高能物理实验中的数据获取

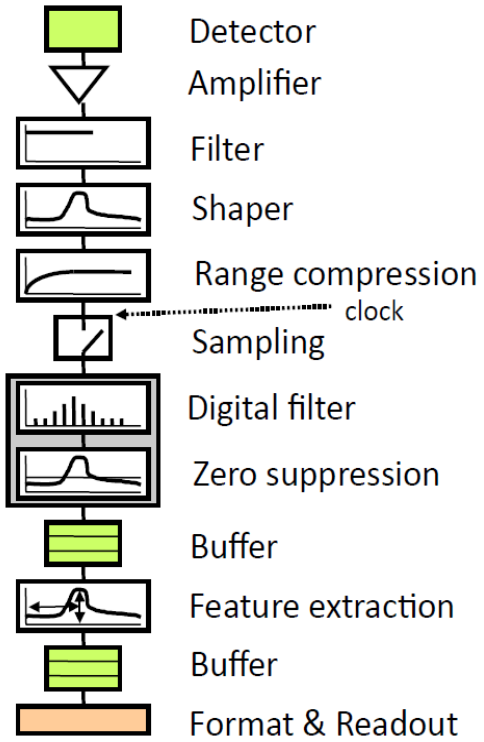
朱科军

2017年11月17日

中国科学院
高能物理研究所、中国科学技术大学

内容

- 数据获取简单介绍
- 数据获取在高能物理实验的位置
- 数据获取的发展
- 几个大型实验的数据获取
- **BESIII数据获取系统**
- 未来发展趋势



Introduction to Data Acquisition, Excellence in Detectors and Instrumentation Technologies (EDIT) school 2015, Niko Neufeld, CERN-PH

DAQ – Transporting & Filtering Data from 1 PB/s to 600 MB/s, Openlab Summer Students Lectures, July 2017, Sébastien Valat, CERN-EP

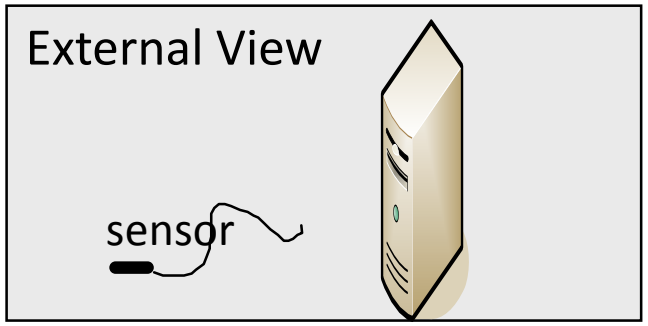
Data acquisition

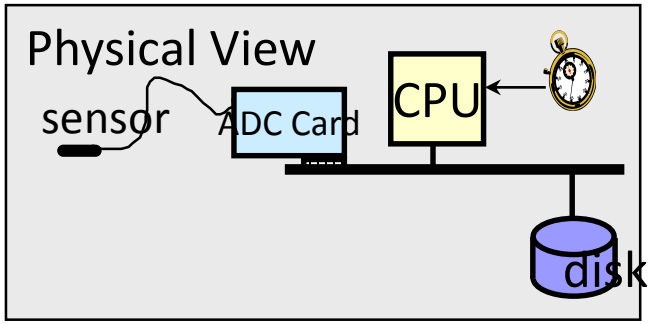
From Wikipedia, the free encyclopedia

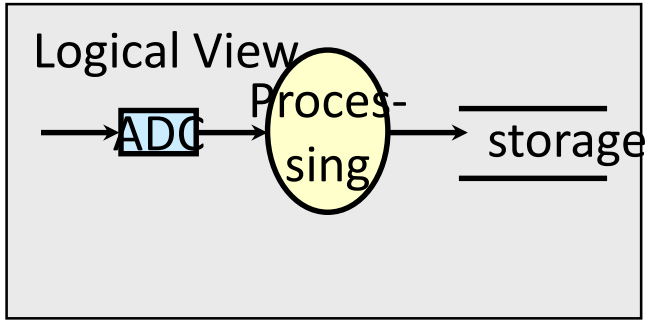
Data acquisition is the process of sampling signals that measure real world physical conditions and converting the resulting samples into digital numeric values that can be manipulated by a computer. Data acquisition systems (abbreviated with the acronym **DAS** or **DAQ**) typically **convert analog waveforms into digital values for processing**. The components of data acquisition systems include:

- Sensors that convert physical parameters to electrical signals.
- Signal conditioning circuitry to convert sensor signals into a form that can be converted to digital values.
- Analog-to-digital converters, which convert conditioned sensor signals to digital values.

Data acquisition applications are usually controlled by software programs developed using various general purpose programming languages such as Assembly, BASIC, C/C++/C#, Fortran, Java, LabVIEW, Lisp, Pascal, etc. Stand-alone data acquisition systems are often called data loggers.





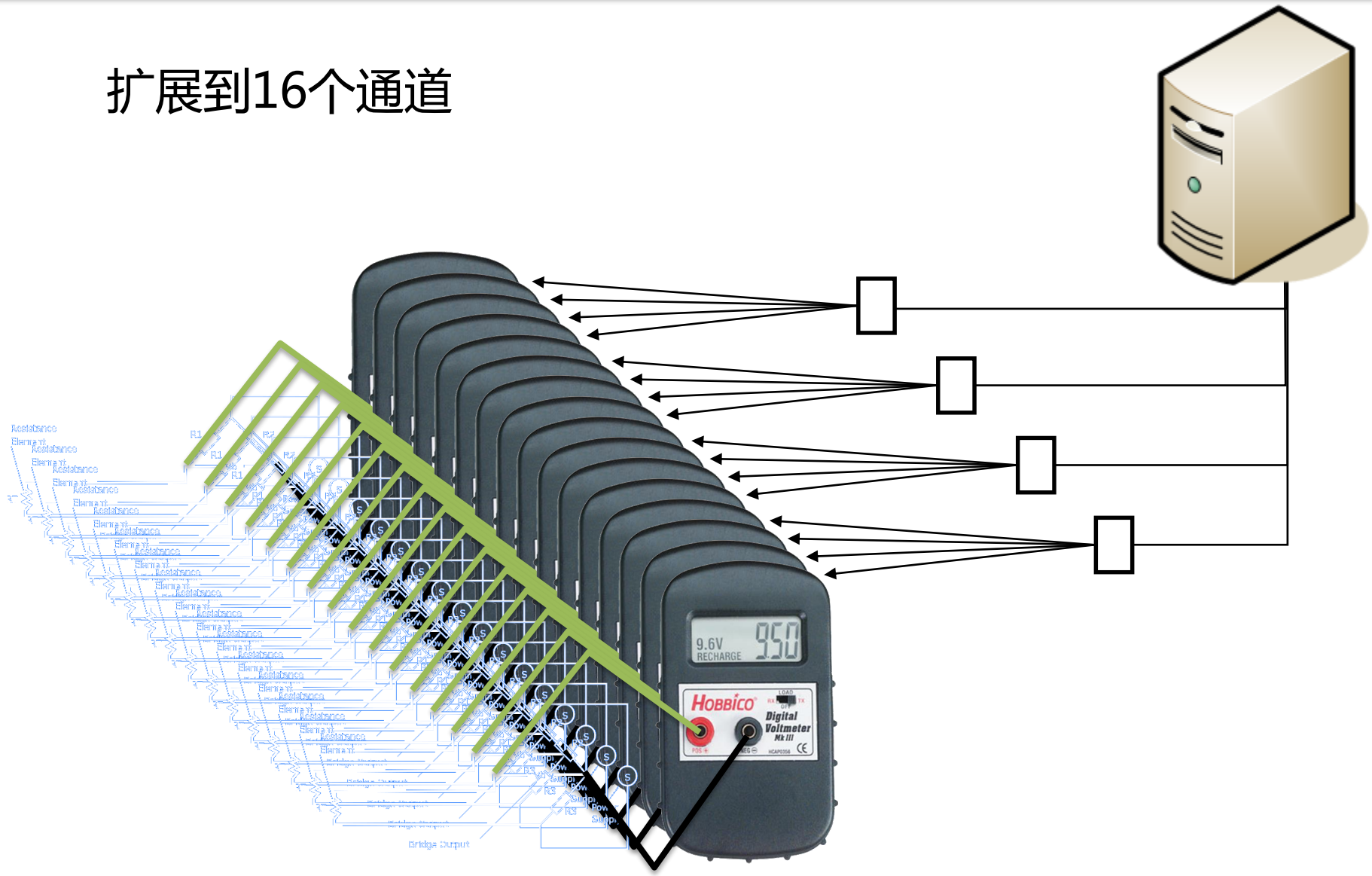


简单的DAQ例子

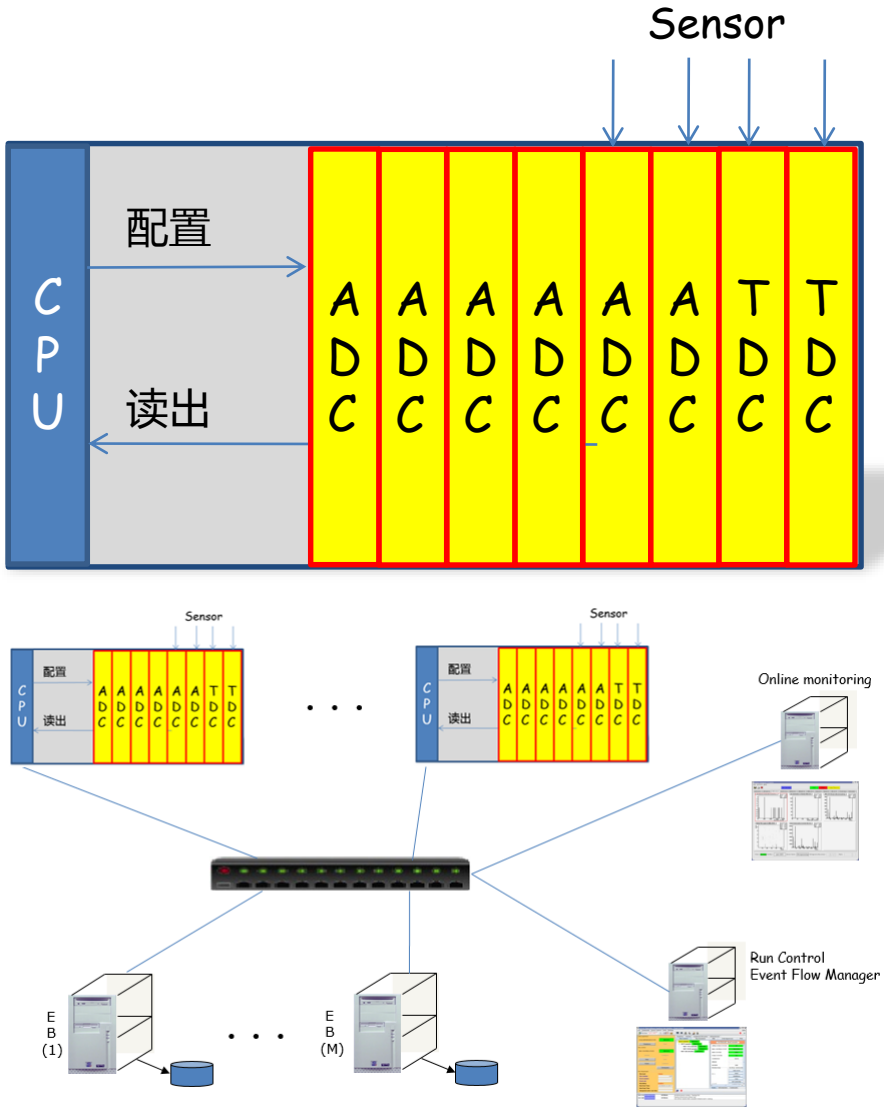
1个通道的温度测量

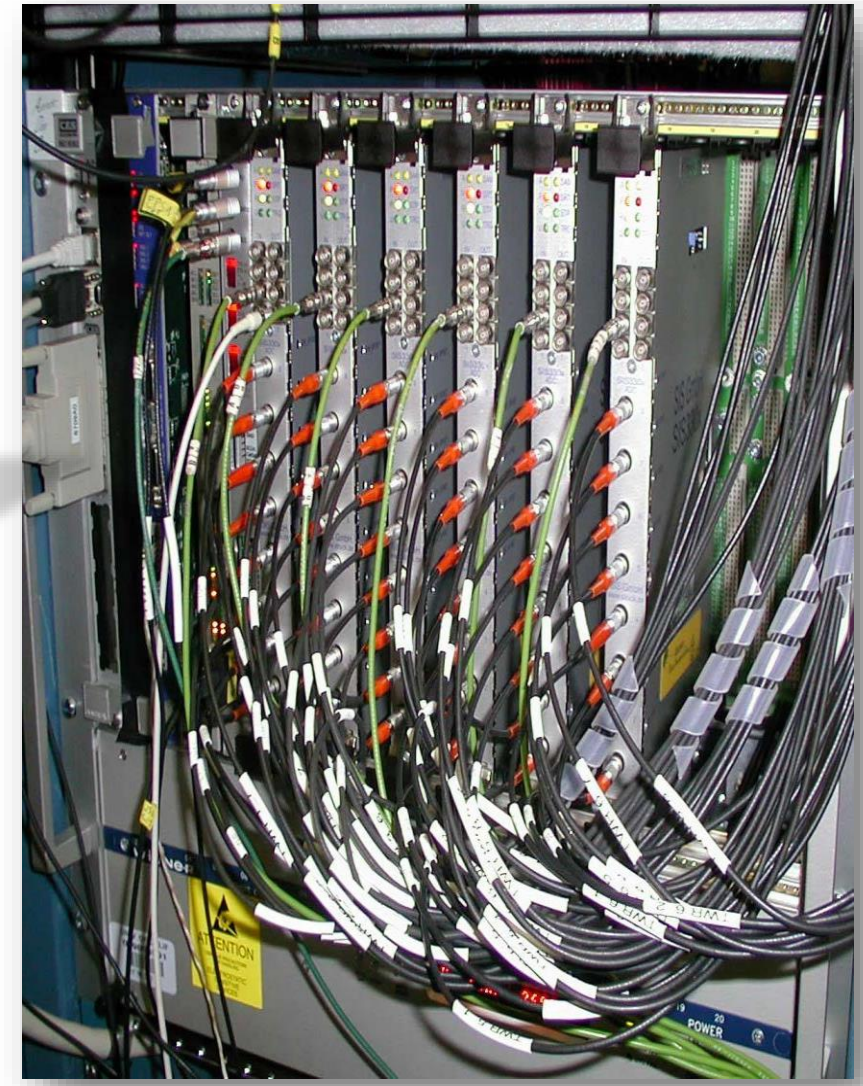
简单的DAQ例子

扩展到16个通道

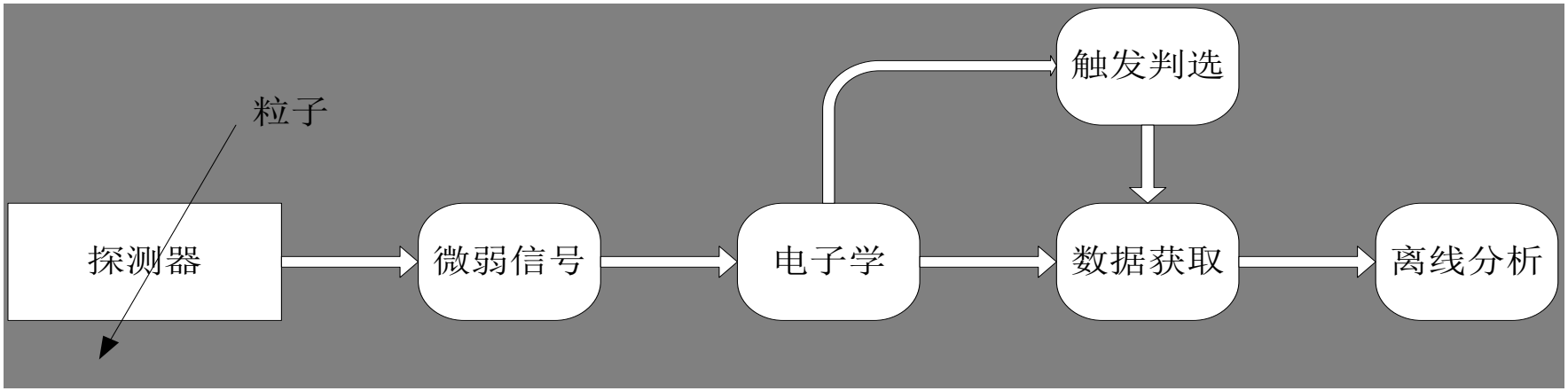


简单的DAQ例子


Sensor
CPU
配置
读出
ADC
ADC
ADC
ADC
ADC
ADC
TDC
TDC
Sensor
Sensor
Online monitoring
Run Control
Event Flow Manager
EB (1)
EB (M)



DAQ在高能物理实验中的位置



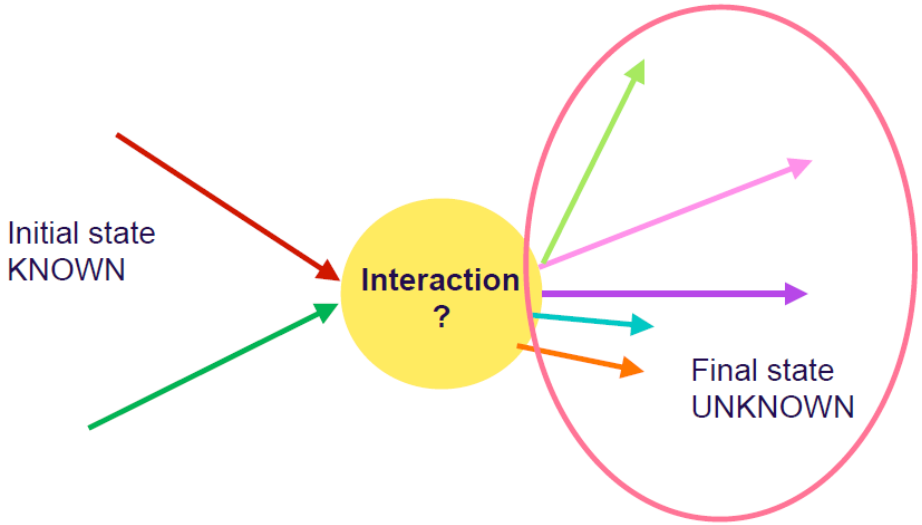
电子学：将探测器输出的信号放大、成形等处理后数字化，将信息暂存并作数据预处理；

触发判选：选择满足物理条件的好事例，压缩本底事例；

数据获取和在线分析：读出通过触发判选的好事例数据，以数字形式记录下来；给出反映探测器性能的各种统计图形以及所获事例的分类统计图形，监测探测器与电子学工作状态；

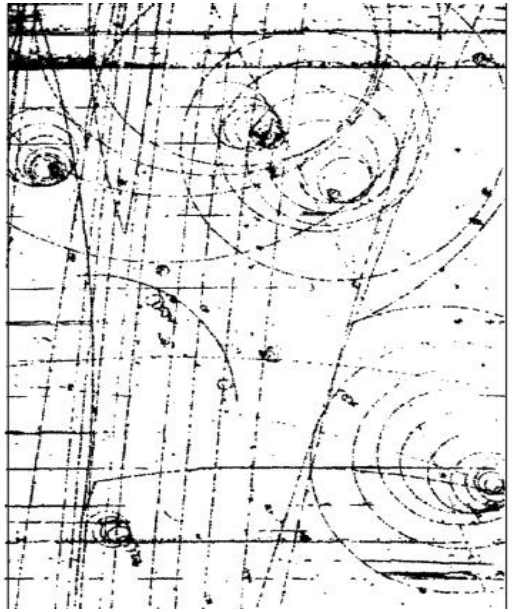
离线数据分析：将在线机上记录下来的数据在离线机上进行分析和处理，把数据还原为粒子种类、能量、动量等物理量。

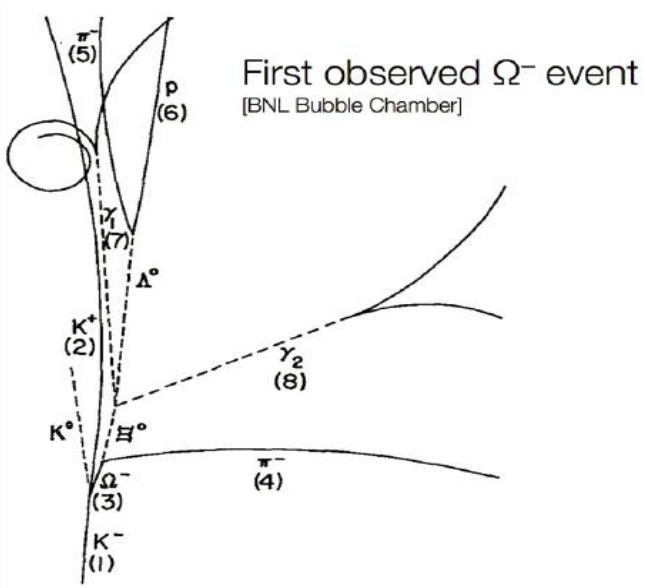
高能物理实验



记录相互作用后的产物

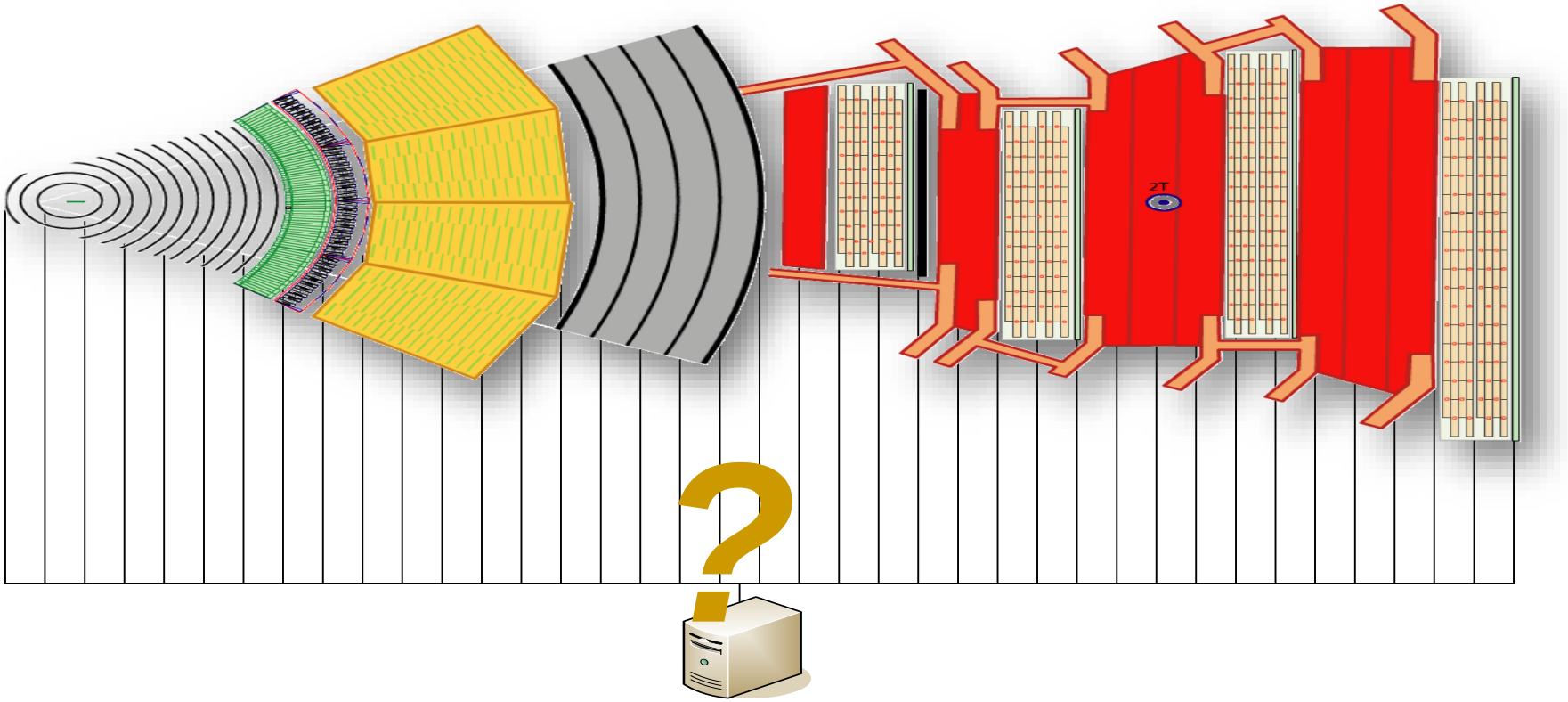
- 测量位置信息/运动径迹
- 测量动量/能量
- 测量时间信息
- ……



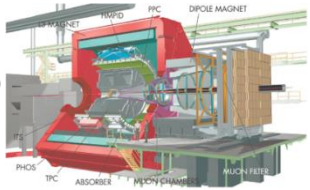
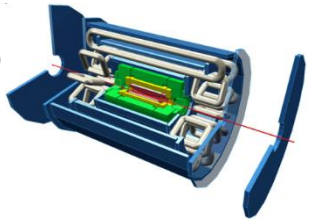
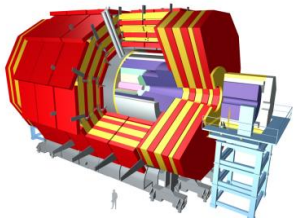
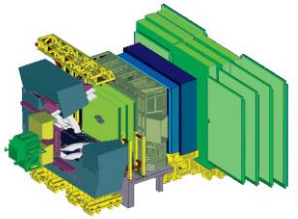


Moving on to Bigger Things

- **15 million** detector channels **@ 40 MHz**
- = ~15,000,000 * 40,000,000 bytes = ~ **600 TB/sec**
- We **cannot store all of this** !

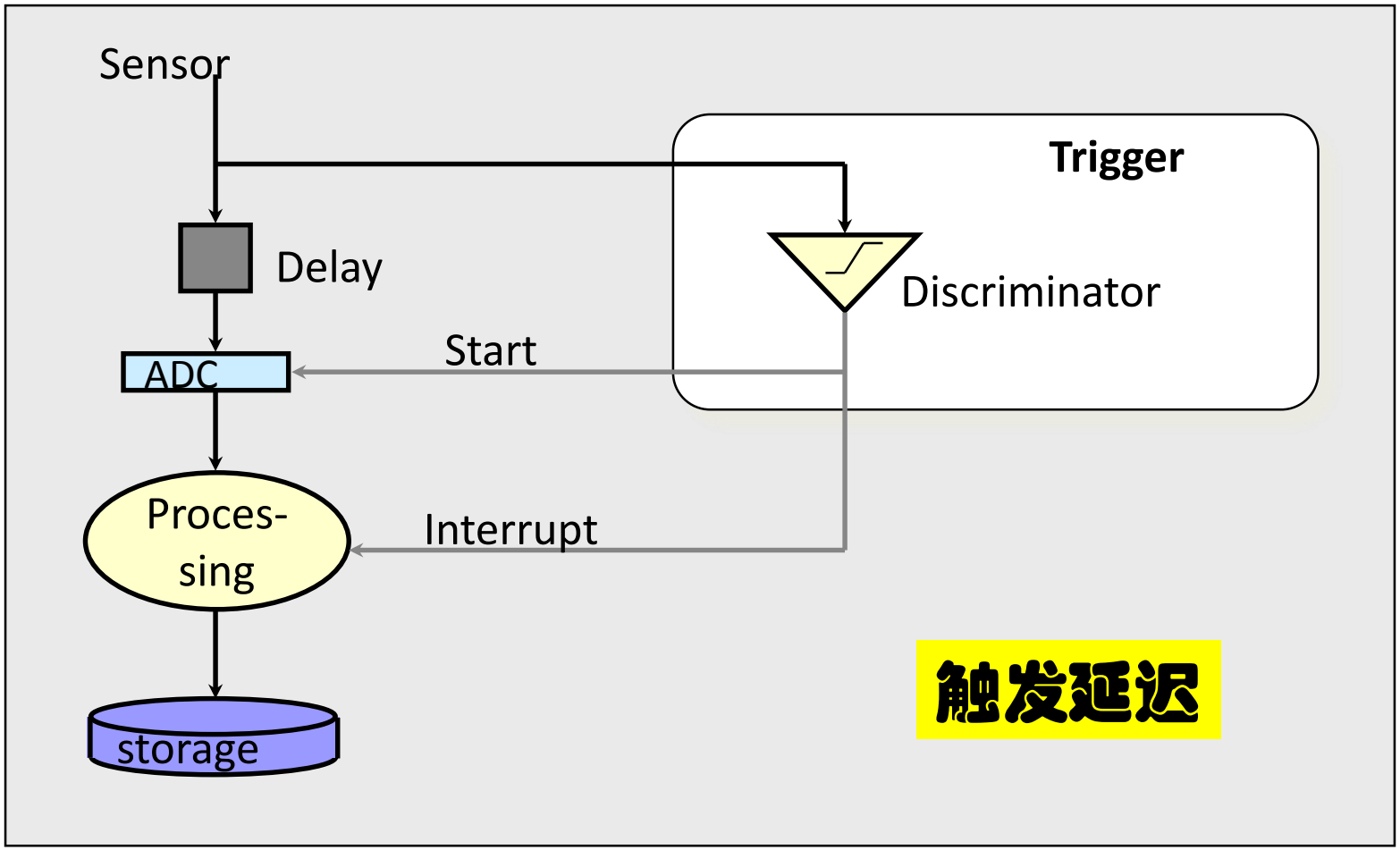


Trigger/DAQ parameters

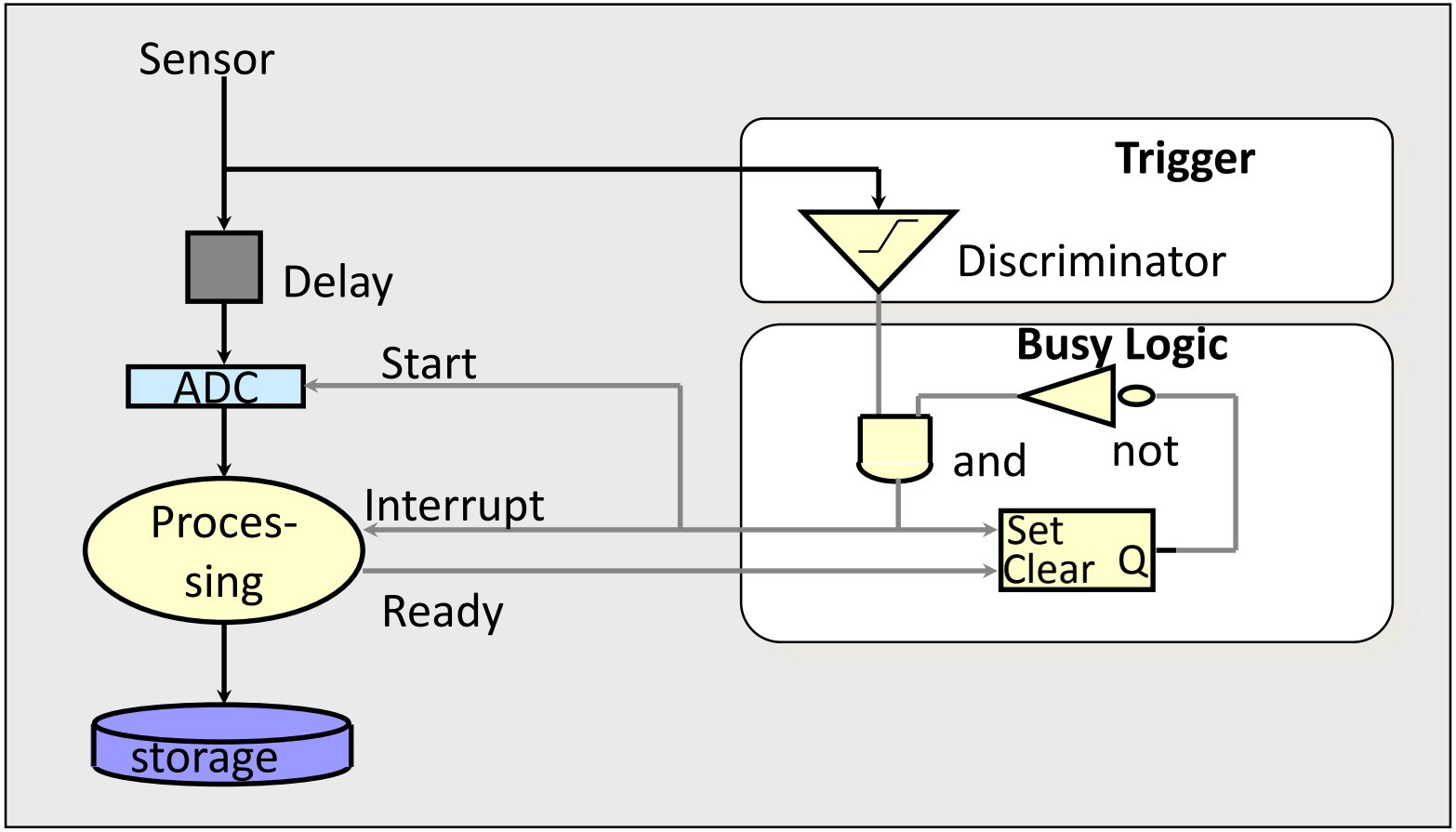
	No.Levels Trigger	Level-0,1,2 Rate (Hz)	Event Size (Byte)	Readout Bandw.(GB/s)	HLT Out MB/s (Event/s)
ALICE 	4	Pb-Pb **500** p-p **10^3**	**$5x10^7$** **$2x10^6$**	**25**	**1250** (10^2) **200** (10^2)
ATLAS 	3	LV-1 **10^5** LV-2 **$3x10^3$**	**$1.5x10^6$**	**4.5**	**300** ($2x10^2$)
CMS 	2	LV-1 **10^5**	**10^6**	**100**	**~100** (10^2)
LHCb 	2	LV-0 **10^6**	**$5x10^4$**	**50**	**600** ($1.2x10^4$)

数据率和触发率

简单的DAQ例子_(触发)

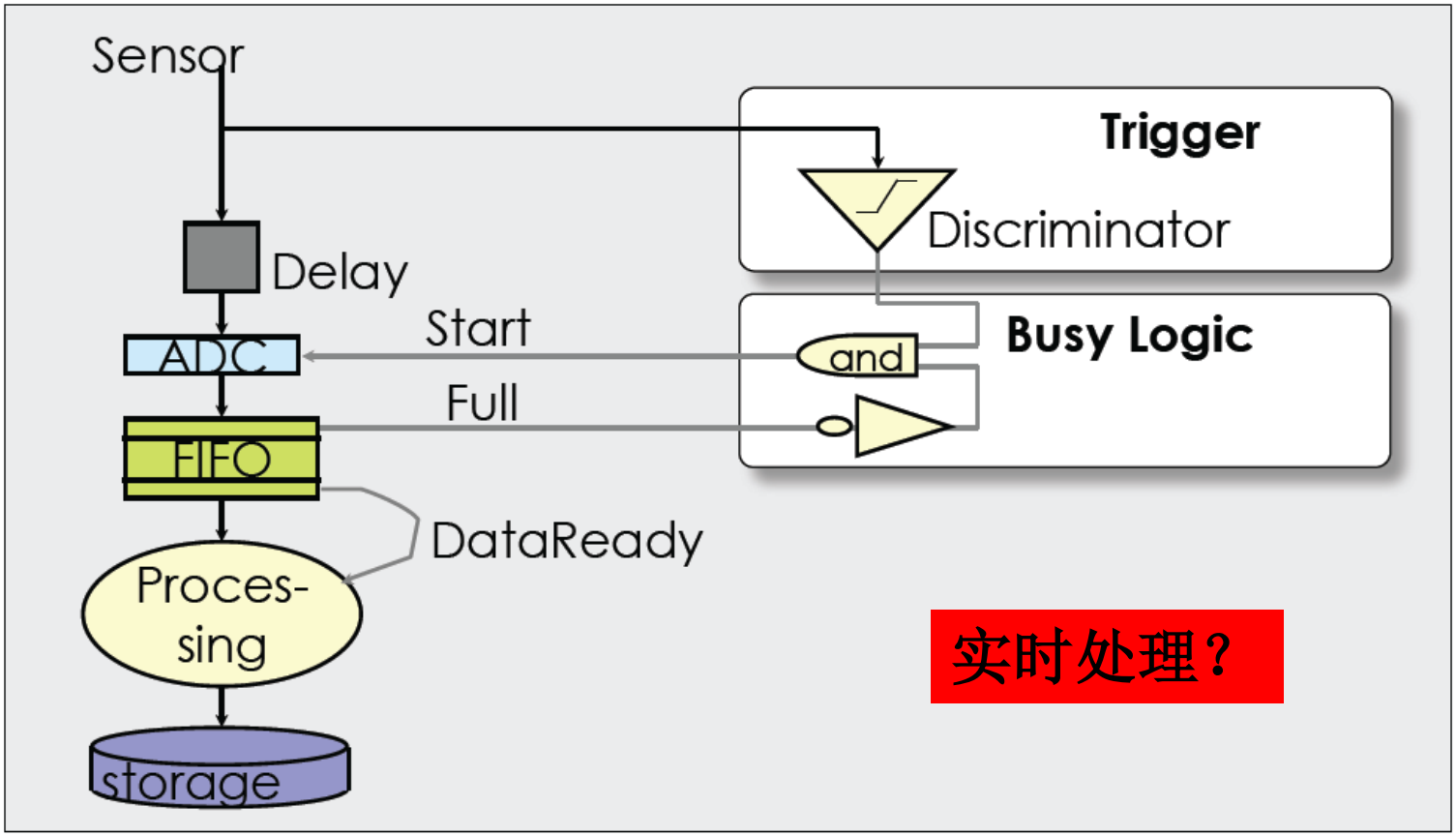

Sensor
Trigger
Delay
Discriminator
Start
ADC
Proces-
sing
Interrupt
storage
触发延迟

简单的DAQ例子_(忙)



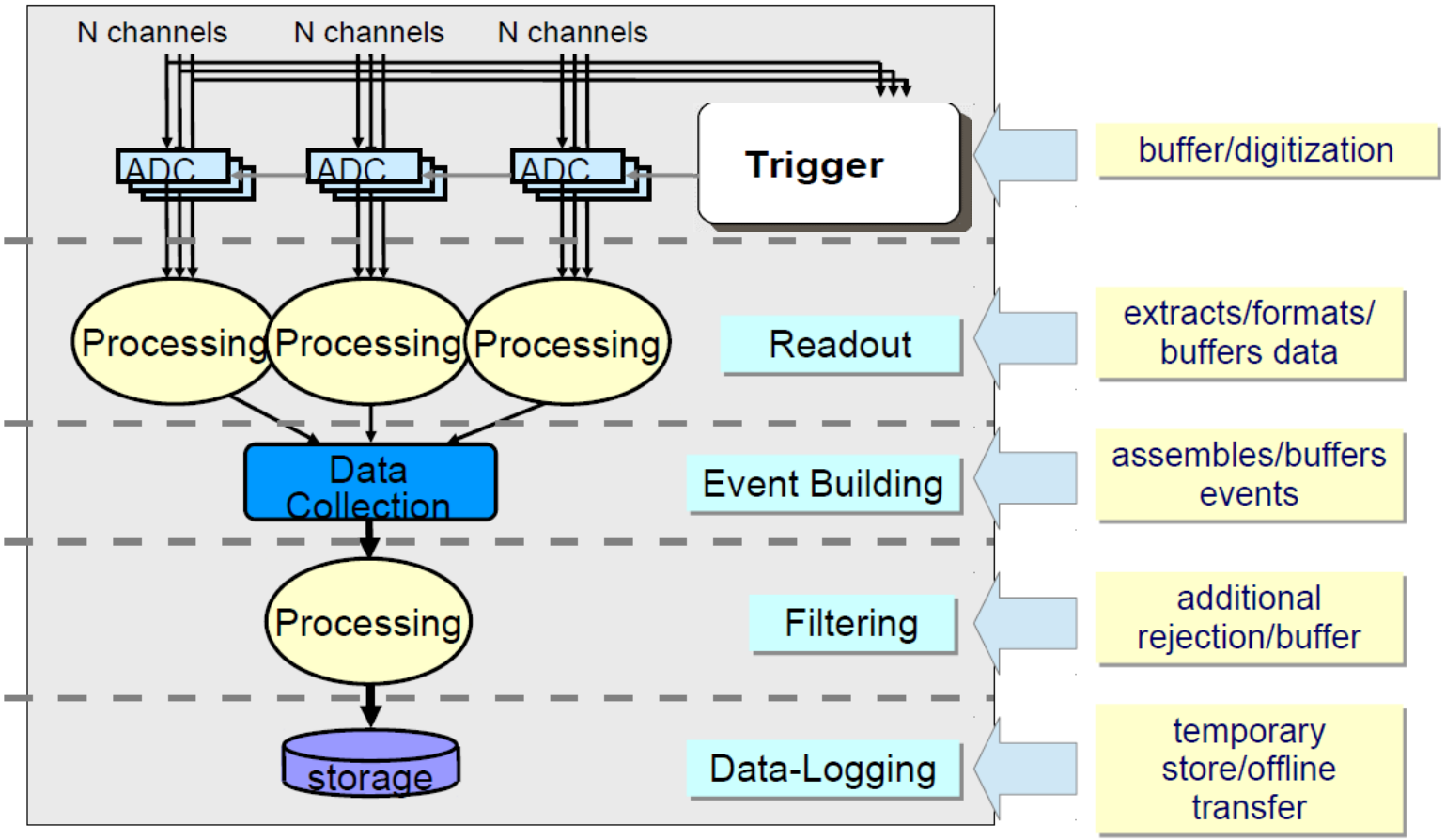
new data: 中断/查询方式，延时？同步模式

简单的DAQ例子_(FIFO)



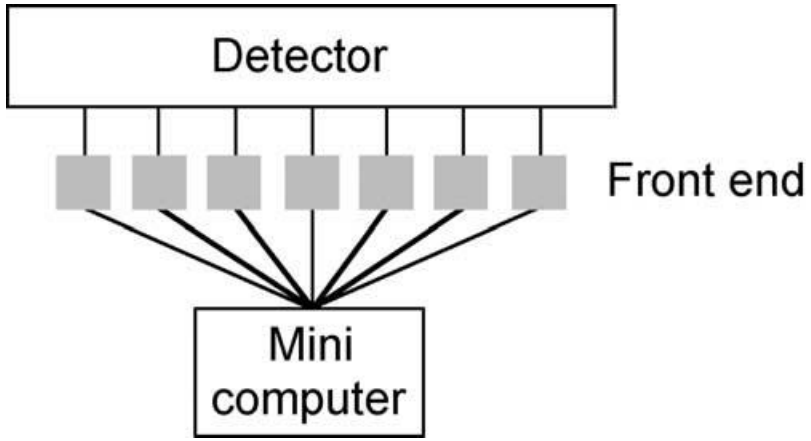
FIFO的作用：去随机化，减少死时间，提高性能
尽量少的内存拷贝，通过指针传递

多通道DAQ

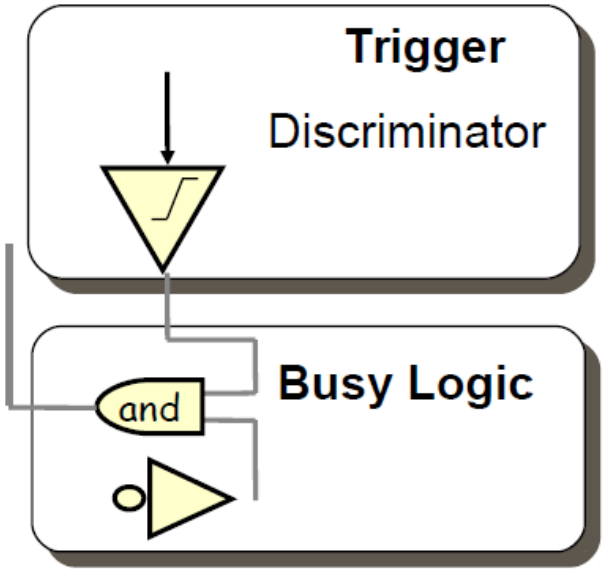

N channels
N channels
N channels
ADC
ADC
ADC
Trigger
buffer/digitization
Processing
Processing
Processing
Readout
extracts/formats/
buffers data
Data
Collection
Event Building
assembles/buffers
events
Processing
Filtering
additional
rejection/buffer
storage
Data-Logging
temporary
store/offline
transfer

六七十年代的DAQ

- 基于NIM的前端电子学插件，通过CAMAC读出数据
- 缺少并行处理能力
- 数据处理率kB/s

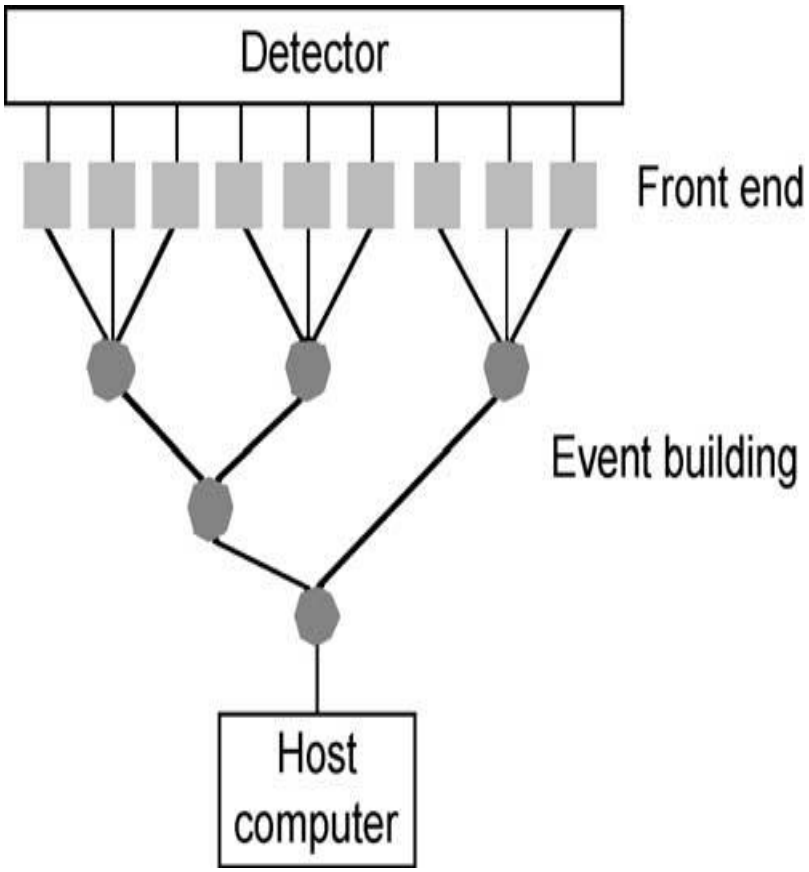


NIM（1964）



- ✓不与计算机连接，不需要软件控制插件
- ✓只提供电子学信号的逻辑处理和互连
  - 甄别，符合，放大，逻辑门，。。。
- ✓可以实现基本的trigger和busy逻辑
- ✓目前在实验室还在使用

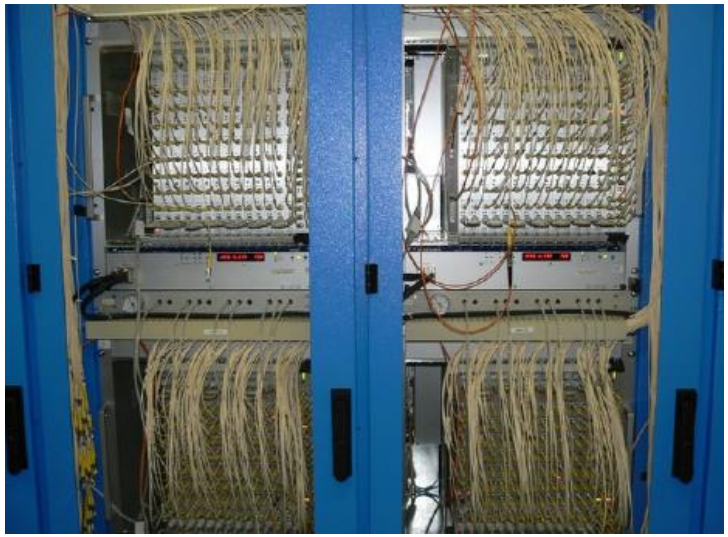
八十年代的DAQ

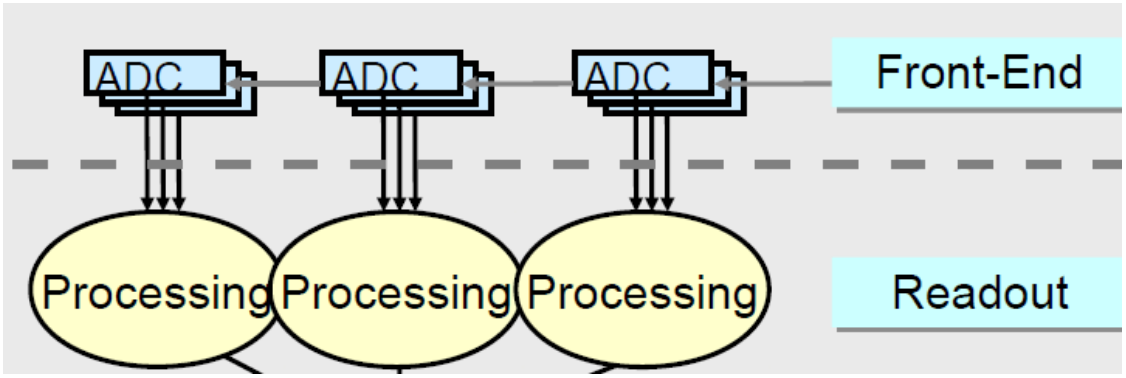


- 分布式处理
- 树状结构
- 多节点数据处理
- 数据处理率MB/s
- 存储100kB/s，需要触发

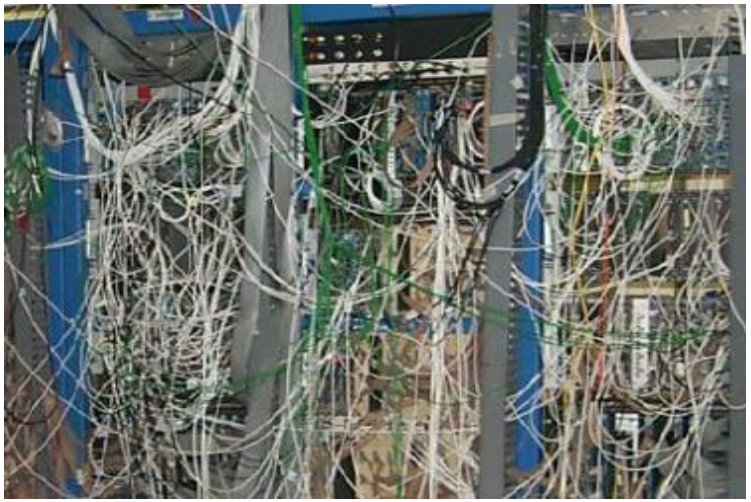
VME

- ✓插件之间通过“背板”相互通讯，提供电气、机械和通讯协议
- ✓高能物理实验中广泛使用
  - 协议简单
  - 商业插件

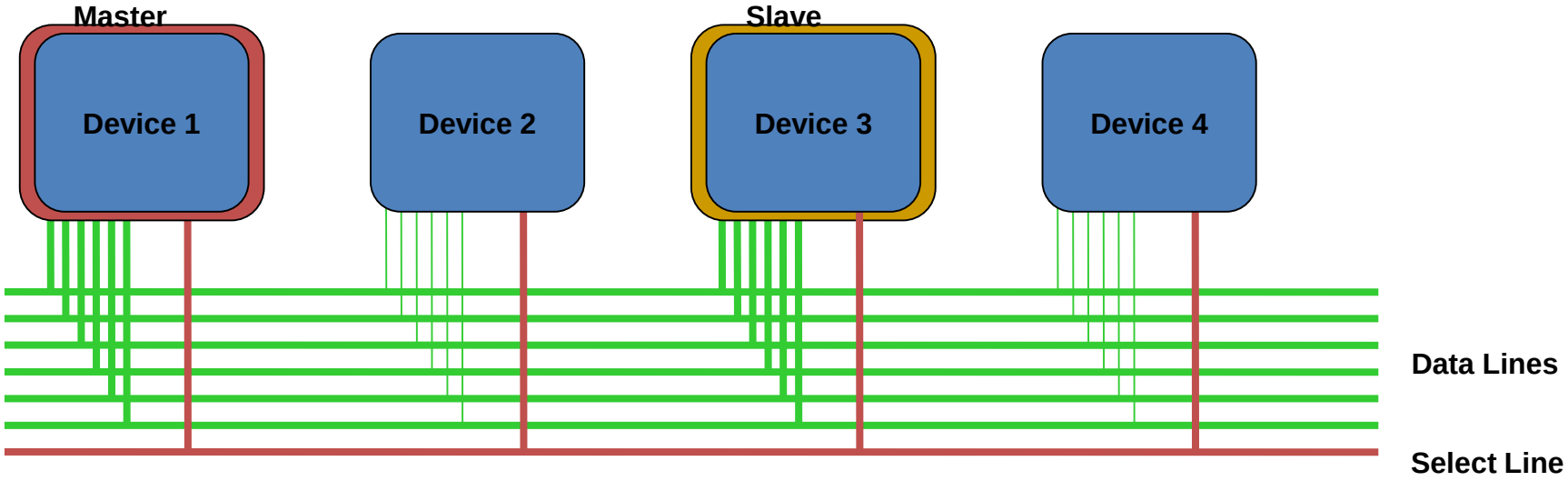




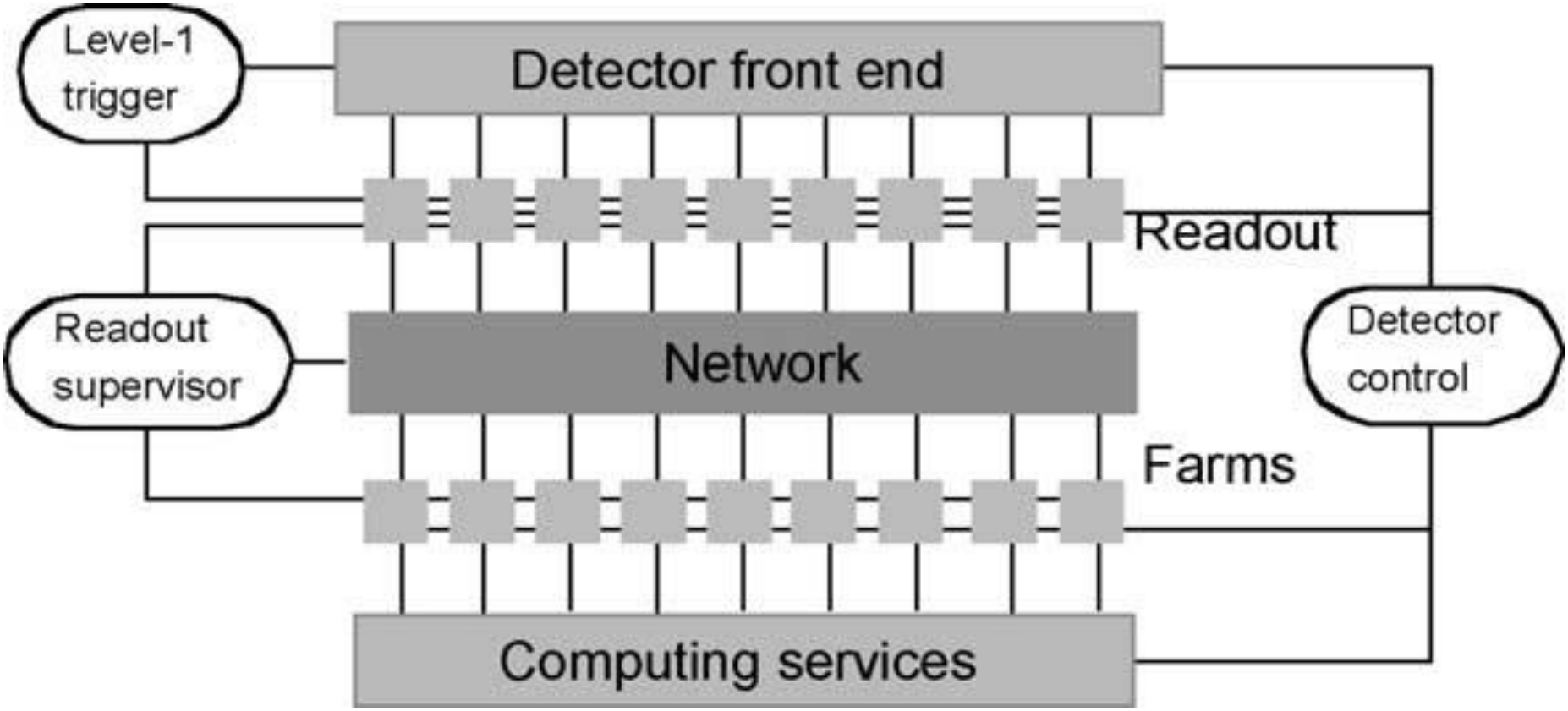
总线



- ✓ 插件通过仲裁共享总线，导致扩展性差
  - x 共享的总线带宽
  - x 有限的总线宽度
  - x 总线频率与长度成反比
  - x 插件数目与长度有关



九十年代以后的DAQ



* PC based computer farms running Linux
* Interconnected via network technologies
* Pull to push architectures → flow control

DAQ控制室

ISR. 1970

Digital display, no terminal

P-aP. 1980

A lot of persons in front of one screen

LEP. 1990

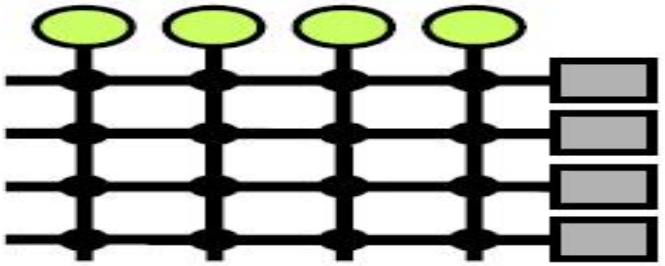
A lot of screens in front of one person

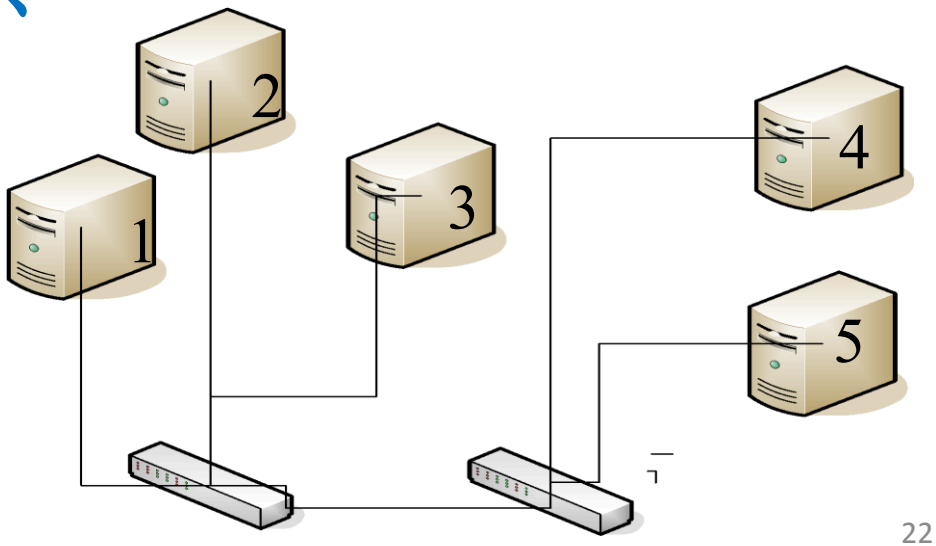
LHC 200x

The person is onto the screen

网络技术

- ✓ Ethernet，Telephone，Infiniband，……
- ✓ 网络上所有的设备通过交换机互联，都是平等的，通过点对点互联，发送和接收消息，互不干扰
- ✓ 商业设备，广泛应用
- ✓ 可扩展
- ✓ 网络阻塞，丢包？





TCP client/server example

```
struct sockaddr_in sinhim;
sinhim.sin_family      = AF_INET;
sinhim.sin_addr.s_addr = inet_addr (this_host);
sinhim.sin_port = htons (port);

if (fd = socket (AF_INET, SOCK_STREAM, 0) < 0)
{ ; // Error ! }
if (connect (fd, (struct sockaddr *)&sinhim,
             sizeof (sinhim)) < 0)
{ ; // Error ! }
```

```
while (running) {
  memcpy ((char *) &wait, (char *) &timeout,
          sizeof (struct timeval));
  if ((nsel = select (nfds, 0, &wfds,
                      0, &wait)) < 0)
  { ; // Error ! }
  else if (nsel) {
    if ((BIT_ISSET (destination, wfds))) {
      count = write (destination, buf, buflen);
      // test count…
      // > 0 (has everything been sent ?)
      // == 0 (error)
      // < 0 we had an interrupt or
      // peer closed connection
    }
  }
}
```

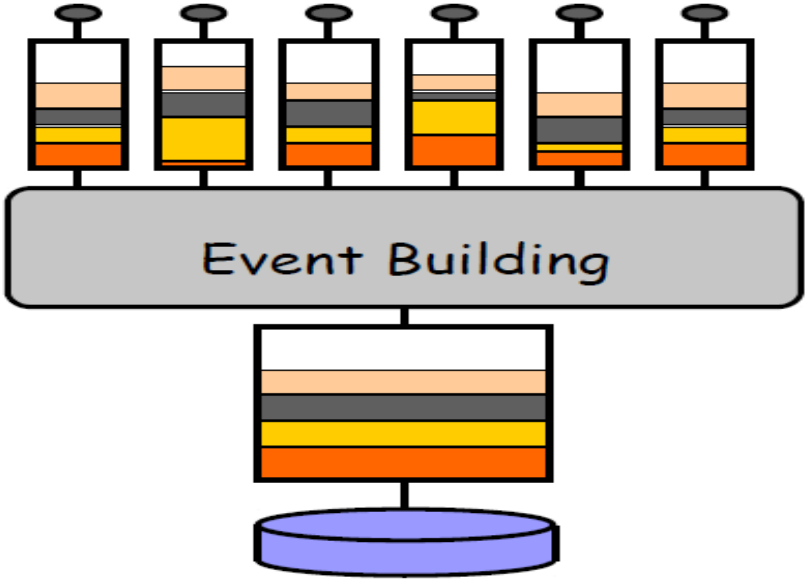
```
close (fd);
```

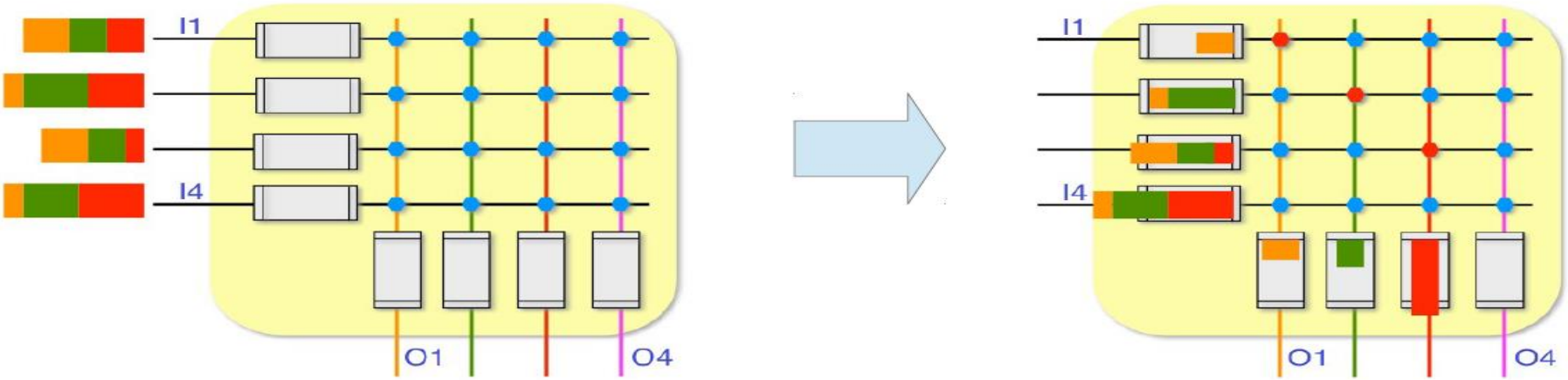
```
struct sockaddr_in sinme;
sinme.sin_family      = AF_INET;
sinme.sin_addr.s_addr = INADDR_ANY;
sinme.sin_port        = htons(ask_var->port);

fd = socket (AF_INET, SOCK_STREAM, 0);
bind (fd0, (struct sockaddr *) &sinme,
      sizeof(sinme));
listen (fd0, 5);

while (n < ns) { // we expect ns connections
  int val = sizeof(this->sinhim);
  if ((fd = accept (fd0,
       (struct sockaddr *) &sinhim, &val)) >0) {
    FD_SET (fd, &fds);
    ++ns;
  }
}
```

```
while (running) {
  if ((nsel = select( nfds, (fd_set *) &fds,
       0, 0, &wait)) [
    count = read (fd, buf_ptr, buflen);
    if (count == 0) {
      close (fd);
      // set FD bit to 0
    }
  }
}
```

```
close (fd0);
```

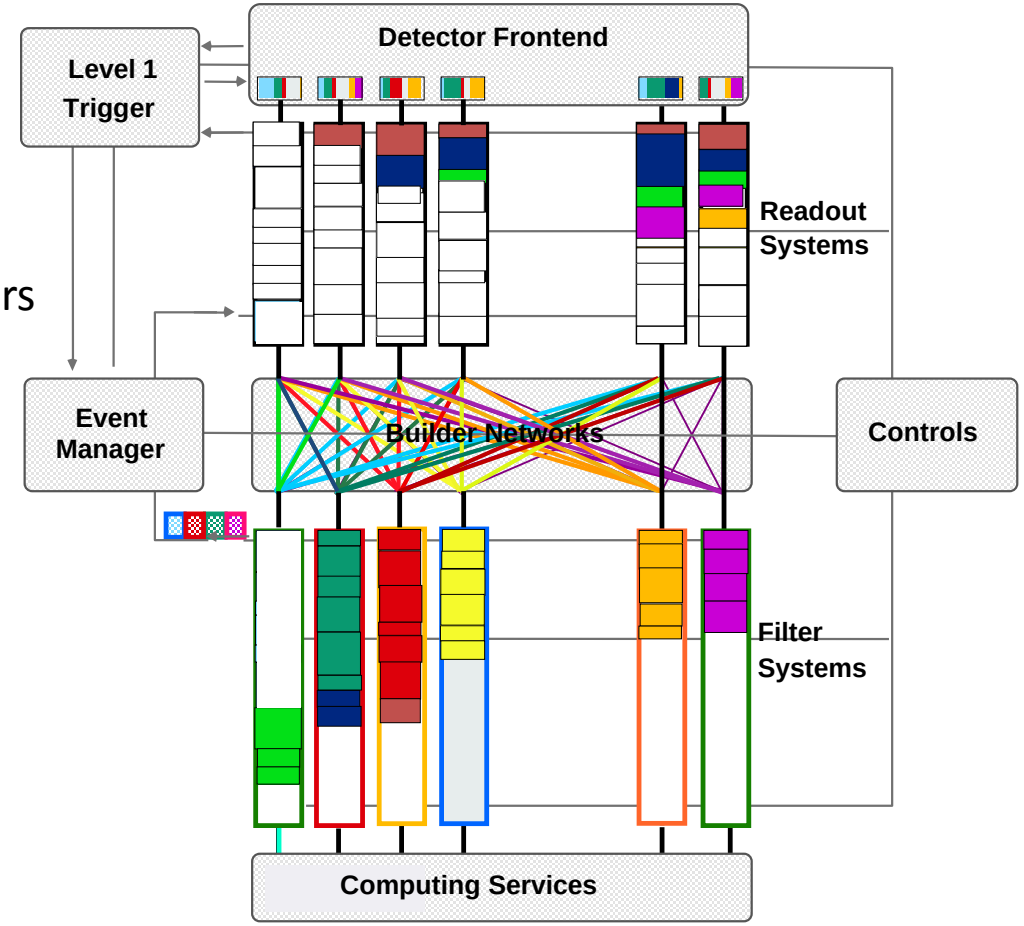
事例组装



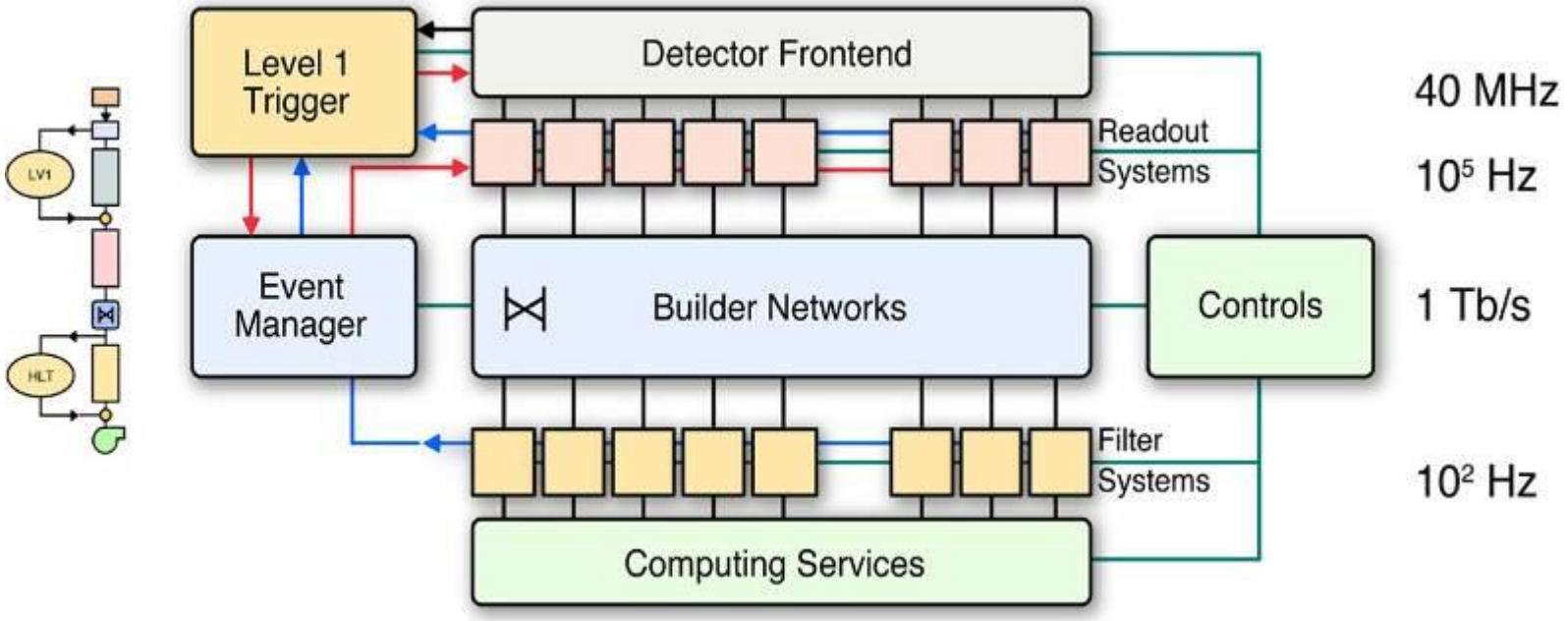


Large DAQ / Event-building

- Large detectors
  - Sub-detectors data are collected independently
    - Readout network
    - Fast data links
  - Events assembled by event builders
    - From corresponding fragments
  - Custom devices used
    - In FEE
    - In low-level triggers
  - COTS used
    - In high-level triggers
    - In event builder network
- DAQ system
  - data flow & control
  - distributed & asynchronous



CMS DAQ

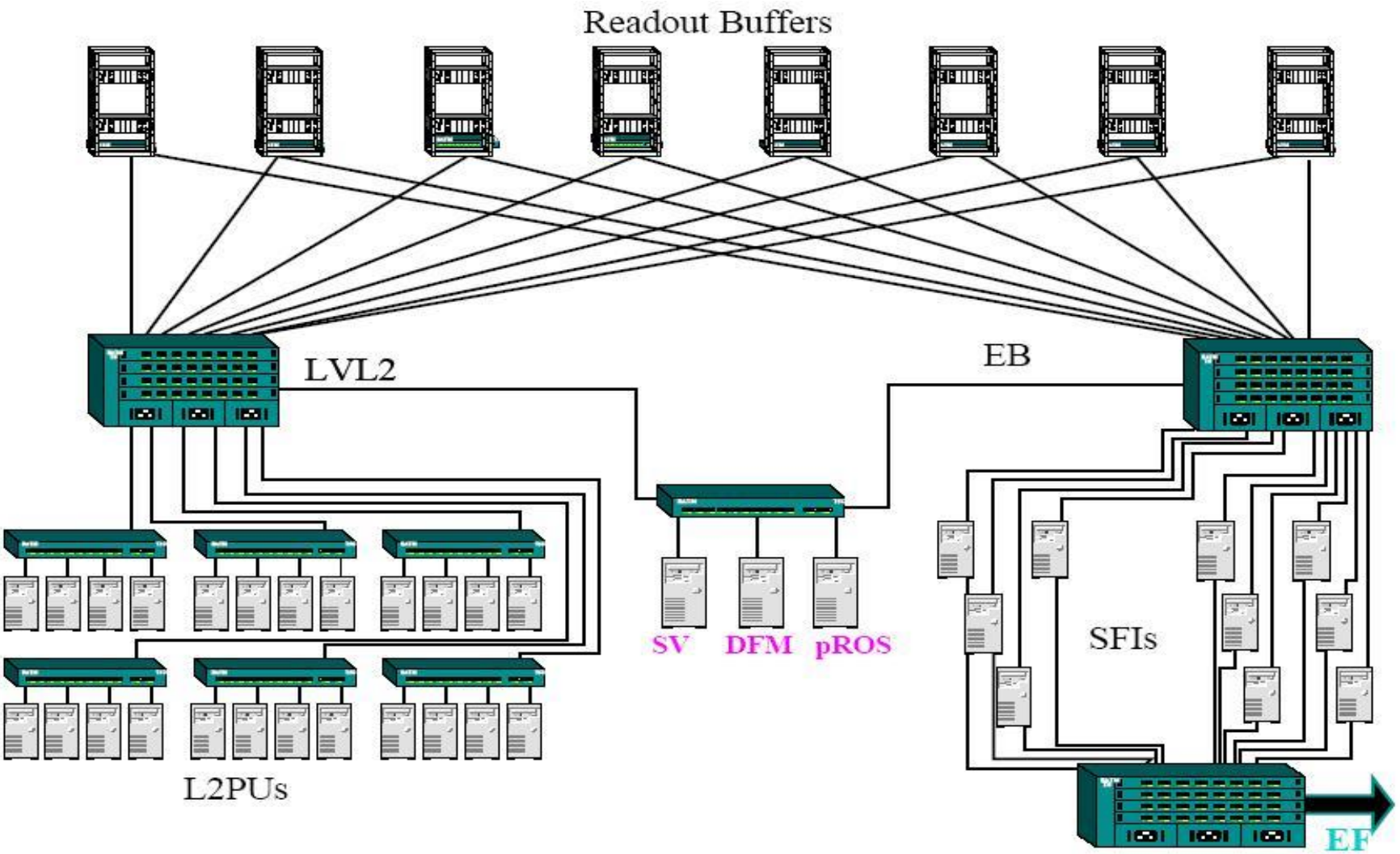


Collision rate	40 MHz	No. of In-Out units	**512**
Level-1 Maximum trigger rate	**100 kHz†**	**Readout network bandwidth**	**≈ 1 Terabit/s**
Average event size	**≈ 1 Mbyte**	**Event filter computing power**	**≈ 10⁶ SI95‡**
Event Flow Control	≈ 10^6 Mssg/s	Data production	≈ Tbyte/day
		No. of PC motherboards	≈ Thousands

† 50 kHz at startup (DAQ staging)

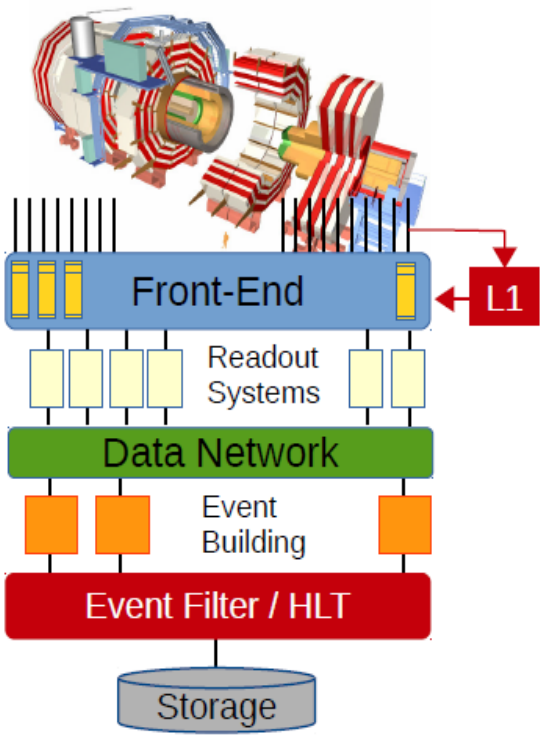
‡ 6×10⁵ at startup

ATLAS T/DAQ

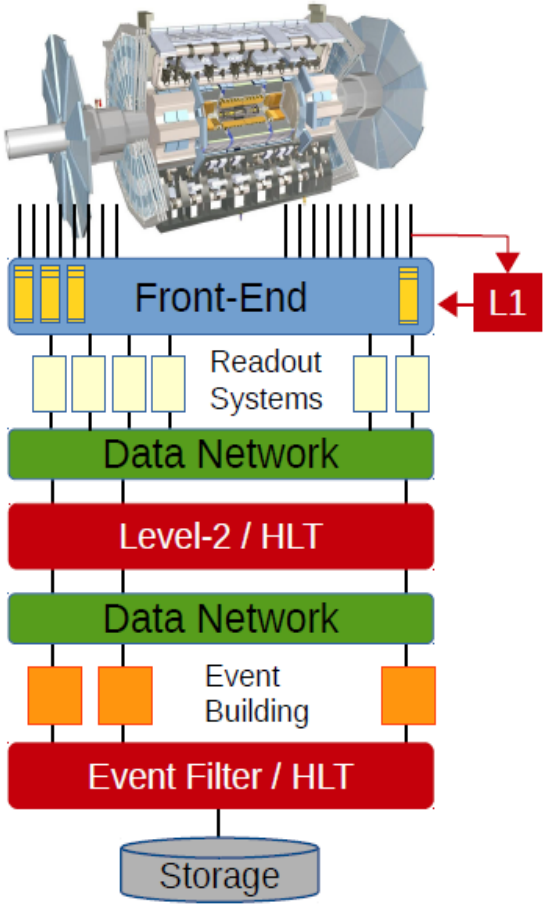


ATLAS/CMS trigger and DAQ

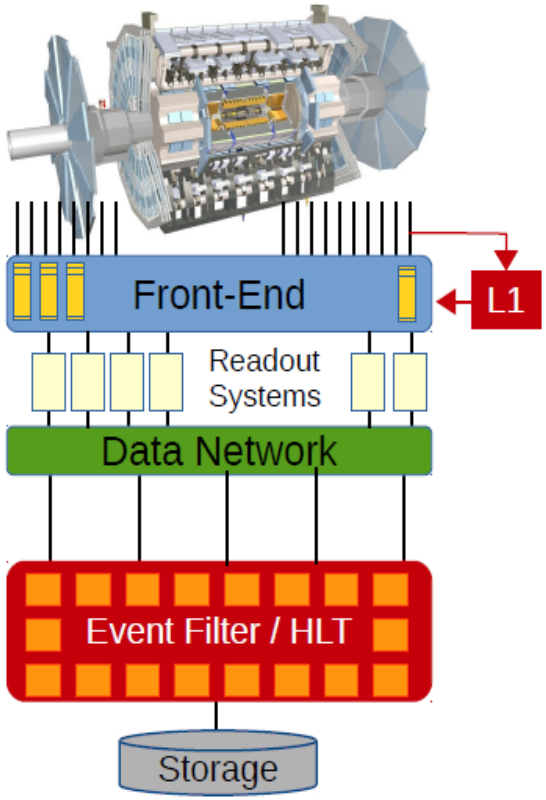
CMS Run-1&2
- 100 kHz EB
- Dedicated EB farm



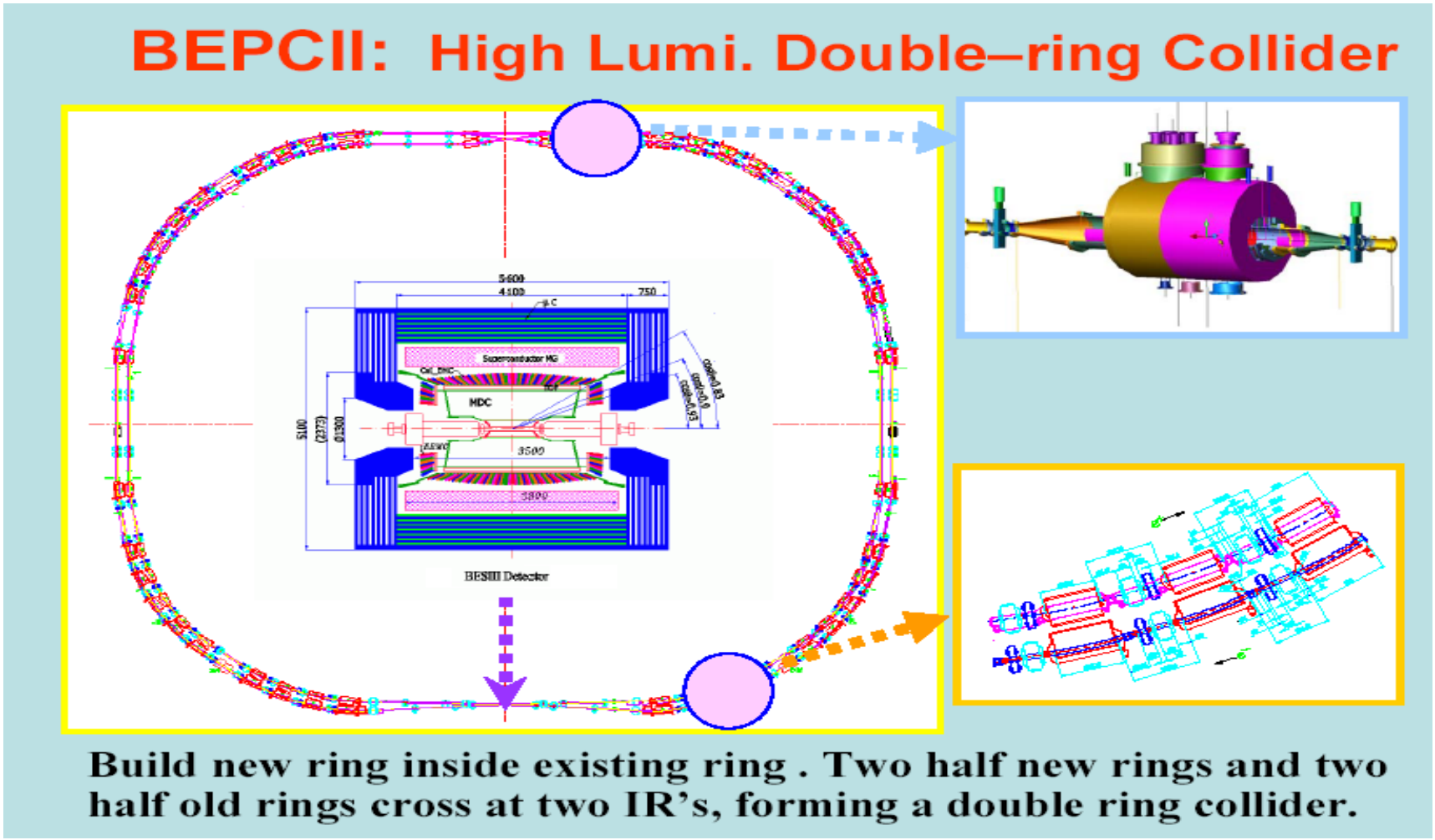
ATLAS Run-1
- 5 kHz EB after L2
- Dedicated EB farm



ATLAS Run-2
- incremental EB
- EB on HLT nodes

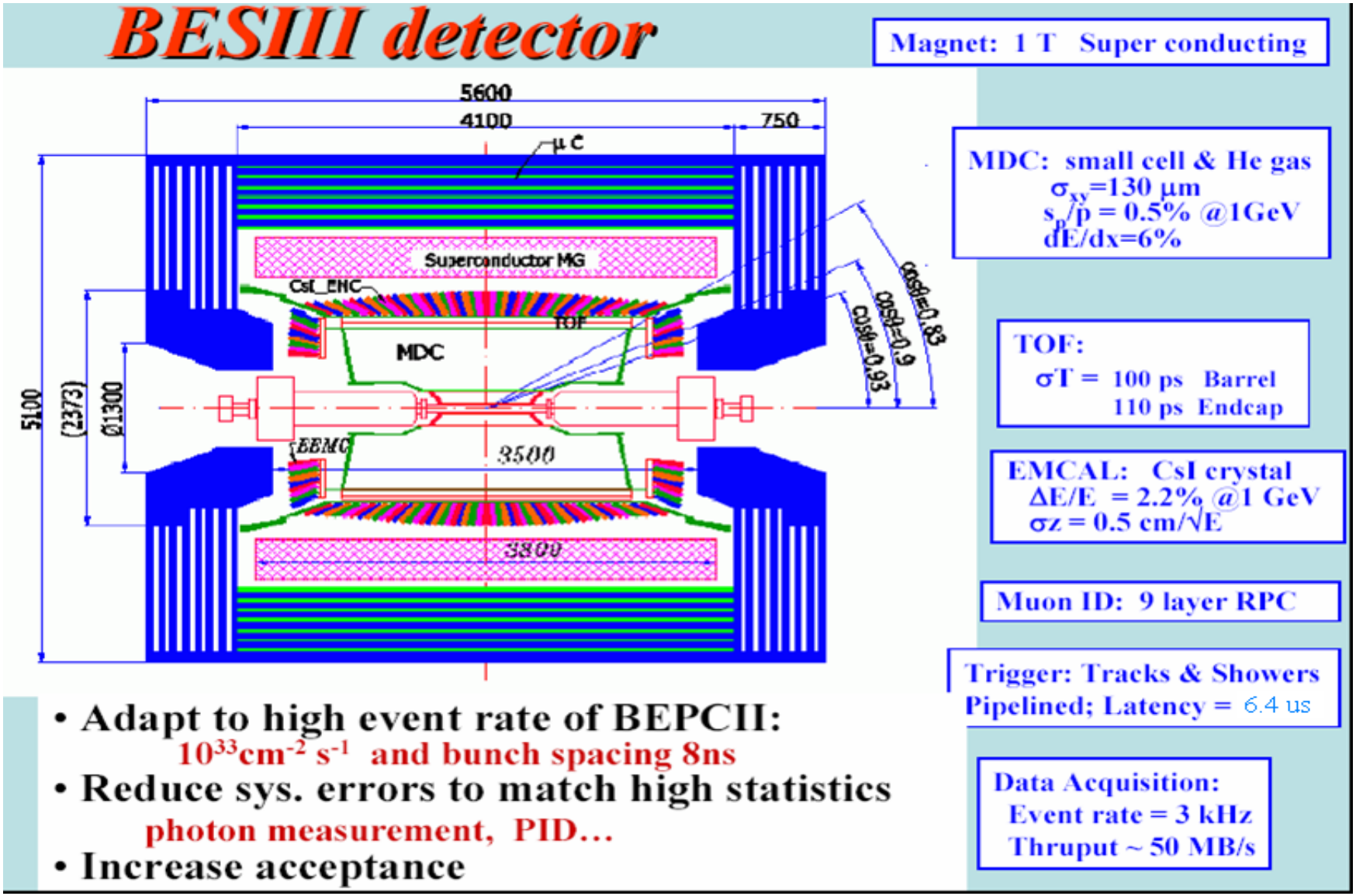


北京正负电子对撞机(BEPCII)



Luminosity: 3~10×$10^{32}cm^{-2}s^{-1}$ @ 3.77GeV

北京谱仪(BESIII)


BESIII detector
Magnet: 1 T Super conducting
5600
4100
750
μ C
Superconductor MG
CsI_ENC
TOF
MDC
EEMC
3500
3800
5100
(2373)
Ø1300
cosθ=0.83
cosθ=0.9
cosθ=0.93
MDC: small cell & He gas
σxy=130 μm
sp/p = 0.5% @1GeV
dE/dx=6%
TOF:
σT = 100 ps Barrel
110 ps Endcap
EMCAL: CsI crystal
ΔE/E = 2.2% @1 GeV
σz = 0.5 cm/√E
Muon ID: 9 layer RPC
Trigger: Tracks & Showers
Pipelined; Latency = 6.4 us
Data Acquisition:
Event rate = 3 kHz
Thruput ~ 50 MB/s
• Adapt to high event rate of BEPCII:
10^33 cm^-2 s^-1 and bunch spacing 8ns
• Reduce sys. errors to match high statistics
photon measurement, PID…
• Increase acceptance

BESIII DAQ设计指标

- 触发率： 4KHz
- 事例长度：12KB
- 数据带宽：48MB/s
- 死时间： < 5%

Sub-Detector	Channels
MDC (T+Q)	6796+6796
EMC	6240
TOF (T+Q)	448+448
MUC	9088
Total	~ 30K

1000 * BESII DAQ

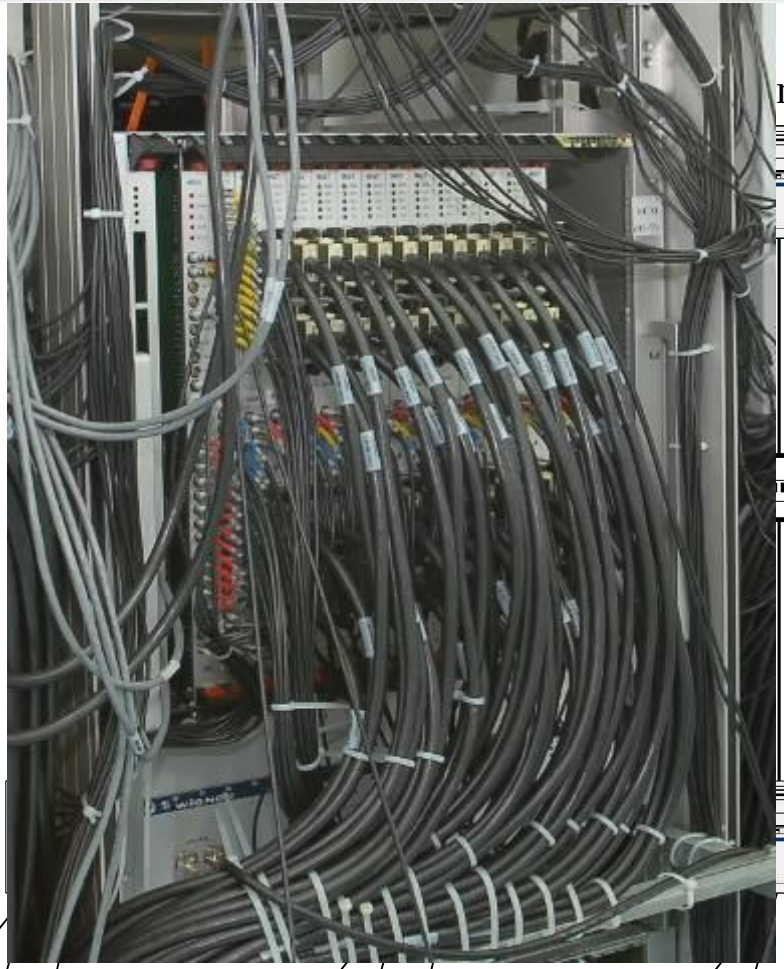
功能需求及系统组成

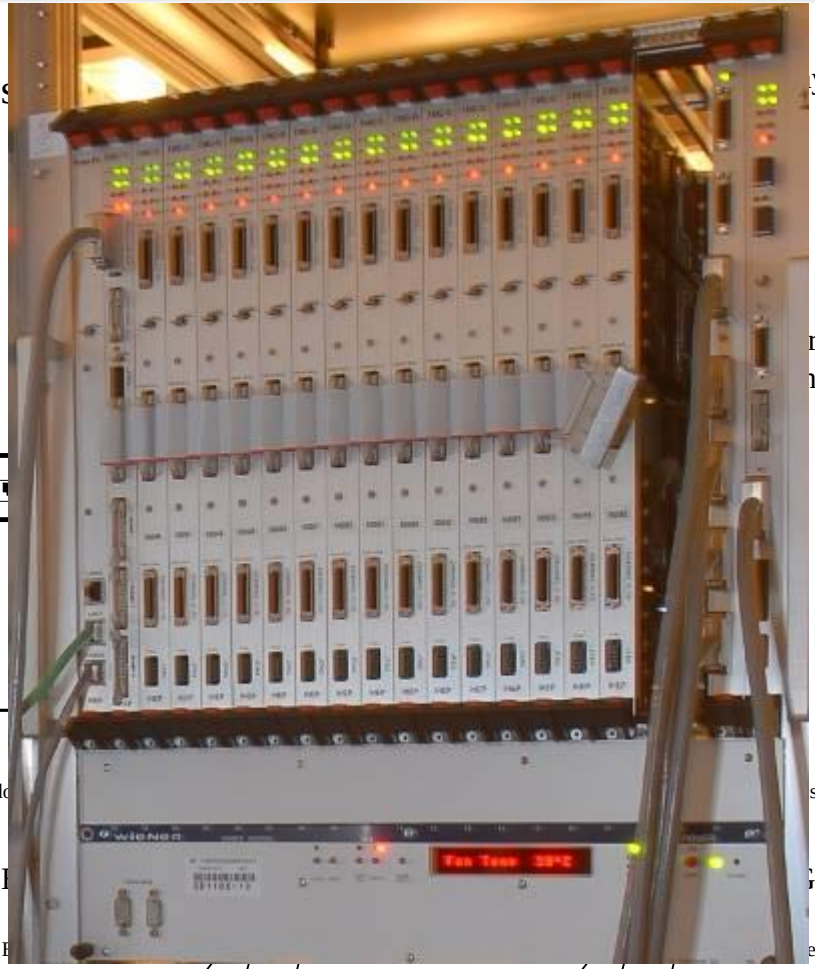
- 数据获取
  - 前端电子学读出：通过VME总线由PowerPC读出
  - 数据流：负责探测器数据通过网络传送
- 运行控制
- 电子学刻度
- 数据监测
  - 单事例显示
  - 直方图显示
- Event Filter
- 数据库

技术措施

- 分布式并行计算、层次结构
- 多线程技术、面向对象
- 基于网络交换技术
  - A multi-processor distributed environment
  - Parallel data streams working independently and concurrently
- 多级数据缓存
- 模块化设计，易升级和扩展
- 系统可靠、稳定

DAQ基本框架

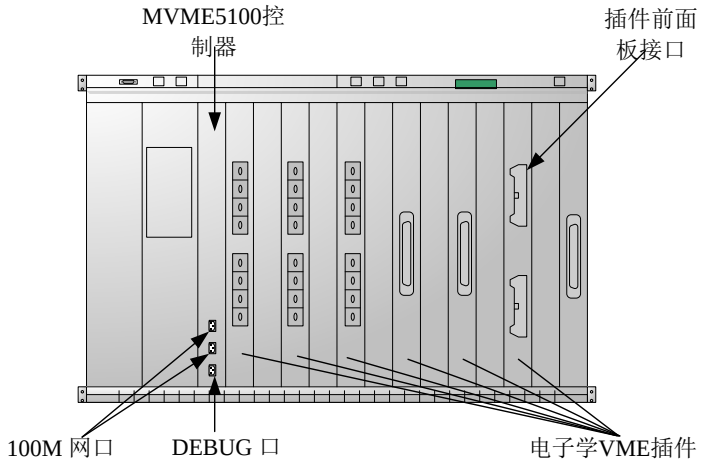




rm PCs
...
y
rnet
n
Readc
s (2)
Fast E
et
VME Crates (16)
VME Crates (2)
VME Crates (16)
VME Crates (4)
VME Crates (10)

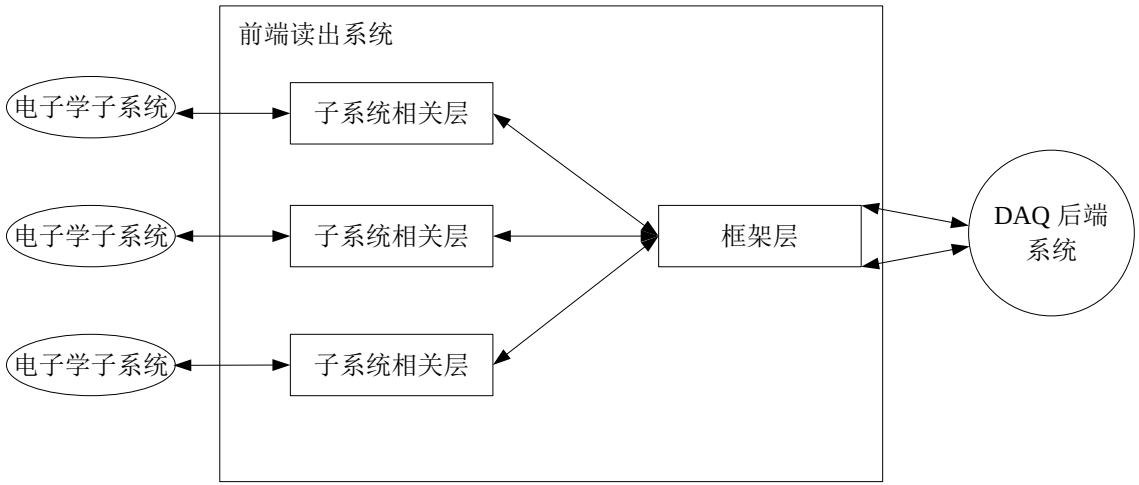
前端电子学读出任务及性能需求

- 系统任务
  - ✓ 响应DAQ后端控制命令、驱动电子学系统
  - ✓ 读取前端电子学插件中的事例数据
  - ✓ 检验数据正确性并完成第一级事例组装
  - ✓ 通过网络将数据上传
  - ✓ 完成前端电子学系统校准
- 系统硬件组成
  - ➢ 9U、6U VME机箱
  - ➢ MVME5100(主频450MHz)
    管理前端电子学插件

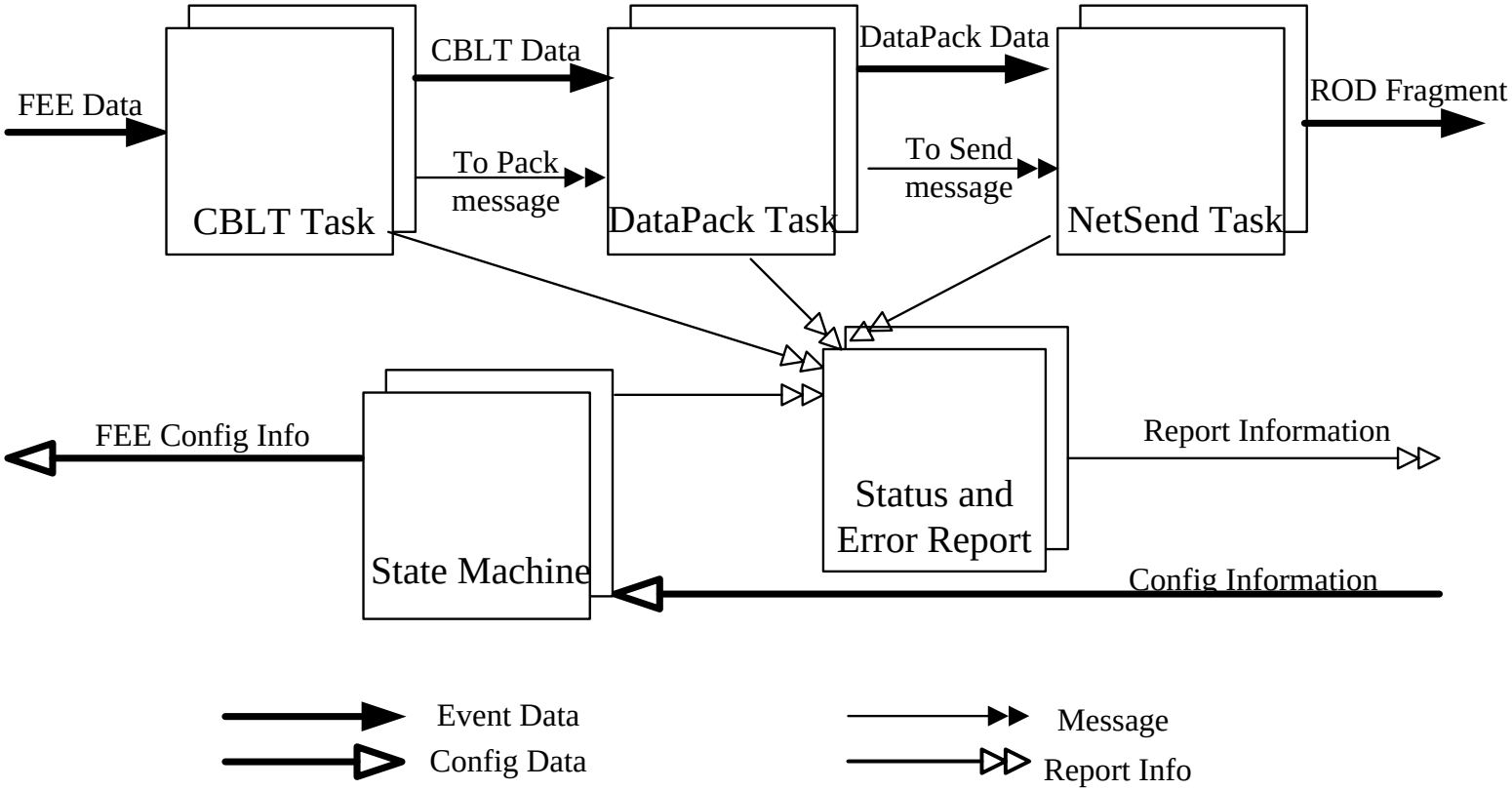


前端电子学读出基础软件框架

- ✓ 涉及5个读出子系统，避免重复设计
- ✓ 框架层：负责整个读出系统的通用功能，包括状态机跳转控制、网络连接、全局变量维护、中断服务等
- ✓ 读出子系统相关层：驱动每个电子学子系统的特有部分，具有硬件针对性



前端电子学读出核心任务与运行数据流

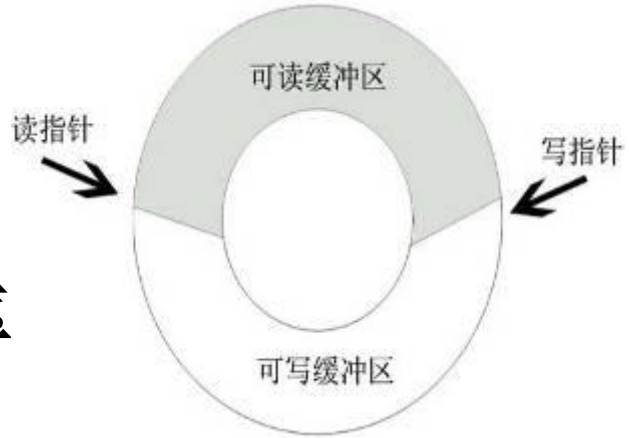


前端VME数据读出和处理（1）

- 单次VME总线读写
  - 平均每次读/写操作：1.7 / 1.1μs
  - 4KHz事例率，单机箱400字节数据长度的条件下，传输耗时：1.7μs×100×4000=0.68s
- DMA方式进行大量数据传输，释放CPU
  - 平均每传输4个字节：0.3μs
  - 4KHz事例率，单机箱400字节数据长度的条件下，传输耗时：0.3μs×100×4000=0.12s
- 网络传输
  - 达到100Mbps极限速度时占用50%的CPU

前端VME数据读出和处理（2）

- CBLT（Chained Block Transfer）方式
  - 启动DMA传输之前需指定传输的数据块长度
  - 每次触发产生的数据长度是随机的
  - 机箱中有多个电子学读出插件
  - 响应中断和启动传输都需要CPU干预
  - CBLT方式不需要事先指定传输的数据块长度，由BERR信号结束传输，插件之间通过令牌传递
- 环形缓冲
  - 可以高效、紧凑地利用空间
- 多事例读出，降低CPU占用率
  - 减少响应中断和启动传输次数



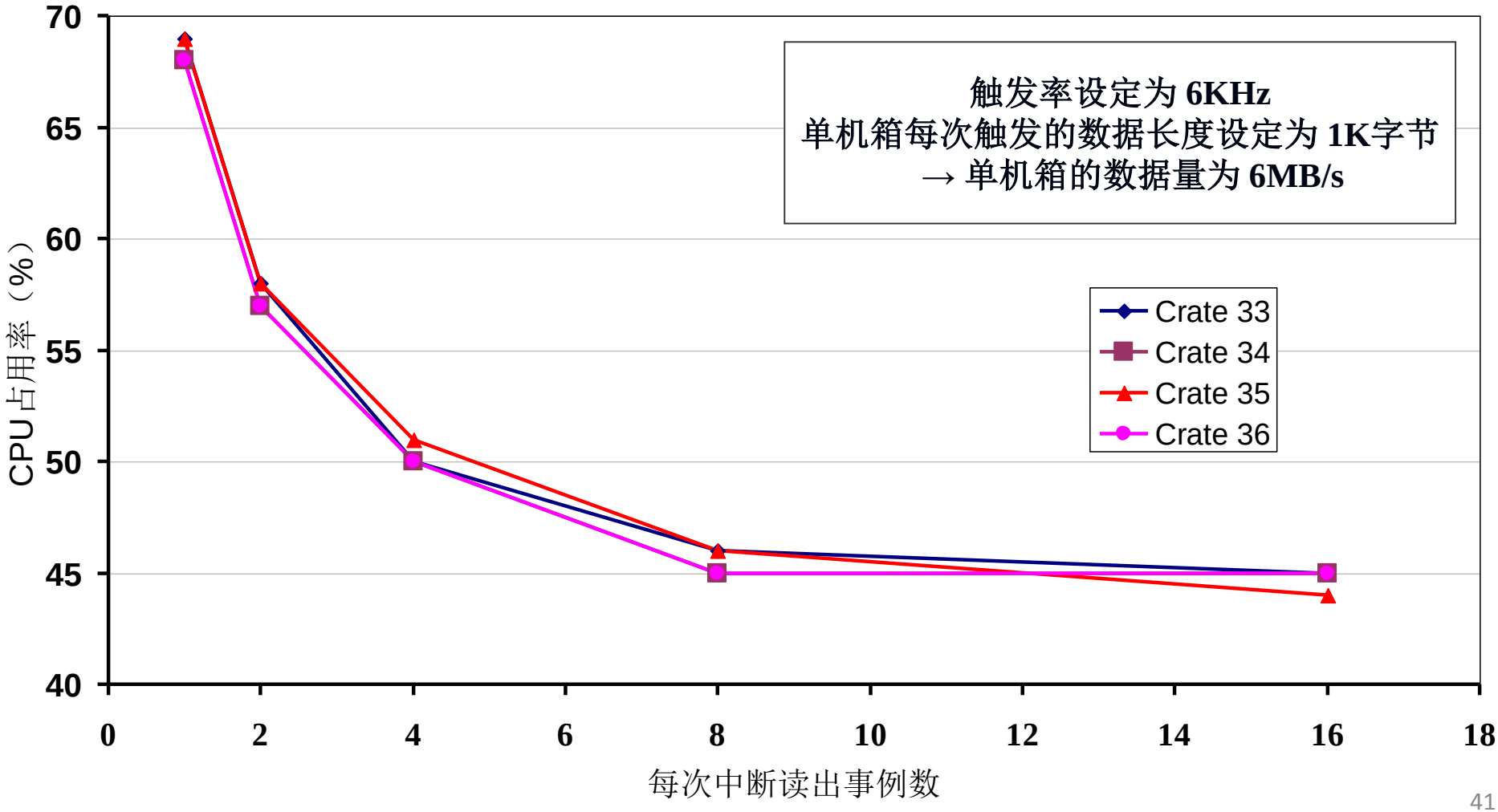
前端VME数据读出和处理（3）

- 并发任务
  - FWCbltTrans：
  - DataPack：
  - FWNetTrans：负责将组装后的数据通过网络发送
  - FWReportStatus：负责报告当前状态和出错信息

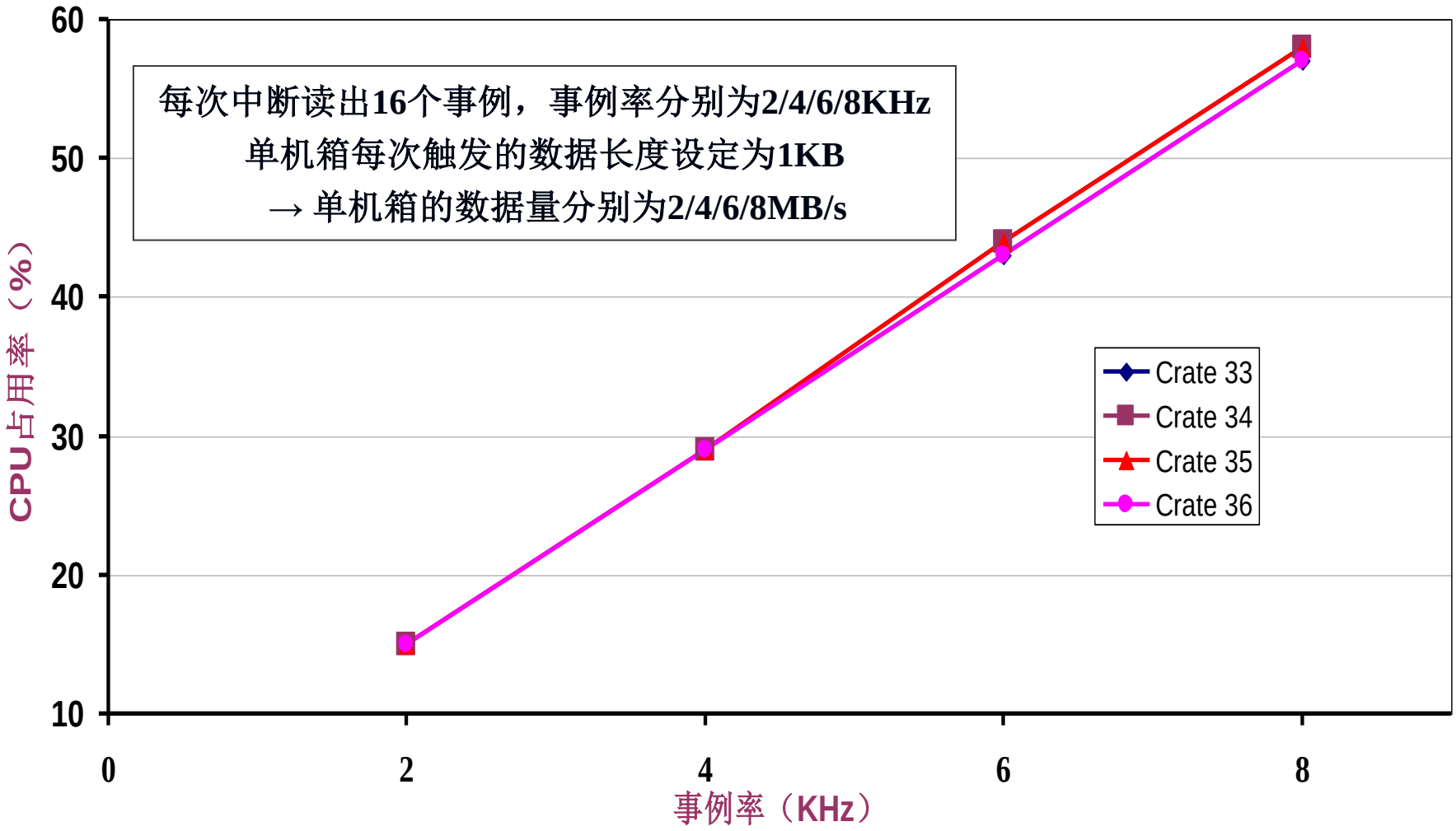
DMA传输任务和网络传输任务	传输1024字节平均时间（μs）	数据通过能力（Mbytes/S）
串行执行	185.9	5.4
并行执行	137.7	7.3

- 出错处理和状态报告
  - 触发号检查
  - 电子学读出数据的格式检查
  - 定时报告运行信息，如本run的总中断事例数
- 自动识别不同的电子学系统
  - 简化程序设计，减少运行维护的工作量

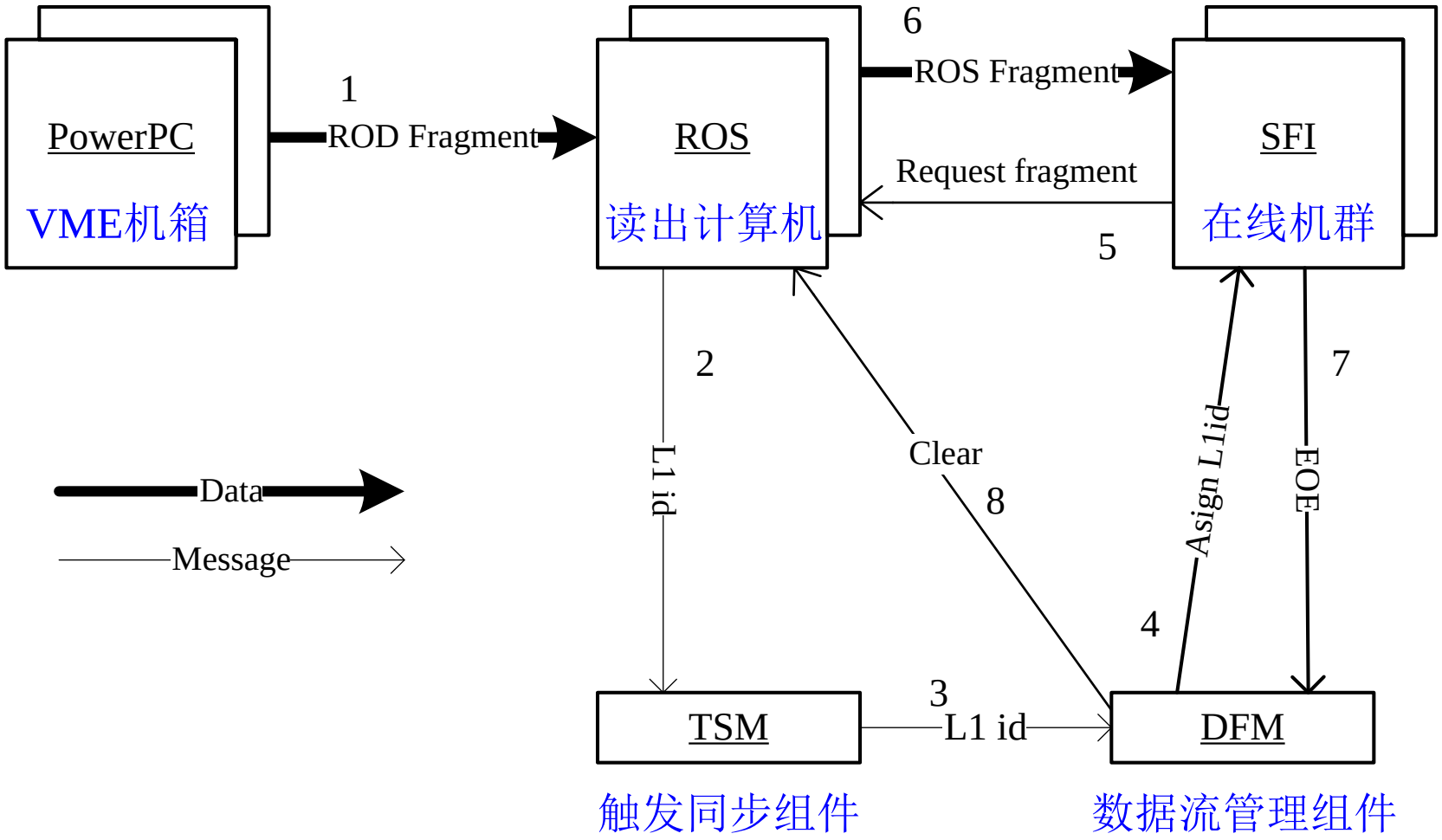
PowerPC CPU占用率 与每次中断读出事例数的关系



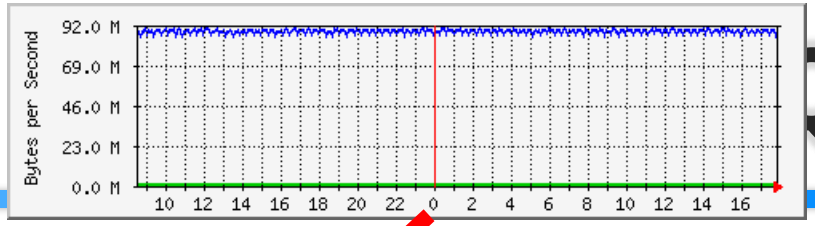
PowerPC CPU占用率与事例率的关系



控制和数据流程


PowerPC
VME机箱
1
ROD Fragment
ROS
读出计算机
6
ROS Fragment
SFI
在线机群
Request fragment
5
2
L1 id
Clear
8
7
EOE
Asign L1id
4
3
L1 id
TSM
DFM
触发同步组件
数据流管理组件
Data
Message

Q性能测试



BESIII DAQ Software Graphical User Interface - Expert Control

File Commands Access Control Tools Settings Help

Partition part_ppceall2

DAQ supervisor

DAQ SUPERVISOR STATE RUNNING

Shutdown Boot

Run control

RUN CONTROL STATE RUNNING

Unload Configure

Stop Start

Pause Continue

Checkpoint

Run Parameters

Run type	Physics
Run number	1021
Event number	1515272731
Event rate	6.512 kHz
Recording	Disable
Run Start Time	22/09/06 15:22:51
Run Stop Time	
Integrated active run time	

Monitor | Segment & Resource | CalibMon | Infrastructure

Run Control | Run Parameter | MRS | DAQ Supervisor | PMG | DataFlow

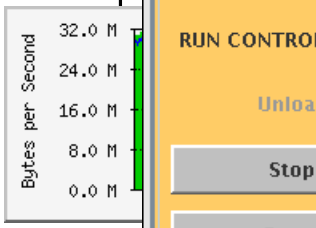
DFcontroller-1
- L2SV-1
- DFM-1
- SFI-1
- SFI-2
- SFI-3
- SFI-4
- SFI-5
- SFI-6

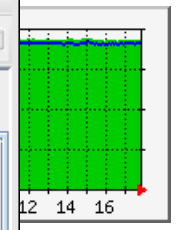
L2SV-1

Identity	L2SVResource
errors	0
LVL1_events	1515272769
LVL2_events	1515272769
AcceptedEvents	0
RejectedEvents	0
ForcedAccepts	1515272769
Throughput	6425.9443359375

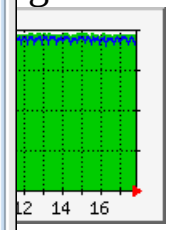
right button

08:49:23	INFORMATION	INTERNAL	Complete infrastructure is running. Connecting & starting IGUI.
08:49:21	INFORMATION	INTERNAL	Starting infrastructure please wait. IGUI will be started when complete infrastructure is running.





ger
onism
ager



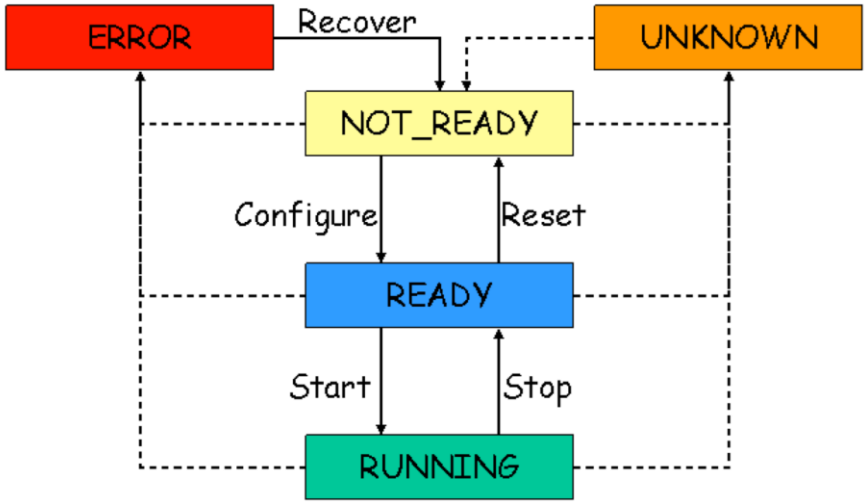
ROS (0)

TRG

PPCE (0)

Run Control

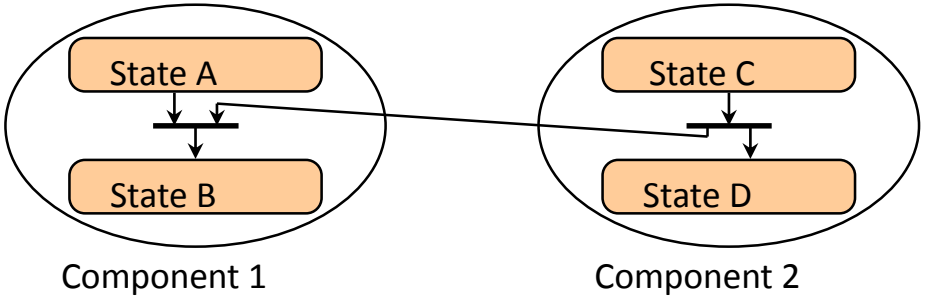
- The run controller provides the control of the trigger and data acquisition system. It is the application that interacts with the operator in charge of running the experiment.
- The operator is not always an expert on T/DAQ. The **user interface** on the Run Controller plays an important role.
- The complete system is modeled as a **finite state machine**. The commands that run controller offers to the operator are state transitions.



LHCb DAQ /Trigger Finite State Machine diagram (simplified)

Finite State Machine

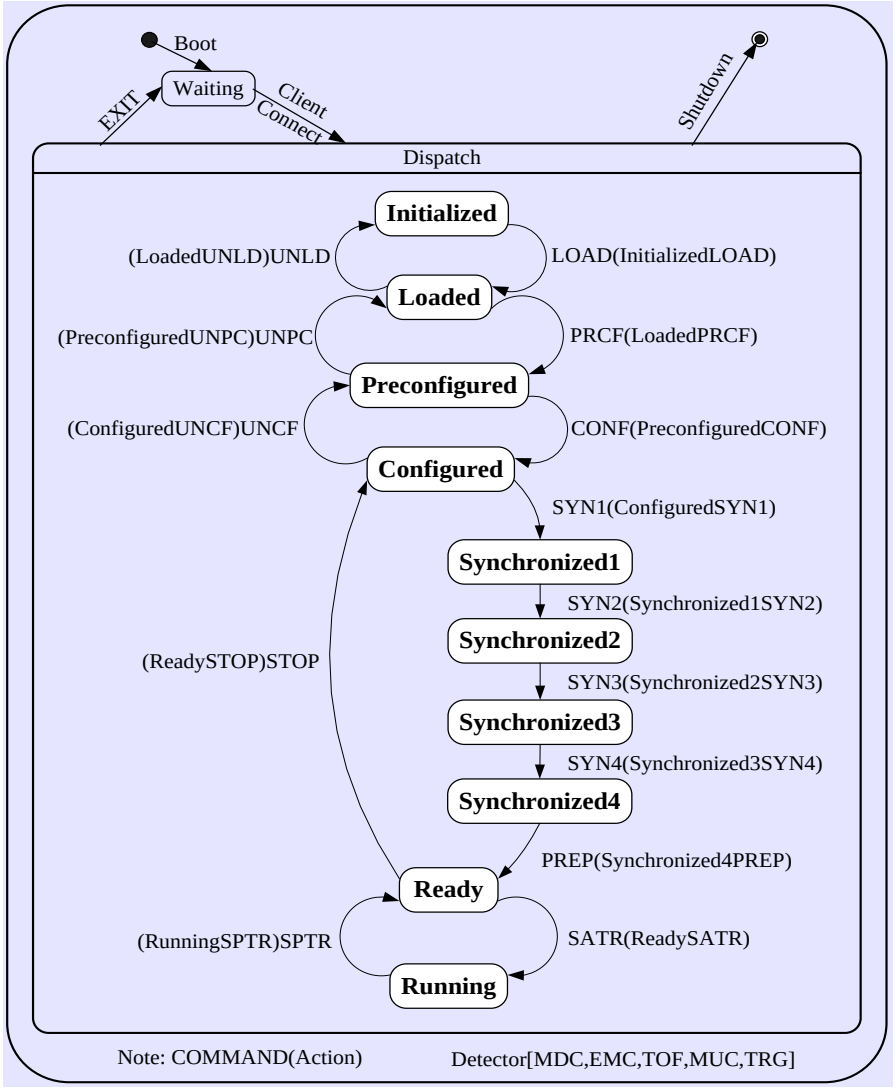
- Each component, sub-component of the system is modeled as a *Finite State Machine*. This abstraction facilitates the description of each component behavior without going into detail
- The control of the system is realized by inducing transitions on remote components due to a transition on a local component



Component 1 can only complete the transition to State B if Component 2 is in state D.

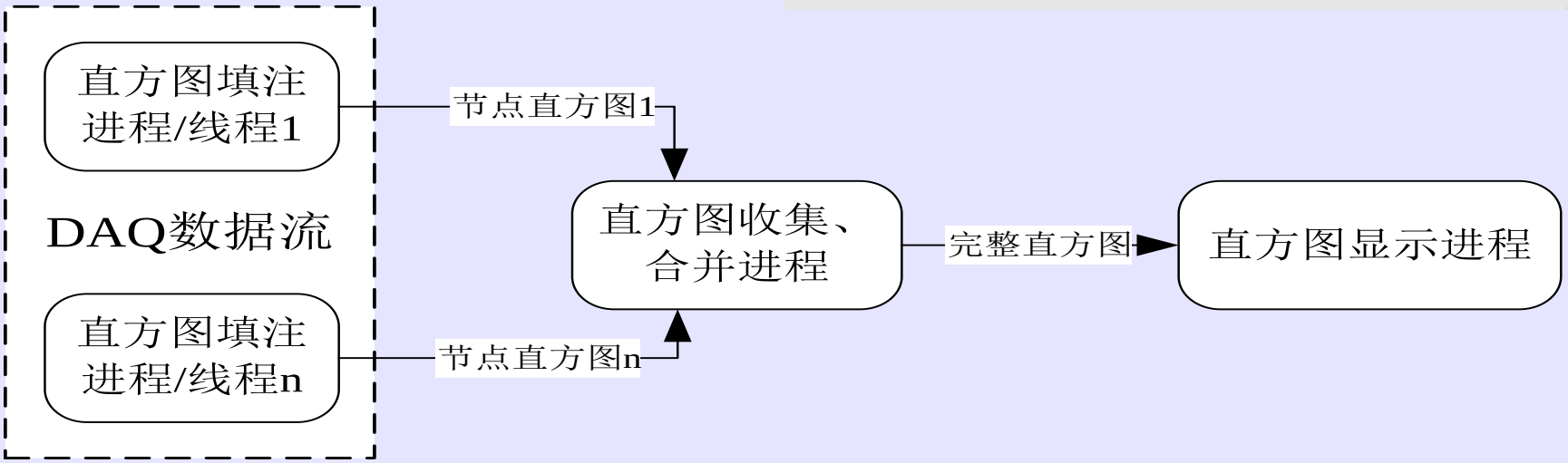
- Each transition may have actions associated. The action consist of code which needs to be executed in order to bring the component to its new state
- The functionality of the FSM and state propagation is available in special software packages such as SMI

BESIII DAQ状态机



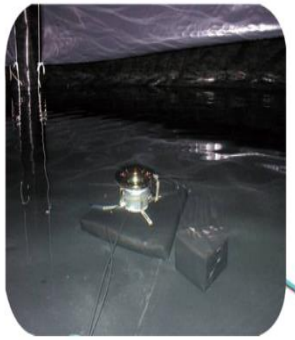
在线直方图

- 监测探测器运行状况，帮助系统调试
- 分布式集群处理，并行填图



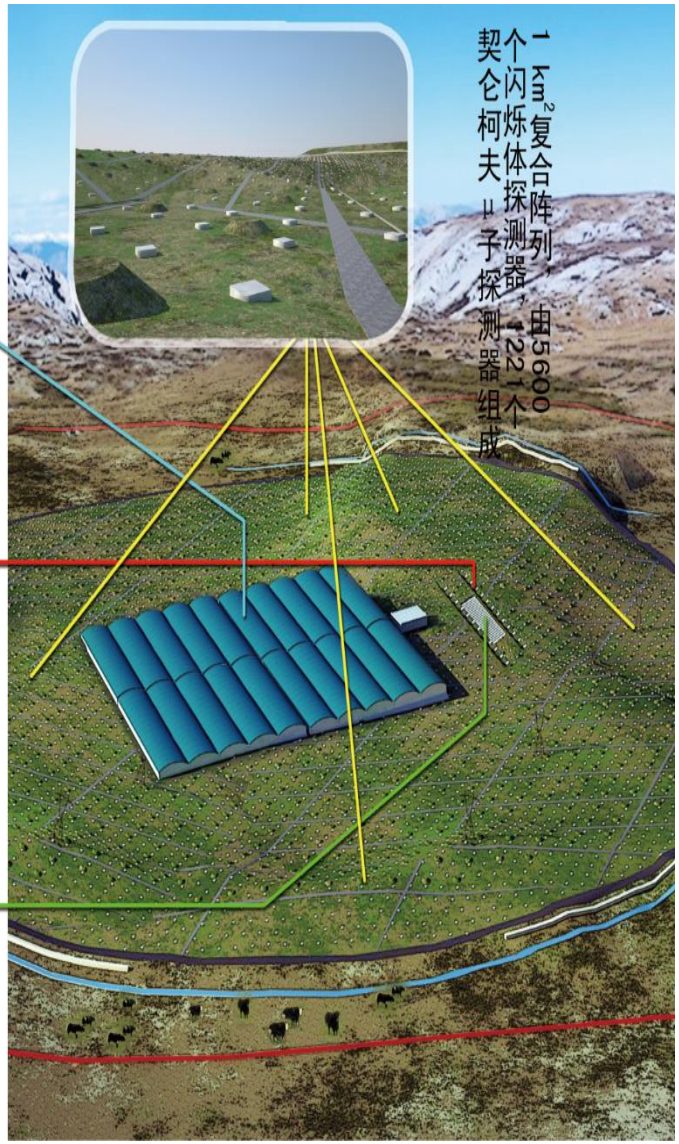
大型高海拔空气簇射观测站(LHAASO)DAQ

90，000 m^2水契仑柯夫探测器阵列



由24台广角成像望远镜组成的契伦柯夫光/荧光探测阵列

由452台闪烁体探测器组成的空气簇射芯探测器阵列



每个探测单元有独立的电子学系统，通过WR交换网络与DAQ建立TCP连接。DAQ最多须处理6800个数据通道。

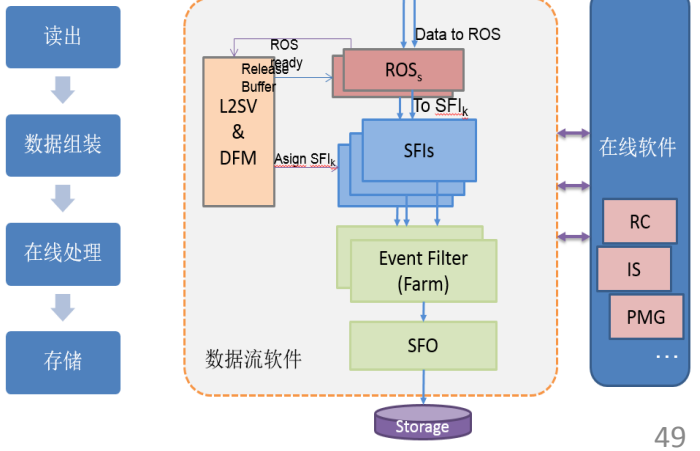
最大2GB/s的读出数据量。

须达到最多330MB/s的数据存储能力

软件触发最大应完成180M Hits/s的Hit处理

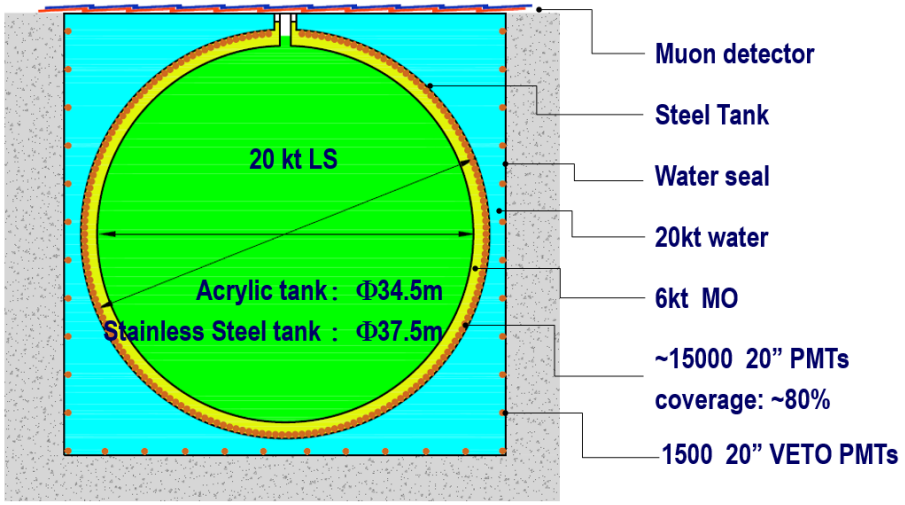
	KM2A	WCDA		WFCA	SCDA
		4水池方案 (每个水池)	1个大水池方案		
通道数 (个)	$e:5635$ $\mu:1221$	900	3600	24576	400
单通道 Hit 率 (Hz)	$e:1K$ $\mu:10K$	50K	50K	5K	<1K
触发前数据量 (MB/s)	450	350	2160	640	<5
全阵列 Hit 率 (Hz)	20.3M	45M	180M	123M	400
参与触发 Hit 率 (Hz)	5.6M	45M	180M	41M	400
触发后数据量 (MB/s)	≈10	≈23	≈330	30	很小

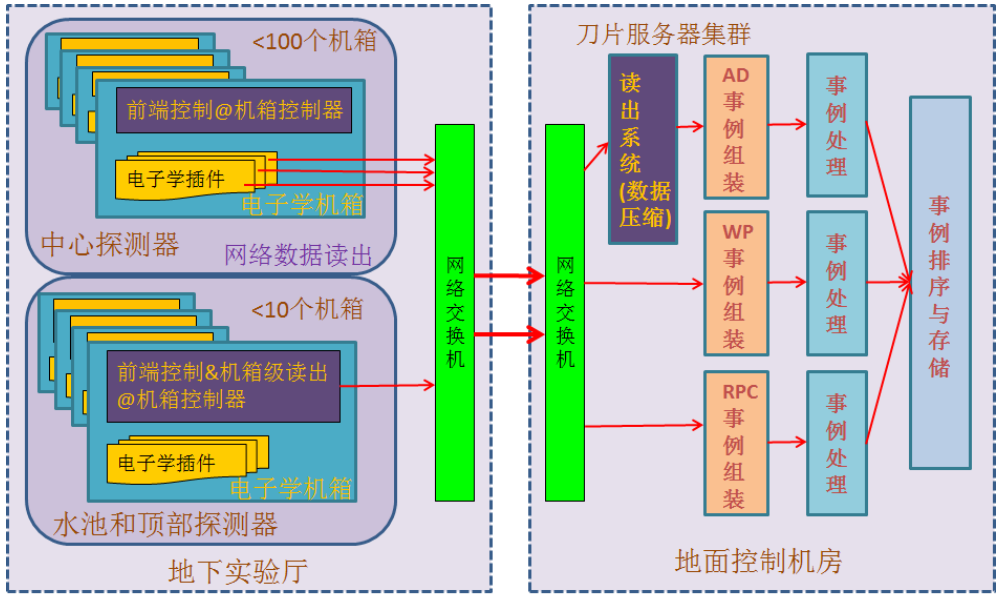
软件数据流



江门中微子实验（JUNO）DAQ

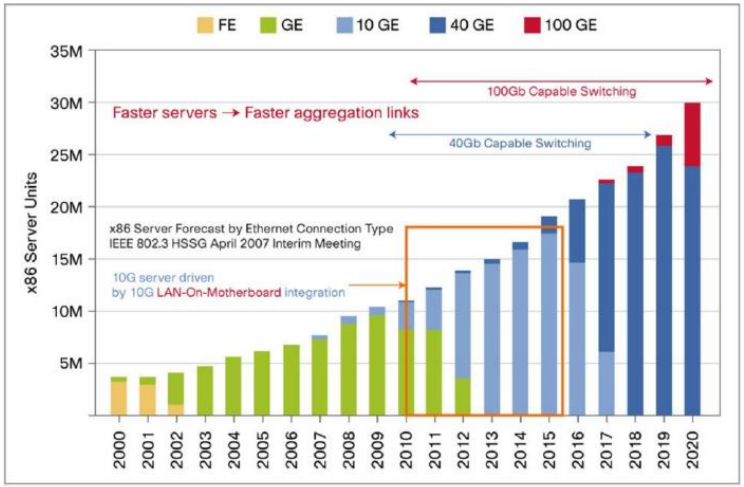
- 中心探测器
  - 1GHz采样，2Byte/采样
  - 1us窗口，2KByte/事例
  - 1kHz事例率，通道数16k
  - 2GB/s读出（1k通道着火）
  - 波型压缩后200MB/s
- 超新星爆发
  - 持续约10s，通道全着火
    - 1MHz@0.5s
    - 25KHz@9.5s
  - 峰值数据率16TB/s
  - 累计数据量<25TB
    - 1.5MB/通道





The upgraded LHC experiments (LS2 & LS3)

- ALICE
  - Continuous readout at TPC limit (~50 kHz)
  - Merge of online and offline computing farm
- LHCb
  - No HW trigger → 40(30) MHz to HLT
- ATLAS/CMS
  - Increase HW trigger output rate to ~ 1 MHz
  - Replacement of the majority of FE electronics
  - New inner trackers incl. HW-based track triggers
  - Details of TDAQ systems still very much under discussion



The Market Need for 40 Gigabit Ethernet, Cisco (2014)

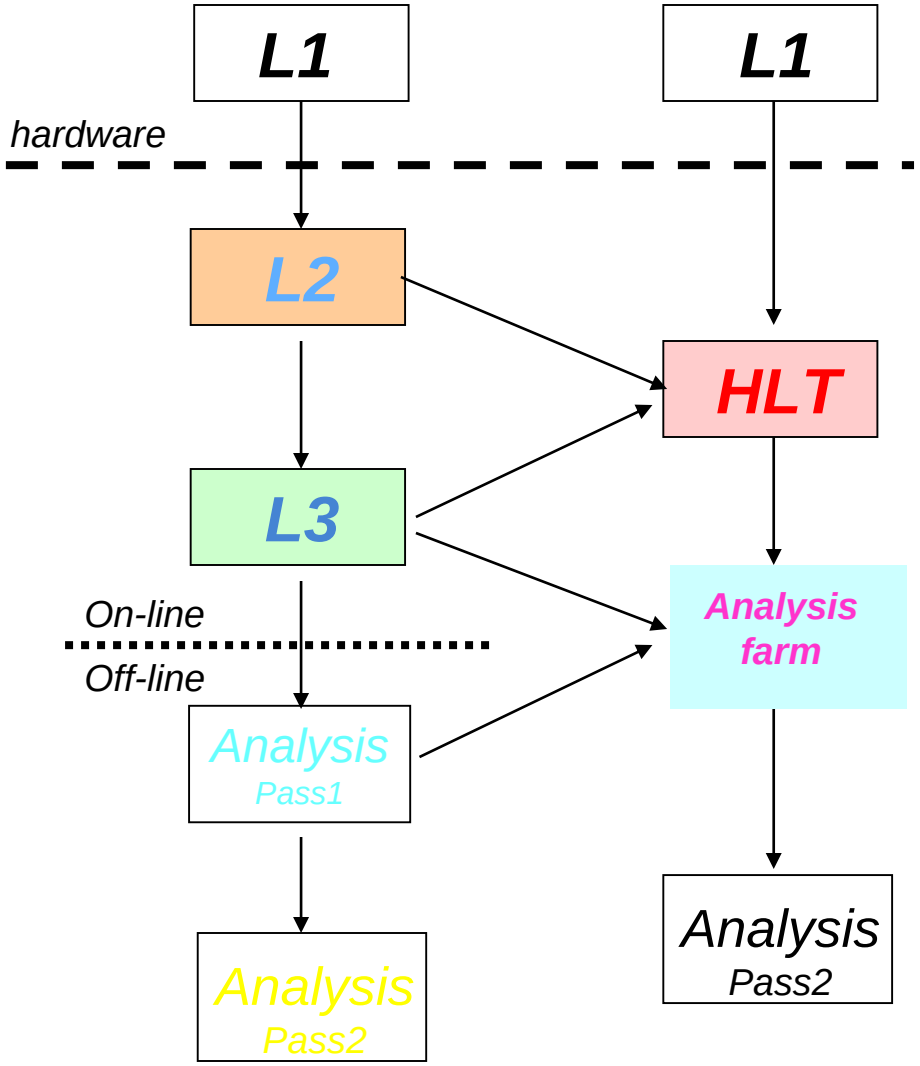
		# Trigger Levels HW	SW	Accept rate		Event size	Event building	Permanent Storage
ALICE (Pb-Pb)	Run-3	0	1	50 kHz		60 MB	†0.5 TB/s	†90 GB/s
LHCb	Run-3	0	1	30 MHz	20 kHz	0.1 MB	4 TB/s	2 GB/s
ATLAS	Run-4	1 (or 2)‡	1	0.4(1) MHz	10 kHz	5 MB	2(5) TB/s	50 GB/s
CMS	Run-4	1	1	0.75 MHz	7.5 kHz	5 MB	4 TB/s	40 GB/s

† Alice: event compression (factor~6) and only storing reconstructed objects
‡ Atlas: One or two-level HW trigger under discussion

触发与数据获取的发展趋势

* 大量采用通用商用硬件：计算机，交换机，……
* VME→xTCA
* 在线运行离线软件，更复杂的算法在线实现
* 软件触发，数据驱动
* 无线技术

* PANDA, LHAASO, JUNO，……



总结

高能物理实验数据获取技术的发展与计算机、网络和存储技术的发展密不可分

- 基于网络数据传输/事例组装
- 采用商业产品，如PC、以太网交换机和存储等
- 数据缓冲，减少死时间
- FPGA实现更加复杂的算法
- 触发和DAQ数据流逐渐合并
- 在线运行部分离线算法