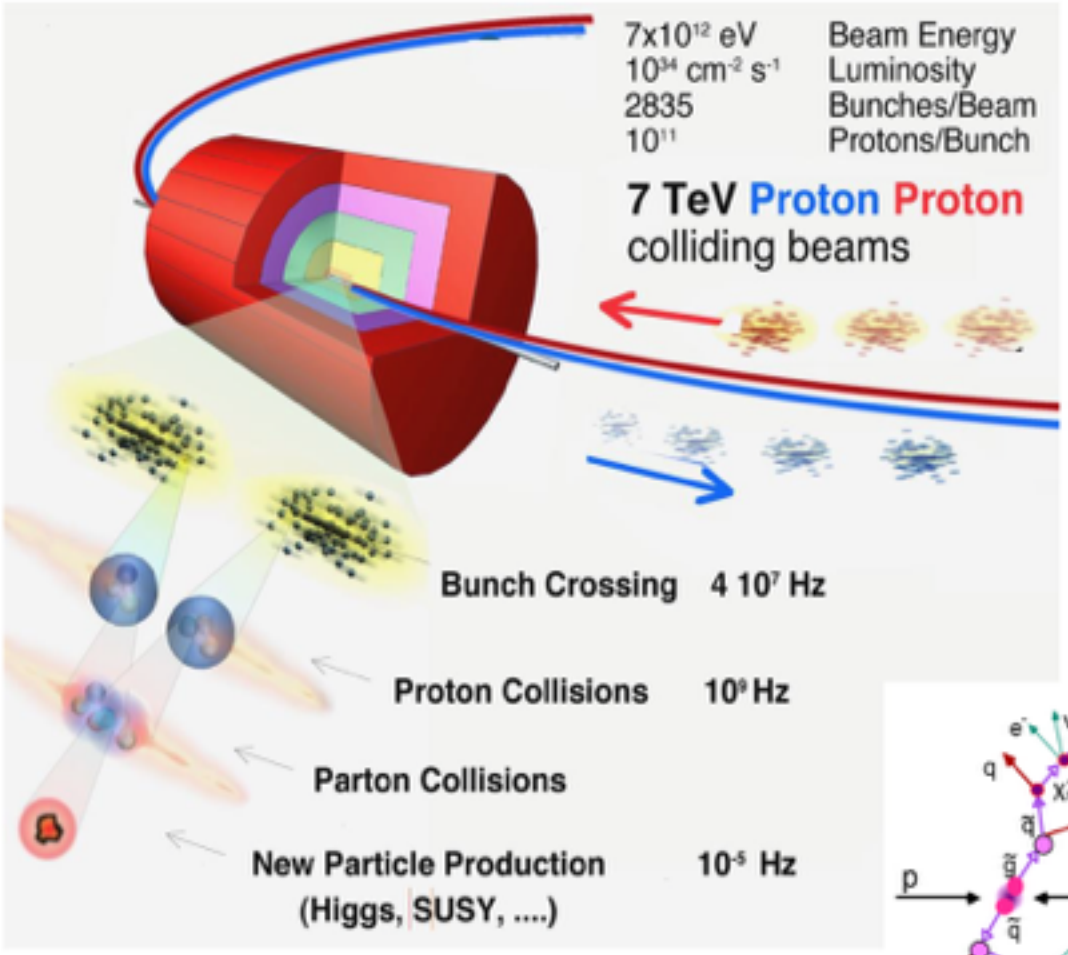
Some experiments have multiple trigger levels

Level-2 accept rate = event building rate
ATLAS: 6 KHz

Rate to disk:
few 100 Hz
up to kHz

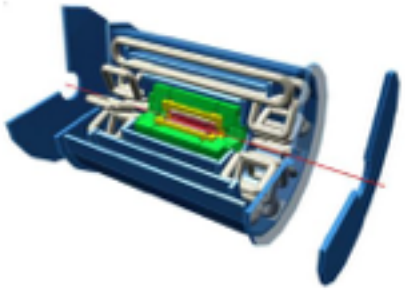
Proton-proton collisions at the LHC. Searching issue

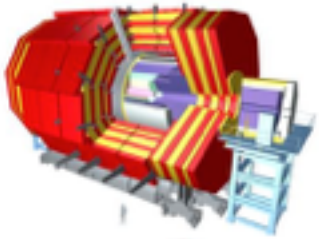


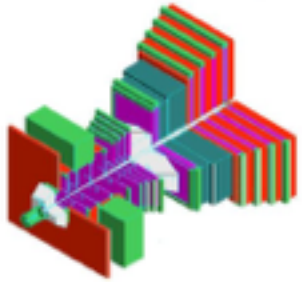


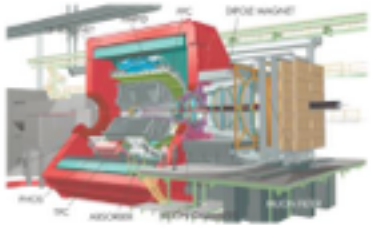
Collision Rate: ~10^9 Hz. Event Selection: ~1/10^{13}

Trigger/DAQ parameters (Run 1)



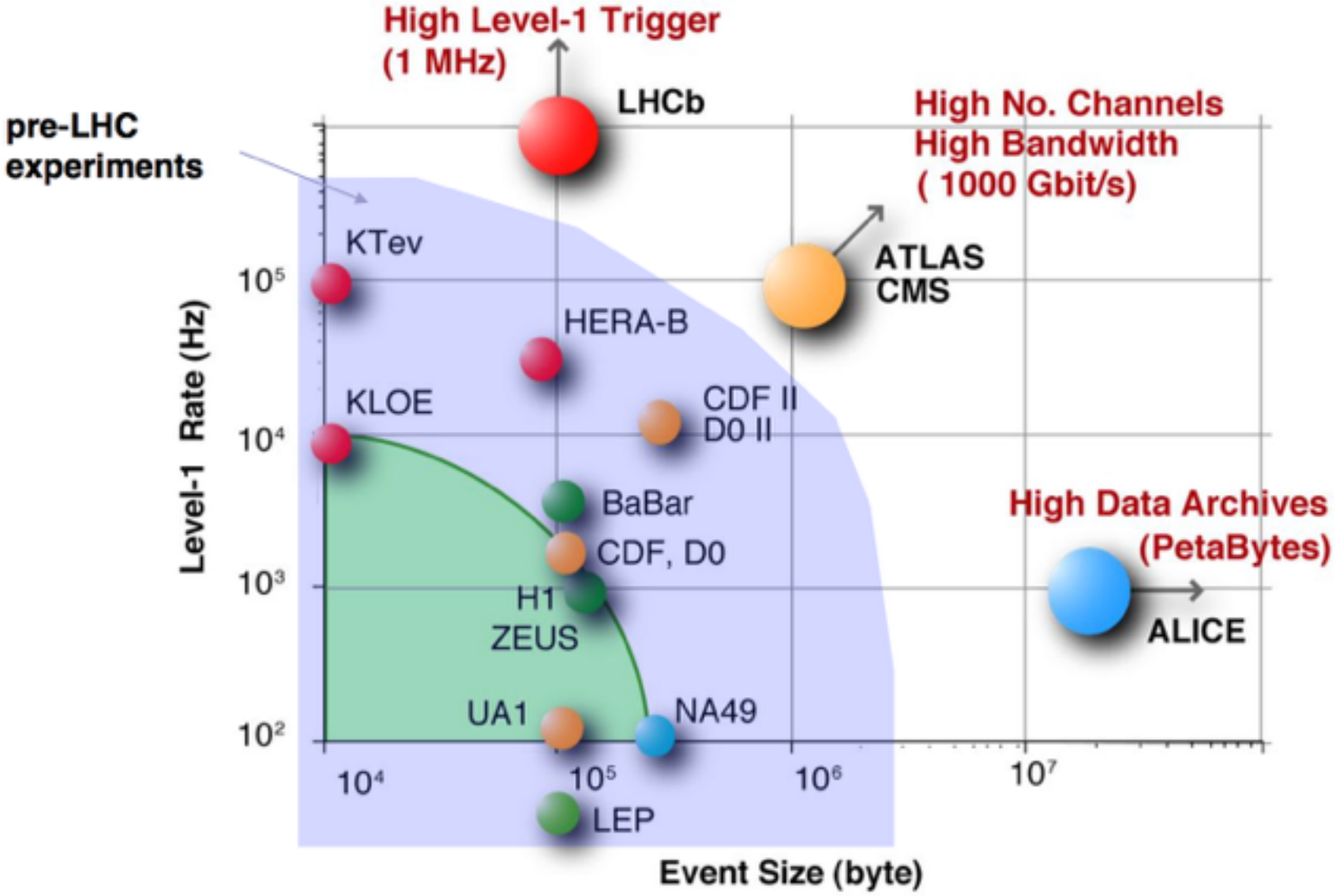






	High Level Trigger			
No.Levels	**Lvl 0,1,2**	**Event**	**Evt Build.**	**HLT Out**
				(design)
Trigger	Rate (Hz)	Size (Byte)	Bandw.(GB/s)	MB/s (Event/s)
3	LV-1 **10^5** LV-2 **$6x10^3$**	**1.5 MB**	**5.25**	**300** (200)
2	LV-1 **10^5**	**1.0 MB** Pb-Pb 1500MB/s	**100**	**300** (200)
2	LV-0 **10^6**	**30 kB**	**40**	**60** (2 kHz)
4	Pb-Pb **500**	**70 MB**	**2**	**1250** (100)

LHC experiments: Lvl 1 rate vs size



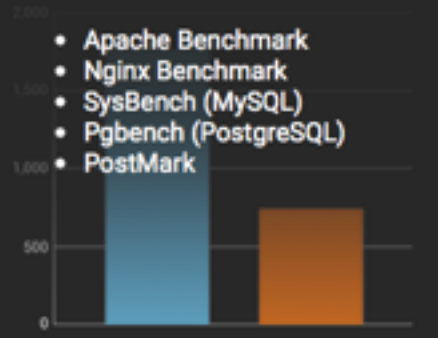
All Cloud Performance Testing Benchmarks Are Not the Same

Learn about Cloud Computing IaaS performance testing and benchmarking.

Cloud performance testing is important. Whether deploying applications or moving workloads to the cloud, your users will judge your team's ability on the actual performance. If it's too slow, you risk low productivity or even a lack of use. And, let's face it - there are plenty of marketing claims that say one cloud is faster than the other cloud.

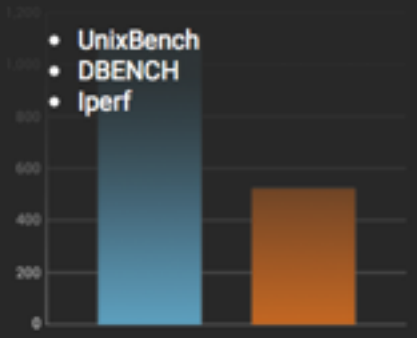
You've also read about or experienced inconsistent cloud performance. We have the results and share the methods here you need to make an assessment.

Workload Based Cloud Computing Performance Benchmarks



See the Results ▼

Synthetic Cloud Computing Performance Benchmarks

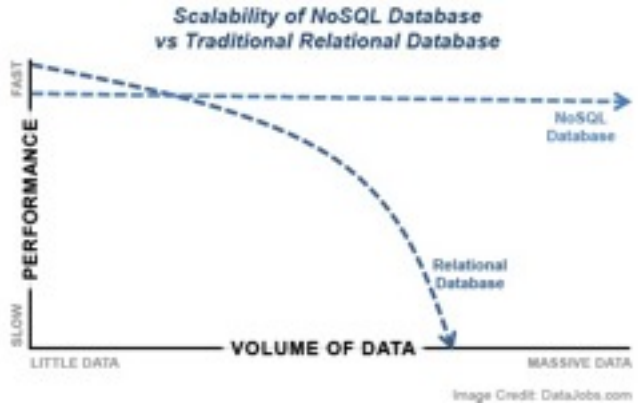


See the Results ▼

ICHAOS

Cloud - Types of testing

- Stress: Stress Test is used to determine ability of application to maintain a certain level of effectiveness beyond breaking point.
- Load: Load testing of an application involves creation of heavy user traffic, and measuring its response
- Performance: Finding out thresholds, bottlenecks & limitations is a part of performance testing
- Functional: internet and non-internet applications
- Compatibility: on demand, making compatibility testing effortless
- Browser performance: application support
- Latency



Bibliography

- http://en.wikipedia.org/wiki/Computer_performance
- http://en.wikipedia.org/wiki/Performance_measurement
- http://en.wikipedia.org/wiki/Benchmark_(computing)
- http://en.wikipedia.org/wiki/Latency_(engineering)
- Overview of the Experimental Physics and Industrial Control System: EPICS - Stephen A. Lewis - Lawrence Berkeley National Laboratory
- TANGO COLLABORATION NEWS J. Meyer (ESRF) on behalf of the Tango community ALBA, DESY, ELETTRA, ESRF, SOLEIL
- Cybersecurity through Real-Time Distributed Control Systems - Oak Ridge National Laboratory - ORNL/TM-2010/30
- AIDA annual meeting - Student Tutorial - https://indico.cern.ch/event/282492/speakers
- Hig-perfomance computing - Computer Physics Communication - 97 (1996) 16-35
- REAL-TIME PERFORMANCE MEASUREMENTS OF EPICS IOCCORE - 10th ICALEPCS Int. Conf. on Accelerator & Large Expt. Physics Control Systems. Geneva, 10 - 14 Oct 2005, PO2.075-5 (2005)
- http://en.wikipedia.org/wiki/Cloud_testing