

FairRoot

Status & Plans



M. Al-Turany

Status



- Works with C++11
- Ready for test beams and online analysis
- Re-engineering of some base classes is ongoing (Clean some historical stuff!)

Hot Topics

- Concurrency
- FairRoot & ALICE O²



Concurrency: Where we are now?



- Single threaded single process ROOT event loop
- User code is in Task hierarchy that runs sequentially
- Grid/batch jobs run embarrassingly parallel (one process/core)

What are the Problems



- C and C++ do not offer any support for concurrency!
- Embarrassingly parallel workload does not scale
 - Memory needed for each process → expensive
 - How this scheme should work for the Online clusters?

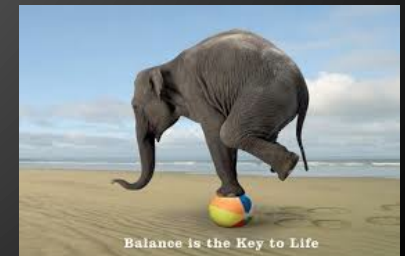
Multi-processing vs. Multi-threading



- Different processes are insulated from each other by the OS, an error in one process cannot bring down another process.
- Inter-process communication can be used across network
- Error in one thread can bring down all the threads in the process.
- Inter-thread communication is fast

Correct balance between reliability and performance

- Multi-process concept with message queues for data exchange
 - Each "Task" is a separate process, which can be also multithreaded, and the data exchange between the different tasks is done via messages.
 - Different topologies of tasks that can be adapted to the problem itself, and the hardware capabilities.



A cloud that let you connect different pieces together

- BSD sockets API
- Bindings for 30+ languages
- Lockless and Fast
- Automatic re-connection
- Multiplexed I/O



nanomsg is under development by the original author of ZeroMQ

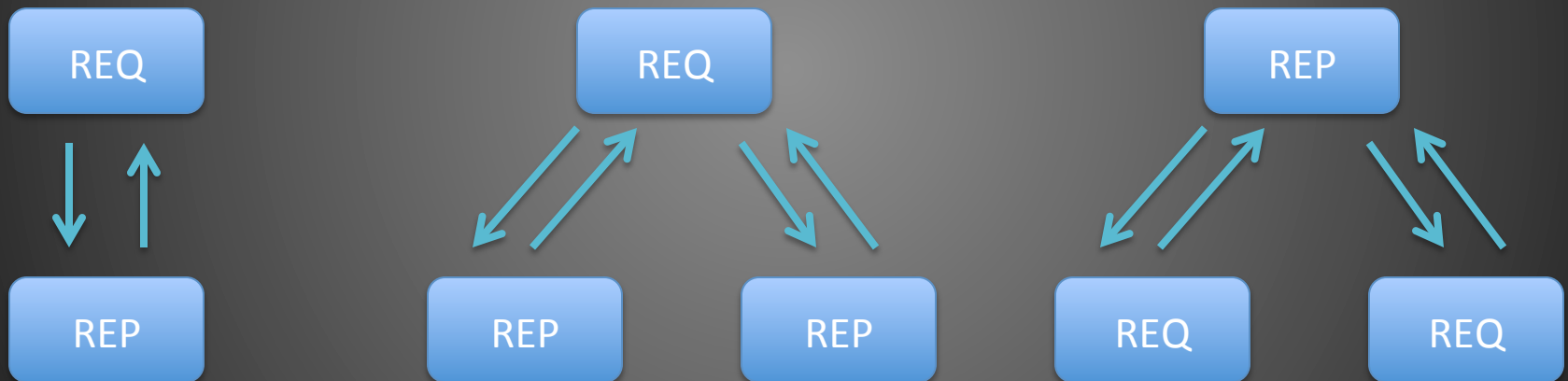
The logo for nanomsg, consisting of the word "nanomsg" in a bold, lowercase, sans-serif font, enclosed in a white rectangular box.

- Pluggable Transports:
 - ZeroMQ has no formal API for adding new transports (Infiniband, WebSockets, etc). nanomsg defines such API, which simplifies implementation of new transports.
- Zero-Copy:
 - Better zero-copy support with RDMA and shared memory, which will improve transfer rates for larger data for inter-process communication.
- Simpler interface:
 - simplifies some zeromq concepts and API, for example, it no longer needs Context class.
- Numerous other improvements, described here:
<http://nanomsg.org/documentation-zeromq.html>
- FairRoot is independent from the transport library
 - Modular/Pluggable/Switchable transport libraries.

Request-reply:

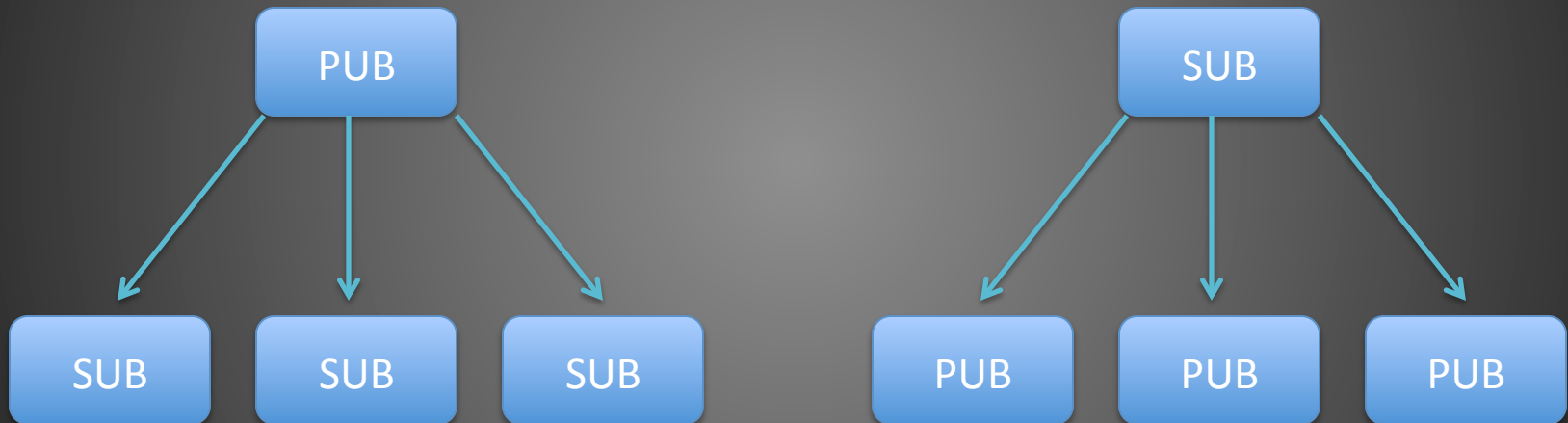
connects a set of clients to a set of services.

(remote procedure call and task distribution pattern)



Publish-subscribe

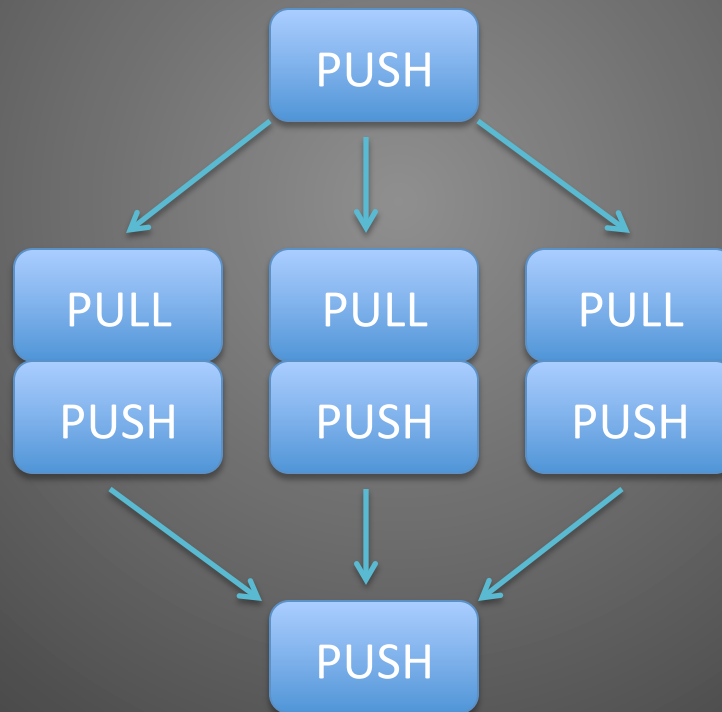
which connects a set of publishers to a set of subscribers.
(data distribution pattern)



Pipeline

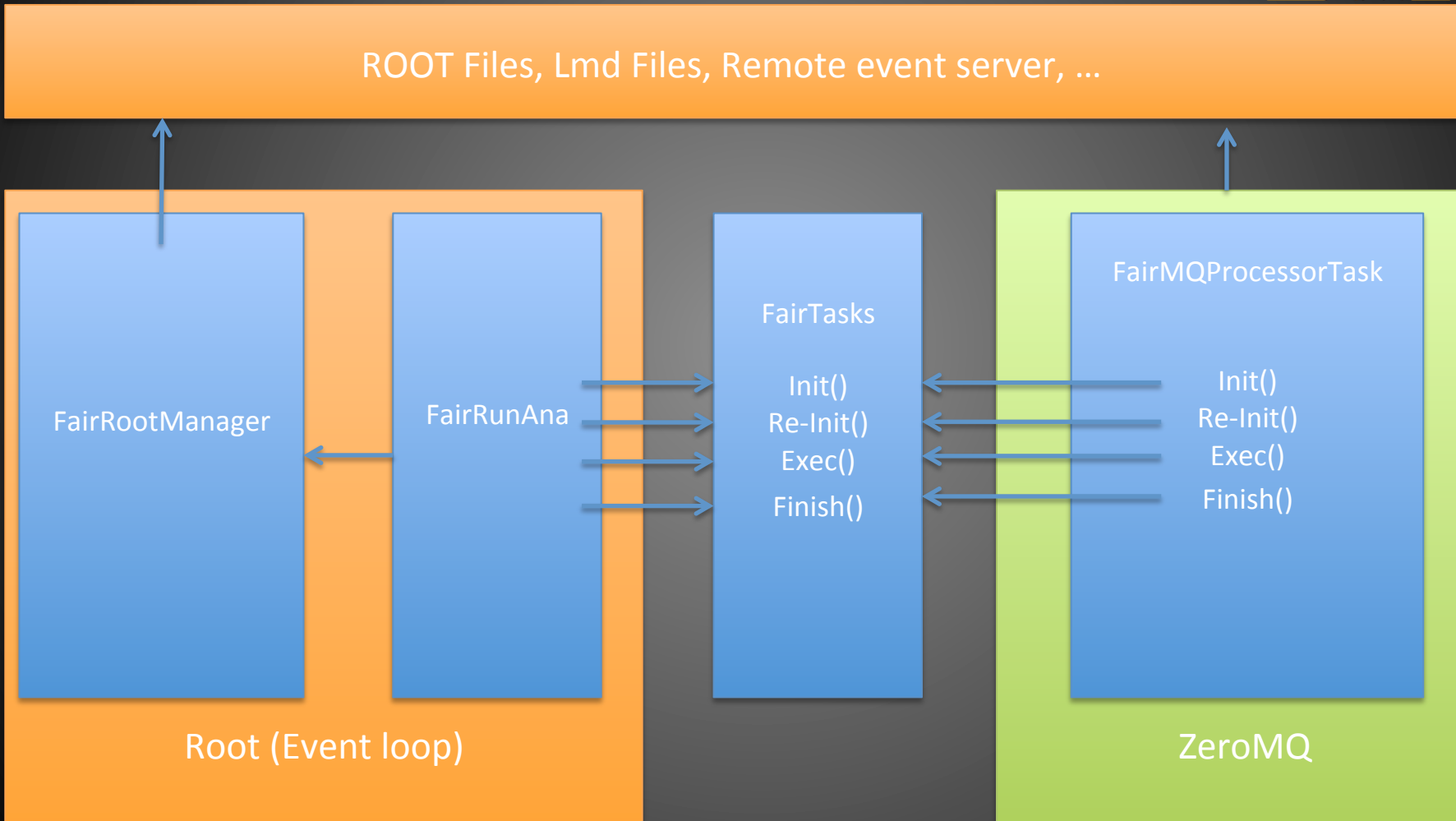
connects nodes in a fan-out / fan-in pattern that can have multiple steps, and loops.

(Parallel task distribution and collection pattern)





Integrating the existing software:



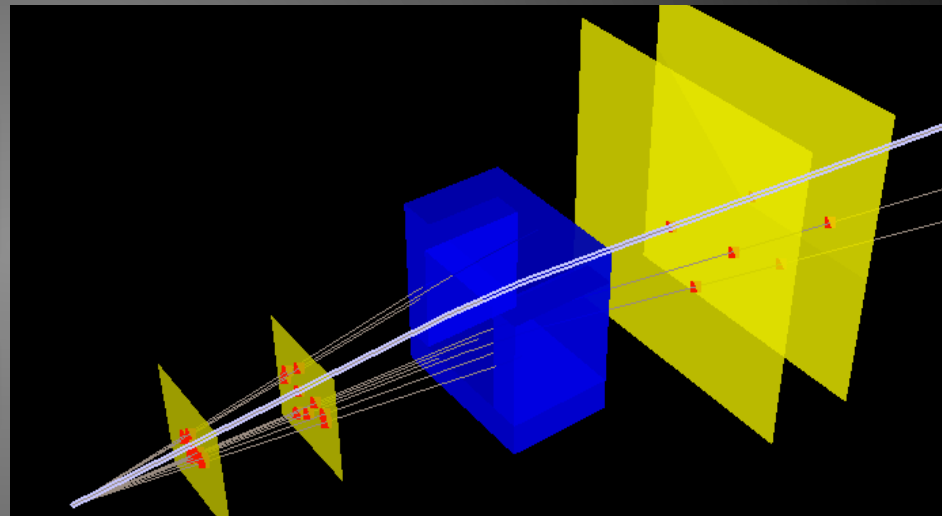
FairRoot: Example 3

4 -Tracking stations with
a dipole field

Simulation:
10k event: 300 Protons/ev

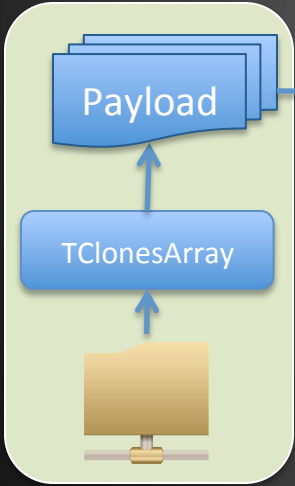
Digitization

Reconstruction:
Hit/Cluster Finder

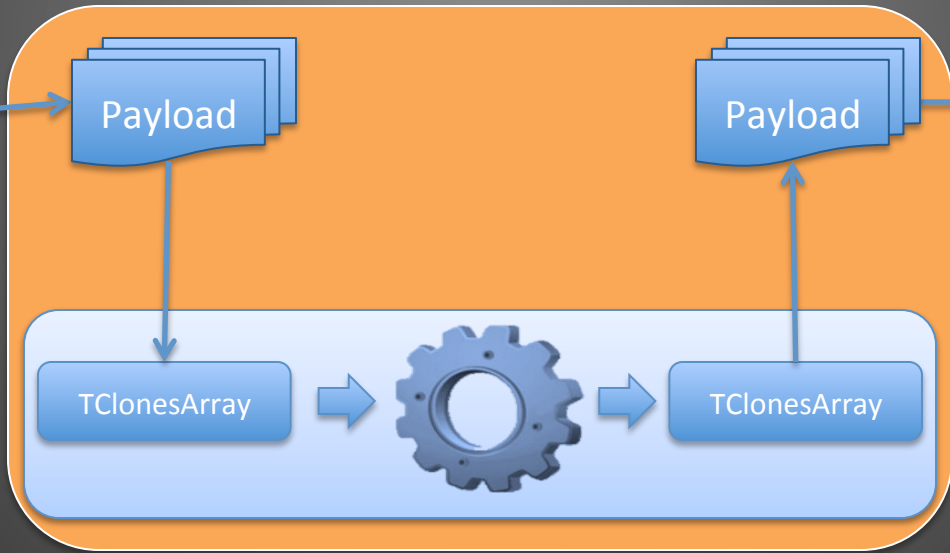




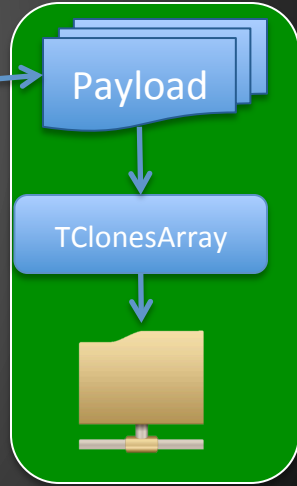
Vs.



Sampler



Processor



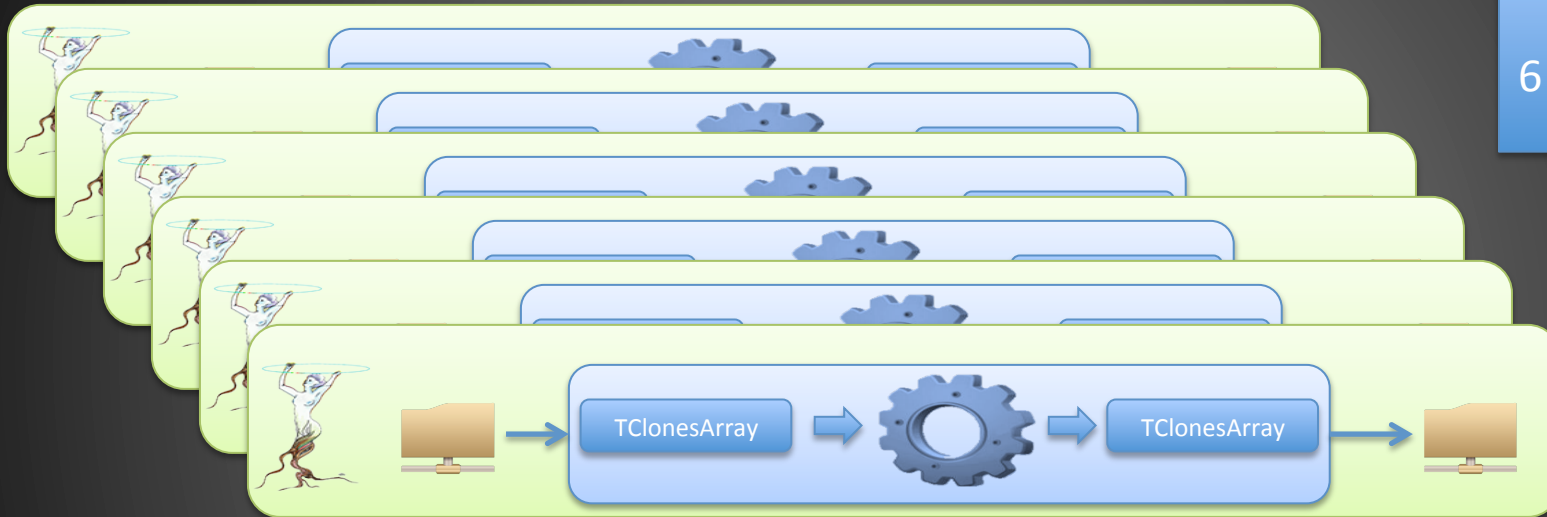
Sink

From digits to hits with ROOT:



RUN	CPU Time (s) (Wall time)	Memory (Mbyte)
10k Events, 300 Protons/event	100	263

2 x 2.4 Xeon Quad core Intel Xeon 16 GB Memory



171 s
6 * 263 MB

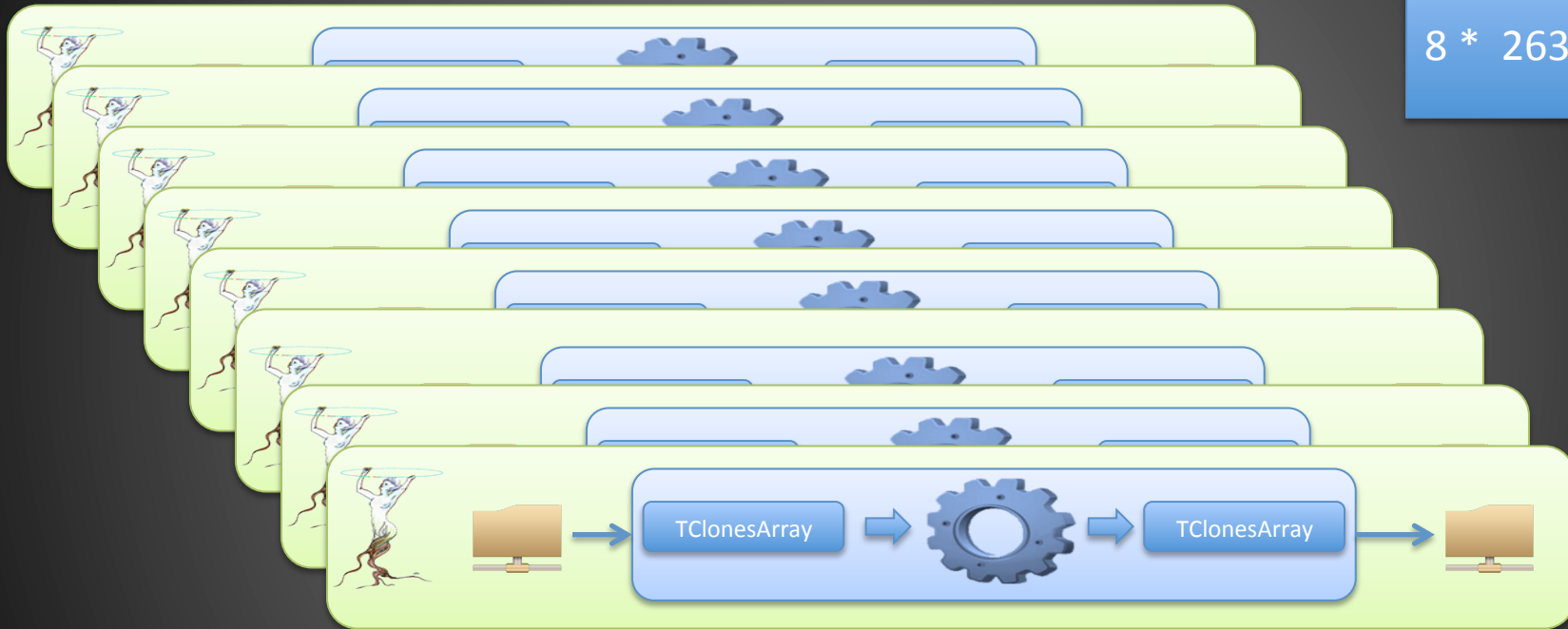
Throughput ~ 3500 ev/s

Wall time: 171 s
Total Event: 60k events

2 x 2.4 Xeon Quad core Intel Xeon 16 GB Memory

300 s

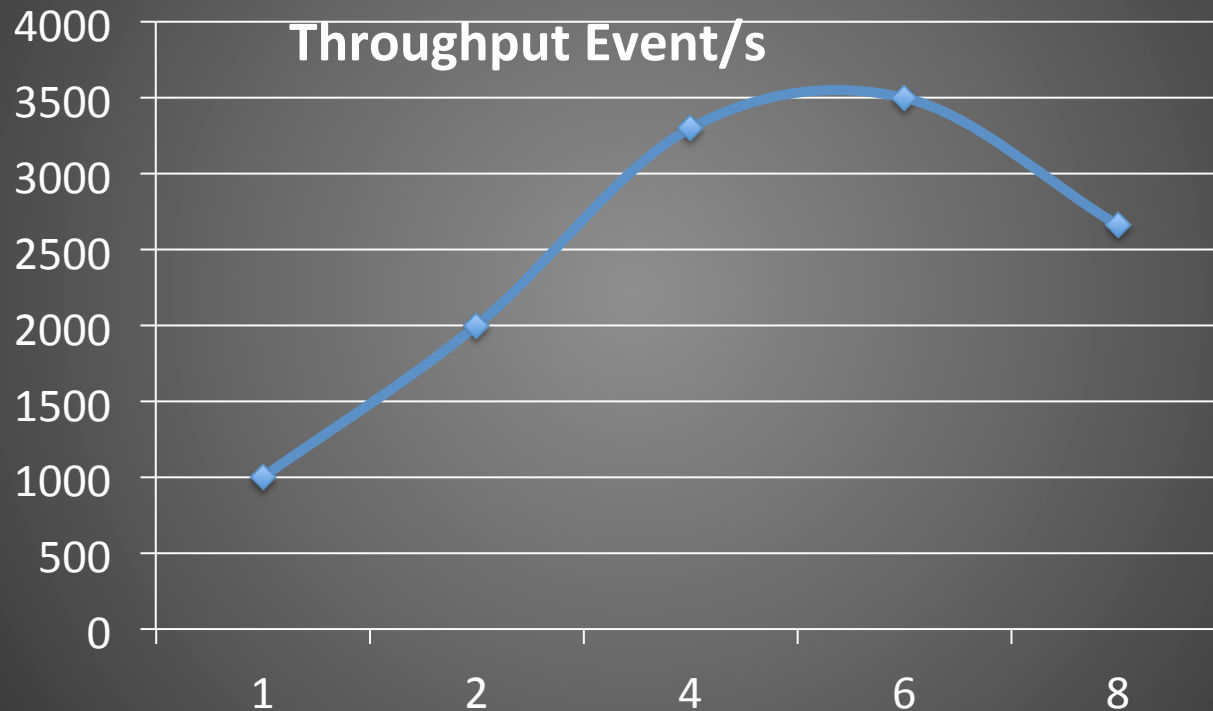
8 * 263 MB



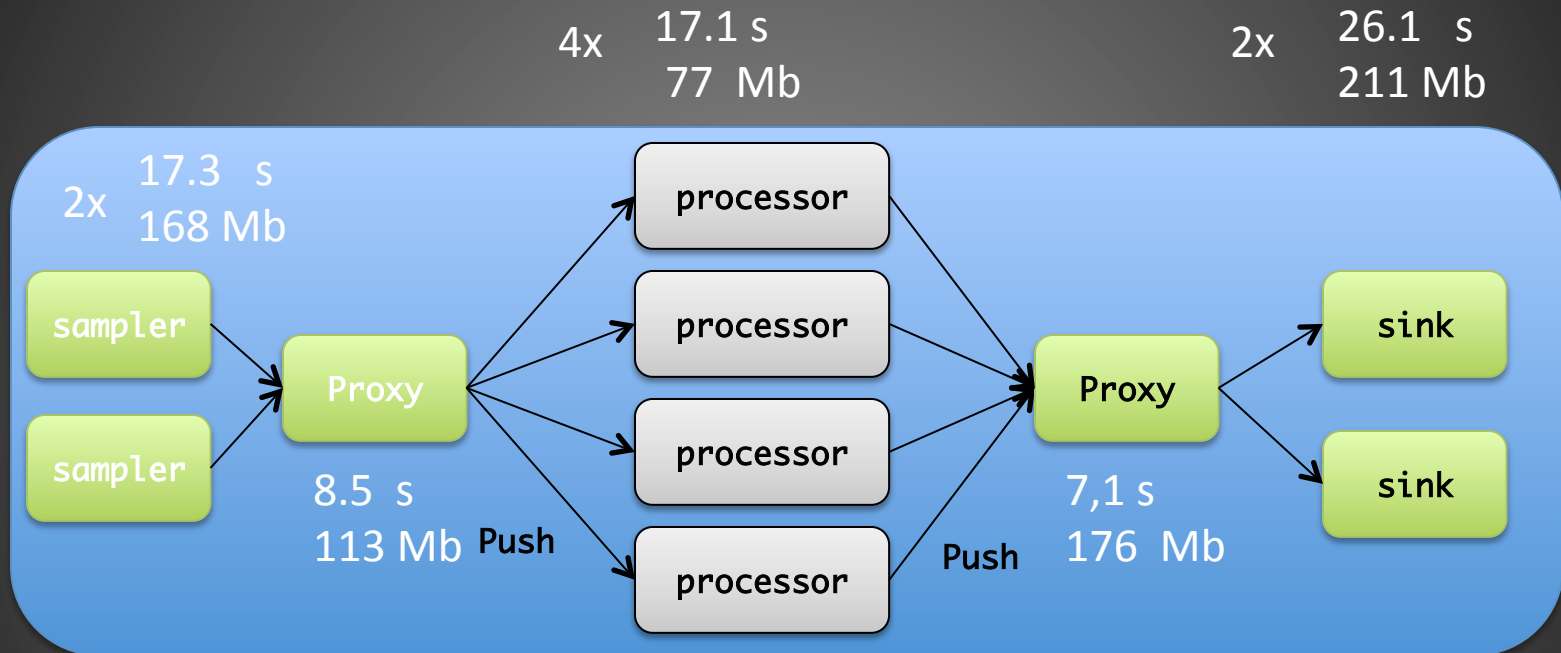
Throughput ~ 2660 ev/s

Wall time: 300 s
Total Event: 80k events

2 x 2.4 Xeon Quad core Intel Xeon 16 GB Memory



2 x 2.4 Xeon Quad core Intel Xeon 16 GB Memory



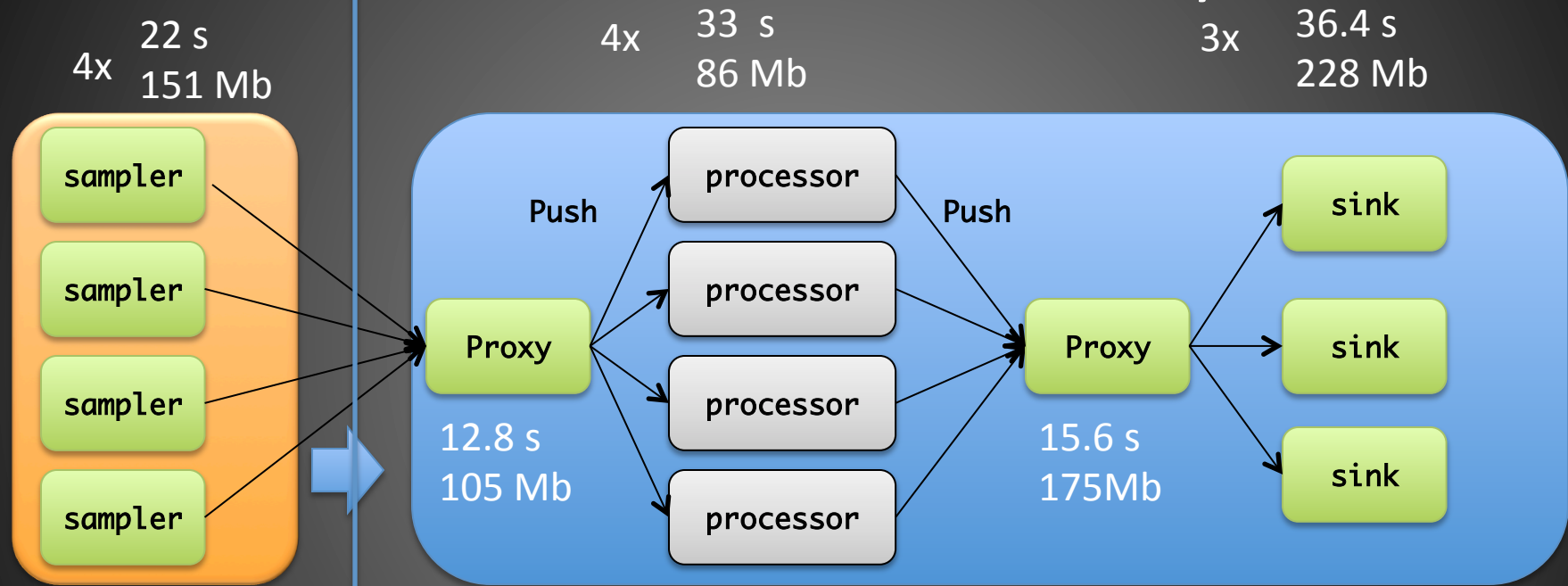
Throughput ~ 7400 ev/s
Total Memory 1355 Mb

Wall time: 26.1 s
Total Event: 20k events





2 x 2.4 Xeon Quad core Intel Xeon 16 GB Memory

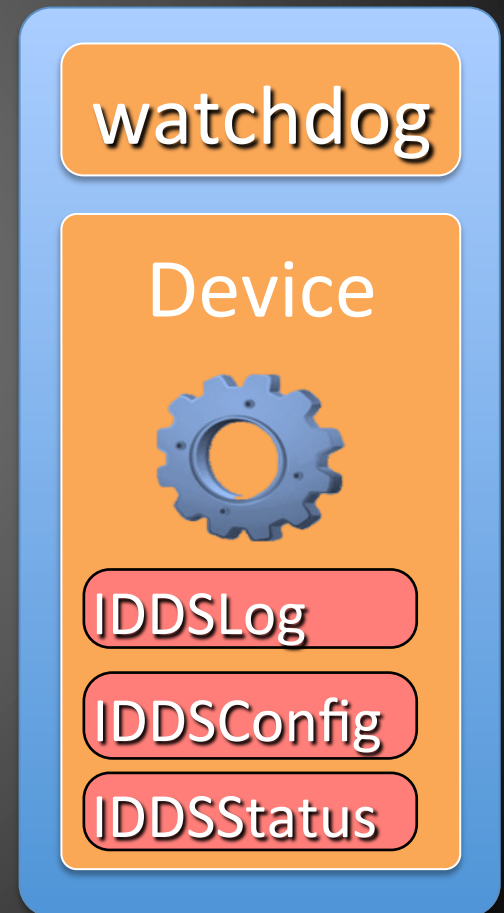


Throughput ~ 10990 ev/s

Wall time: 36.4 s
Total Event: 40k events

New building blocks for FairRoot:

- Each entity has its own watchdog process
- The entity is statically inherited and implement 3 interfaces:
 - IDDSConfig
 - interface to configuration files
 - used by the watchdog and by the user process
 - IDDSStatus
 - High and low level status info used by the watchdog
 - IDDSLog





FairRoot Group at the GSI:



- Support all reasonable environments available
- Run on Laptop and Cluster
- **NOT** participate in any middle ware development, we do not have the man power or the knowhow to do that.

FairRoot & ALICE O²

- Long years of close collaboration with the Alice offline group
- The computing requirement for Alice after upgrade are very similar to FAIR experiments
- The new design suggested for FairRoot fulfill also the requirement of Alice O²



FairRoot & ALICE O² (cont.)

- A common development with ALICE would be beneficial for all
- The HLT of ALICE could be seen as a prototype for FAIR online clusters
- We will benefit from the experience gained by supporting a running experiment!

