



Track 1 Session: #3  
(Data acquisition and electronics)  
April 14, 2015 18:00 –  
Village Center

# The Application of DAQ-Middleware to the J-PARC E16 Experiment

E Hamada<sup>1</sup>, M Ikeno<sup>1</sup>, D Kawama<sup>2</sup>, Y Morino<sup>1</sup>, W Nakai<sup>3, 2</sup>, Y Obara<sup>3</sup>,  
K Ozawa<sup>1</sup>, H Sendai<sup>1</sup>, T N Takahashi<sup>4</sup>, M M Tanaka<sup>1</sup>, S Yokkaichi<sup>2</sup>

<sup>1</sup>High Energy Accelerator Research Organization (KEK)

<sup>2</sup>RIKEN Nishina Center

<sup>3</sup>The University of Tokyo

<sup>4</sup>Research Center for Nuclear Physics, Osaka University

1. **DAQ-Middleware**  
overview and features
2. **DAQ System at E16 Experiment**  
requirements and architecture
3. **DAQ System Performance**  
method and result
4. **Summary and Future Plan**

# 1. DAQ-Middleware

# What DAQ-Middleware Is

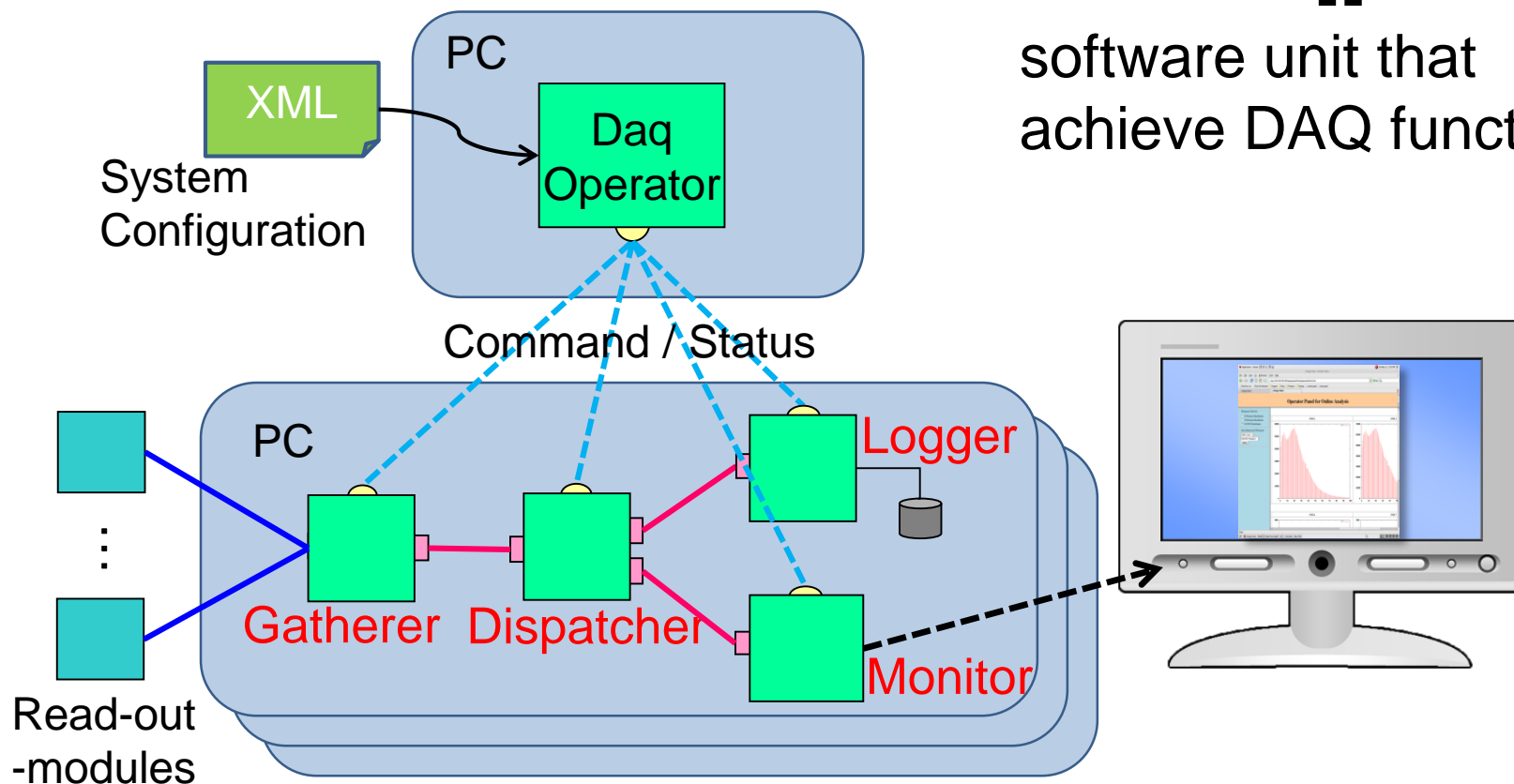
- A framework for network based DAQ software
  - Easy to use, configure and develop
- Target
  - Medium-scale experiments
  - Test benches (sensors, electronics)

# DAQ-Middleware Architecture

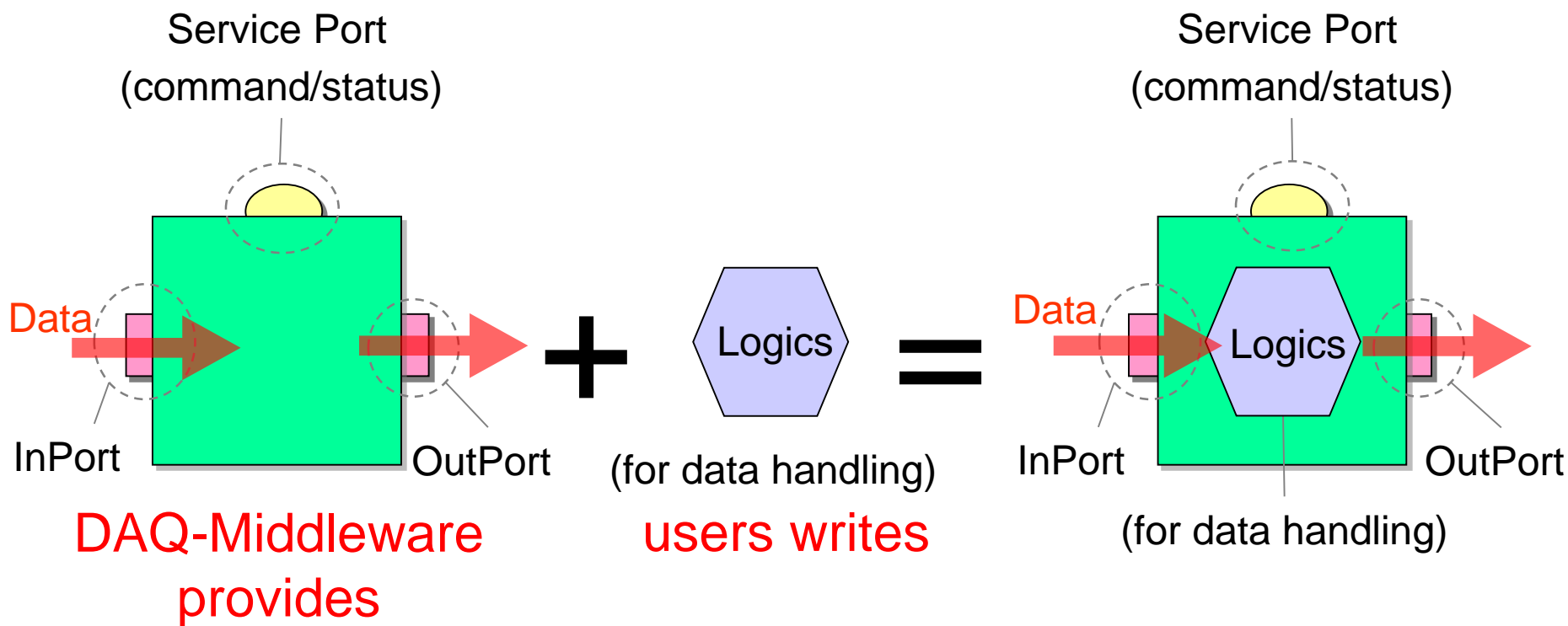
- Develop DAQ system by configuring DAQ components



software unit that achieve DAQ functions

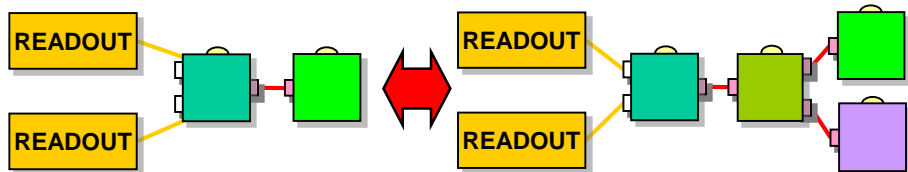


# DAQ Component



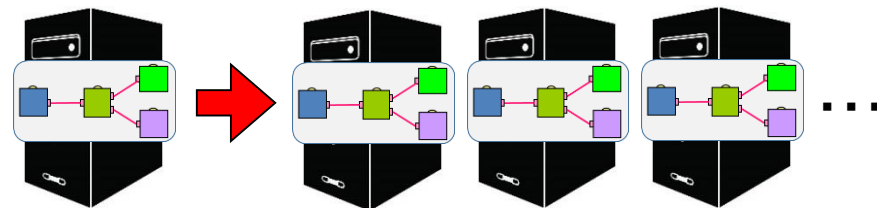
# DAQ Component Features

## Flexibility



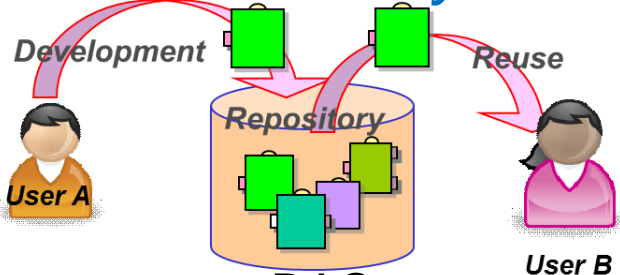
Users can flexibly change DAQ component combination.

## Scalability



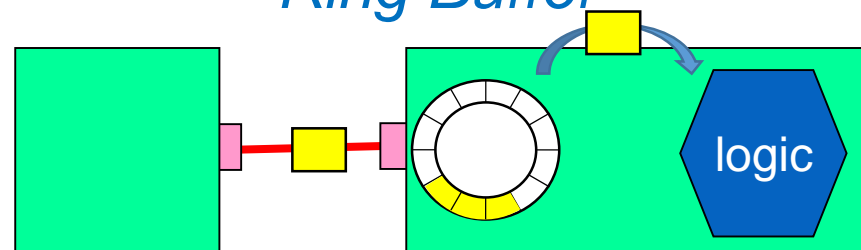
Users can improve performance by adding new PCs and deploying DAQ components.

## Reusability



Users can use a DAQ component in various DAQ system.

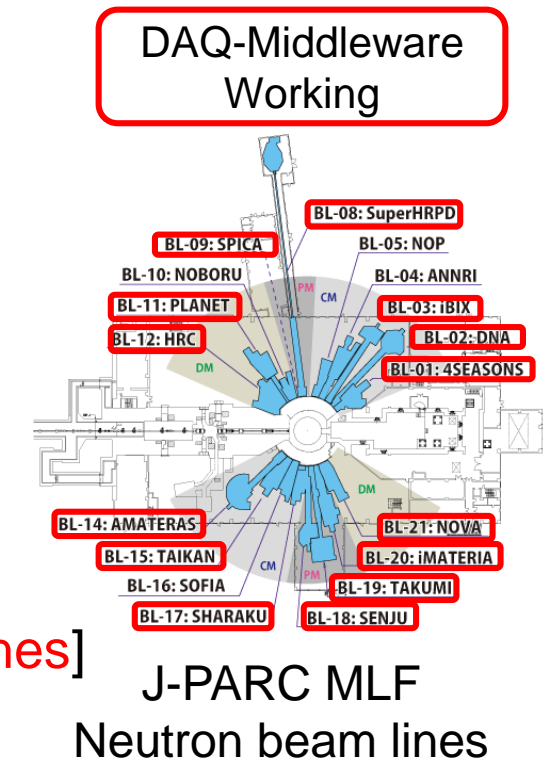
## Ring Buffer



DAQ component has a ring buffer. Users do not need to implement a buffer.

# List of DAQ-Middleware Users

- Experiments [**19 experiments**]
  - Material and Life Science
    - J-PARC MLF Neutron/Muon
    - DAQ system of Depth-resolved XMCD (KEK PF)
  - Elementary Particle/Nuclear Physics
    - J-PARCE16 Experiment
    - CANDLES
    - SuperNEMO (planning)
- Test benches for sensors and electronics [**9 test benches**]
  - Liquid Argon TPC
  - SOI Pixel Detector
  - ILC CCD Vertex
  - ADC-SiTCP
    - 2D gaseous detector with readout ASIC using printing technologies



and so on....

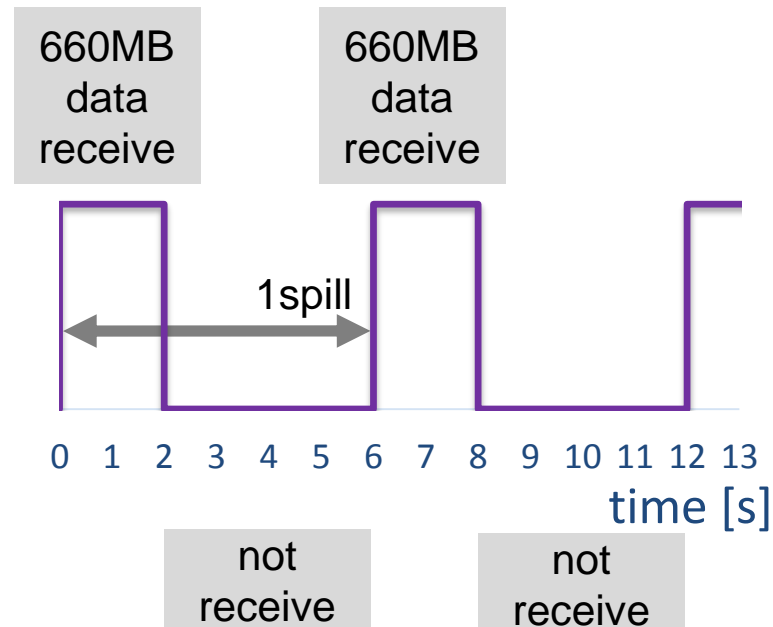


# 2. DAQ System at E16 Experiment

# Requirements

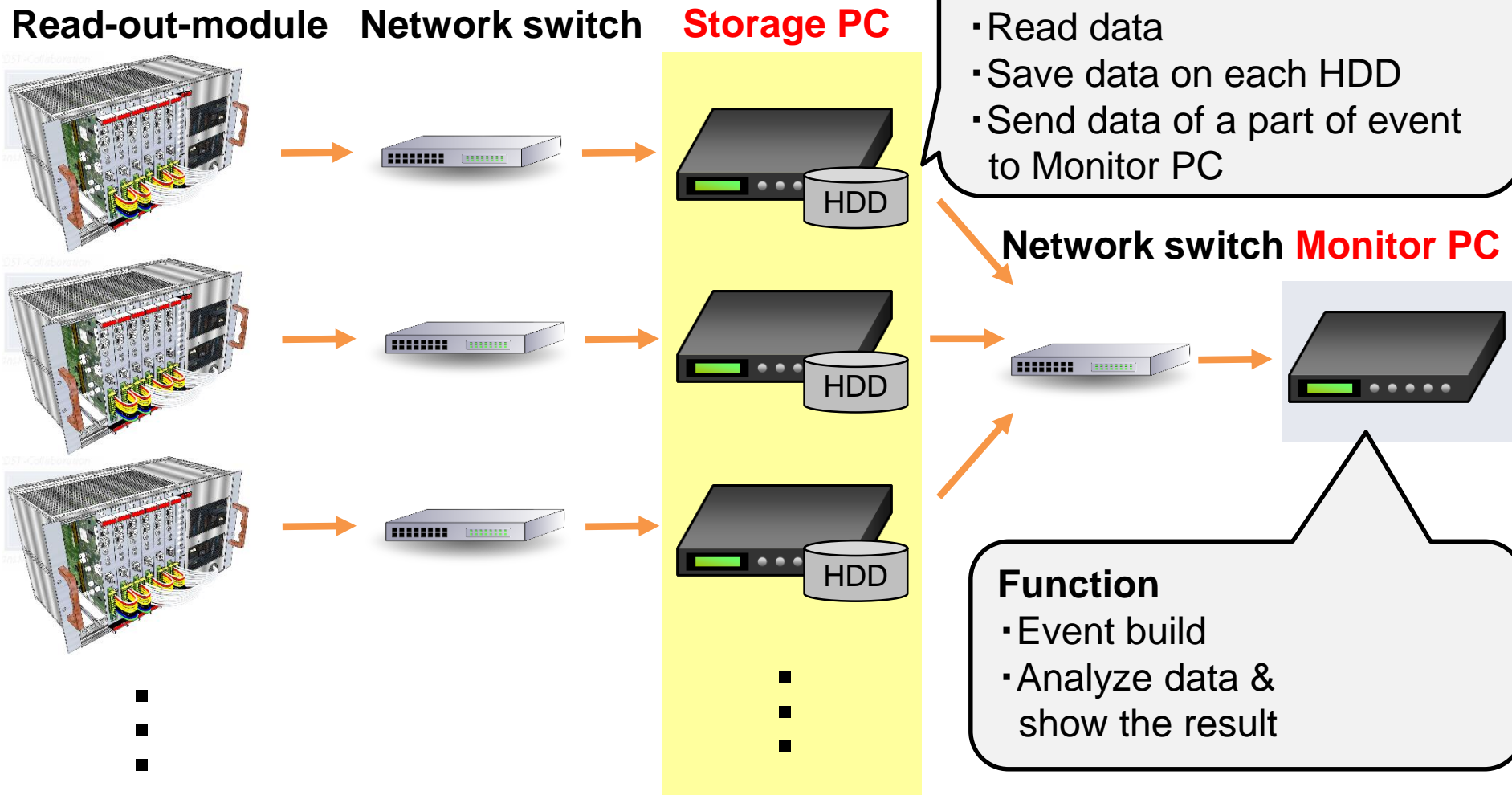
## Estimation of data transfer to DAQ PCs

Data rate per spill	660MB/spill
Trigger rate	(average) 1kHz
	(max) 2kHz
Instantaneous data rate	(average) 330MB/s
	(max) 660MB/s



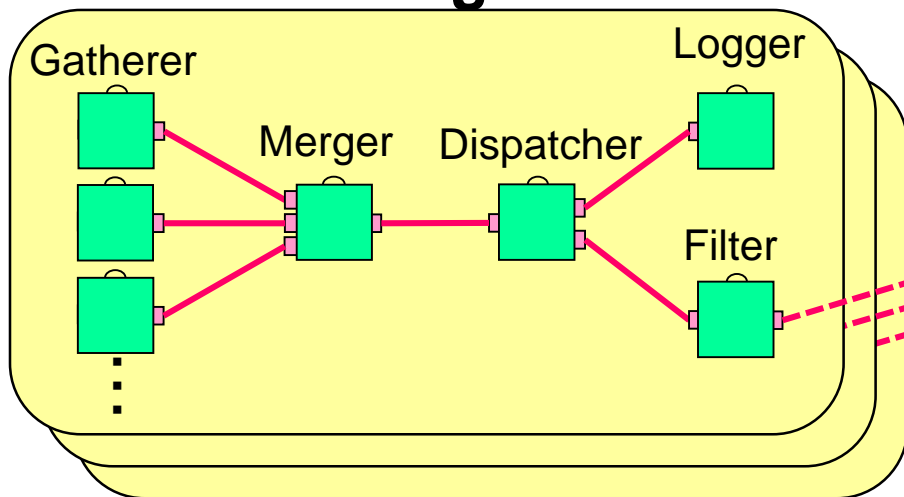
Trigger rate fluctuates due to beam rate variation.  
Event size per one event is almost constant.

# System Architecture

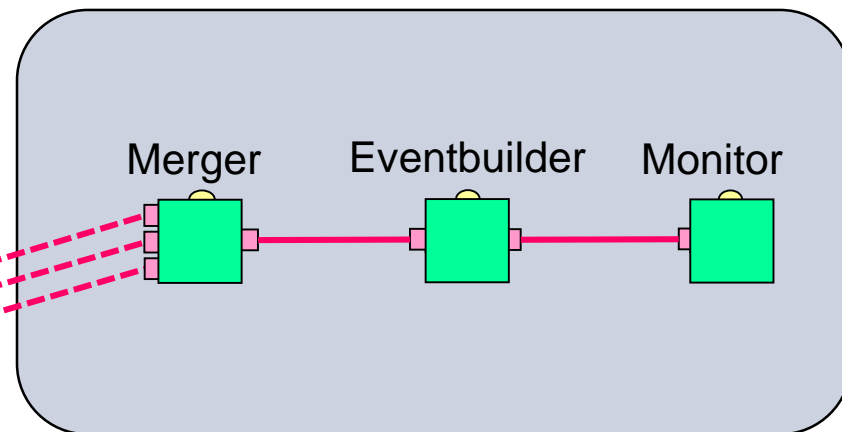


# DAQ Component Configuration

## Storage PC



## Monitor PC

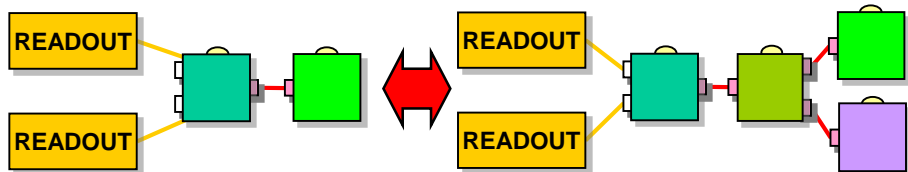


Component	Function
Gatherer	Read data
Merger	Receive data from multiple Gatherers
Dispatcher	Send data to Logger and Filter
Logger	Save data on each HDD
Filter	Send data of a part of event to Merger of Monitor PC

Component	Function
Merger	Receive data from multiple Filters
Eventbuilder	Event build
Monitor	Analyze data & show the result

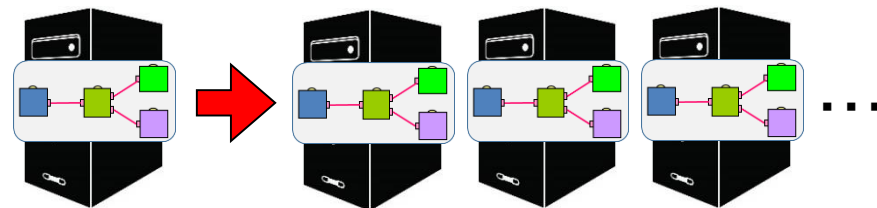
# DAQ Component Features

## Flexibility



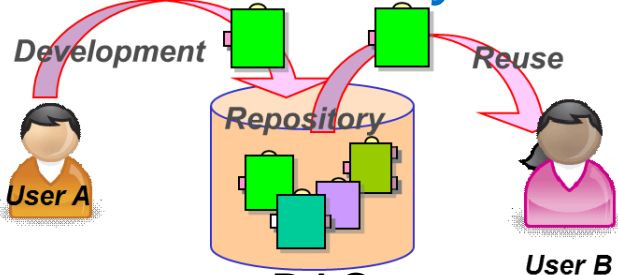
Users can flexibly change DAQ component combination.

## Scalability



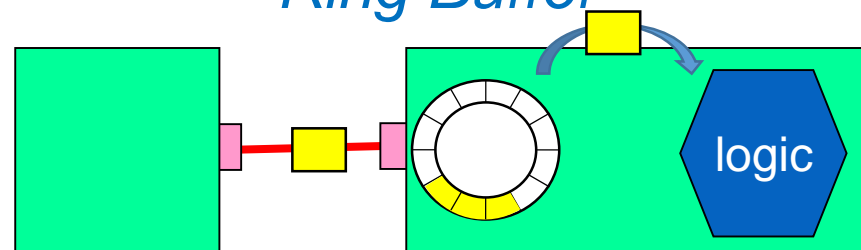
Users can improve performance by adding new PCs and deploying DAQ components.

## Reusability



Users can use a DAQ component in various DAQ system.

## Ring Buffer

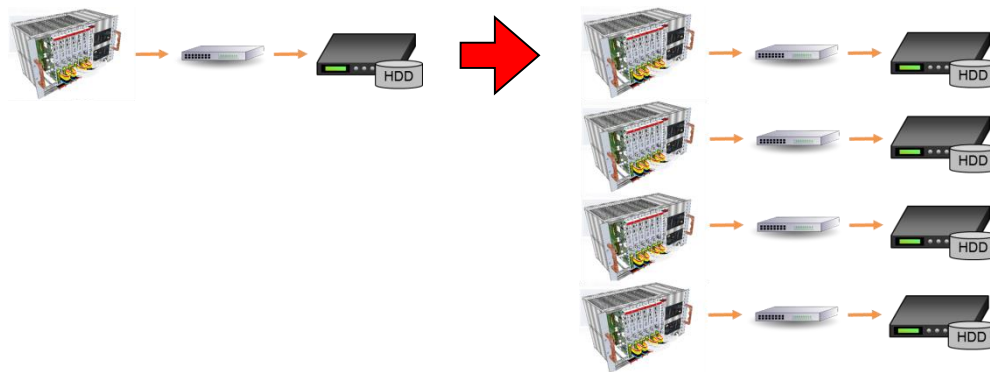


DAQ component has a ring buffer. Users do not need to implement a buffer.

# Advantage of DAQ Component Features

## Flexibility & Scalability

If data volume increase by detector upgrade, we add Storage PC.



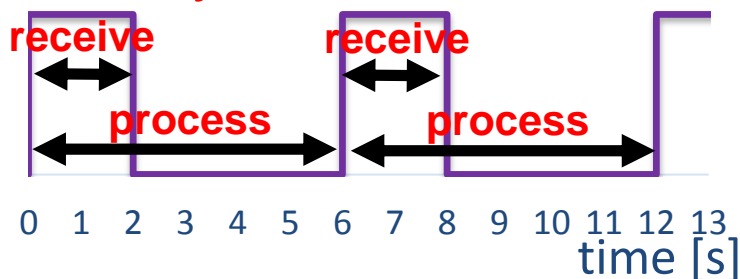
## Reusability

Using DAQ component which is prepared, we saved time and effort for development.



## Ring Buffer

We can use no event time effectively.



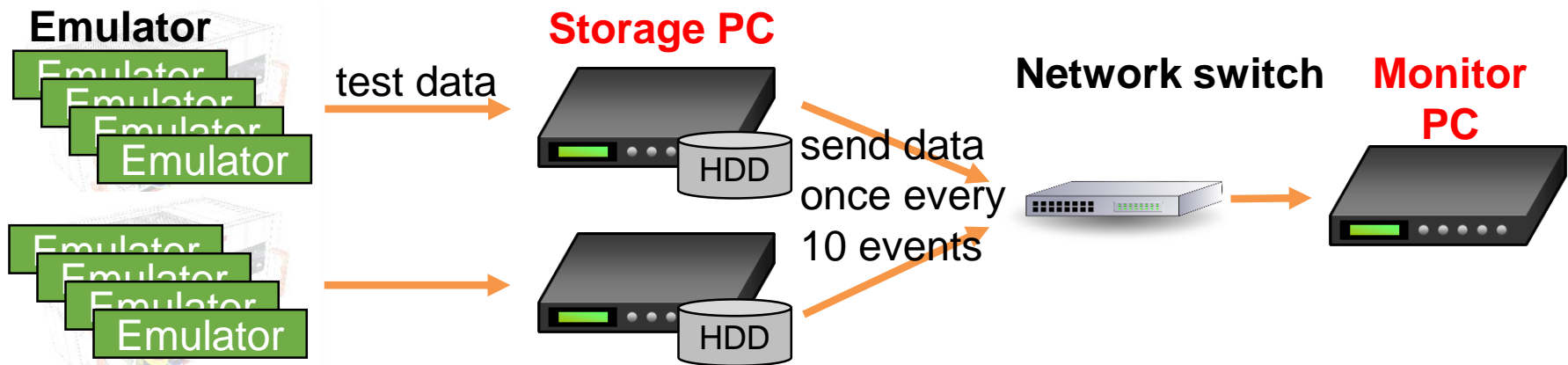
# 3. DAQ System Performance

# Purpose & Method

## Purpose

Evaluation of a total throughput using a DAQ-Middleware for the J-PARC E16 experiment

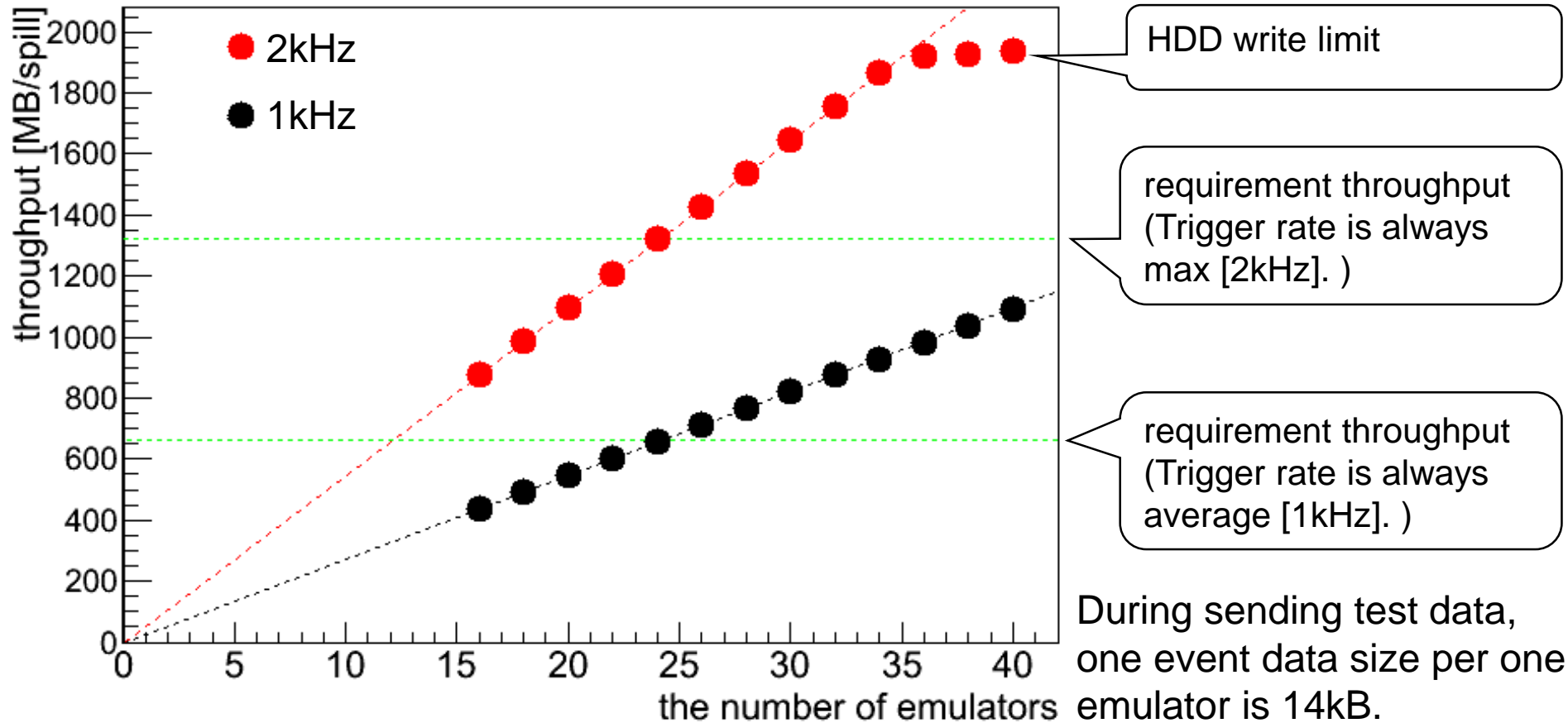
## Method



- Emulators run with 1 cycle per 6 seconds. During one cycle, emulators send test data for 2 seconds and do not send test data for 4 seconds.
- Monitor PC shows data value of a part of data regularly.
- We changed the number of emulators to change transfer data volume and measured processing data speed.



# Result



The result met the requirements of the E16 experiment!

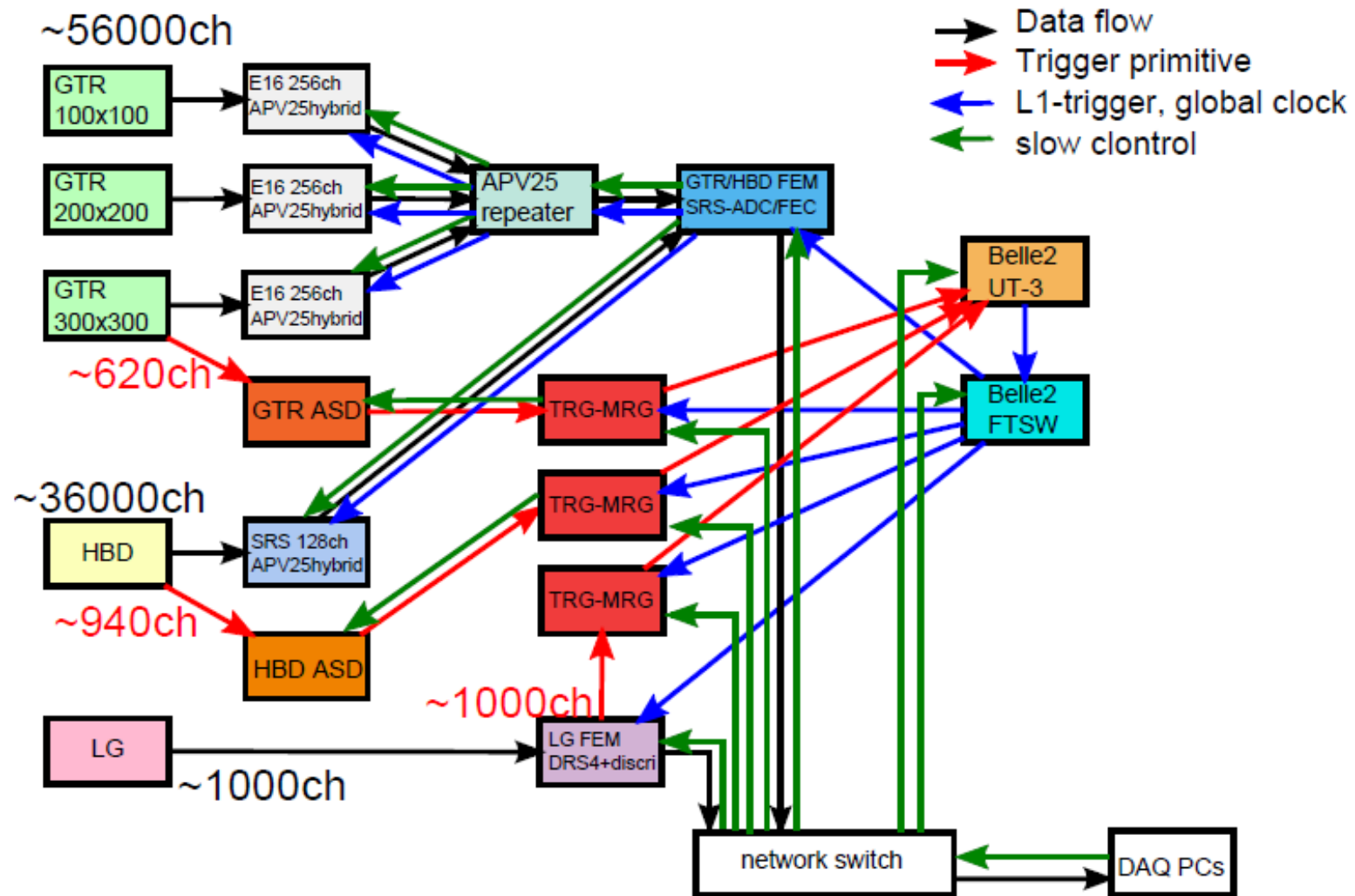
# 4. Summary and Future Plan

# Summary and Future Plan

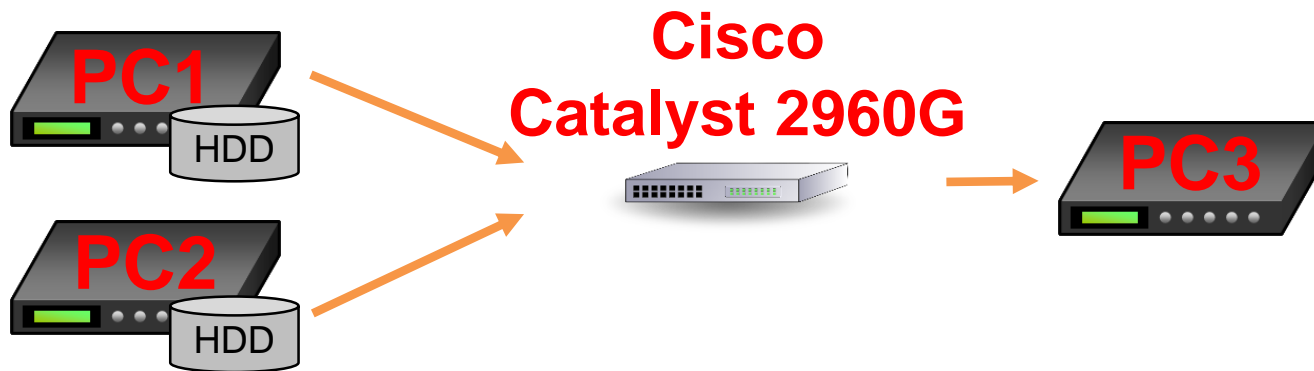
- DAQ-Middleware is a framework for network based DAQ software.
- DAQ component has following features.
  - Flexibility
  - Scalability
  - Reusability
  - Ring Buffer
- We have developed DAQ system for E16 experiment by using DAQ-Middleware.
- The requirements from E16 experiment have been met.
- In the future, we are going to connect DAQ system to read out module, and evaluate the system.

# Backup

# J-PARC E16 Experiment DAQ system overview



# Evaluation Environment



## PC1

## PC2

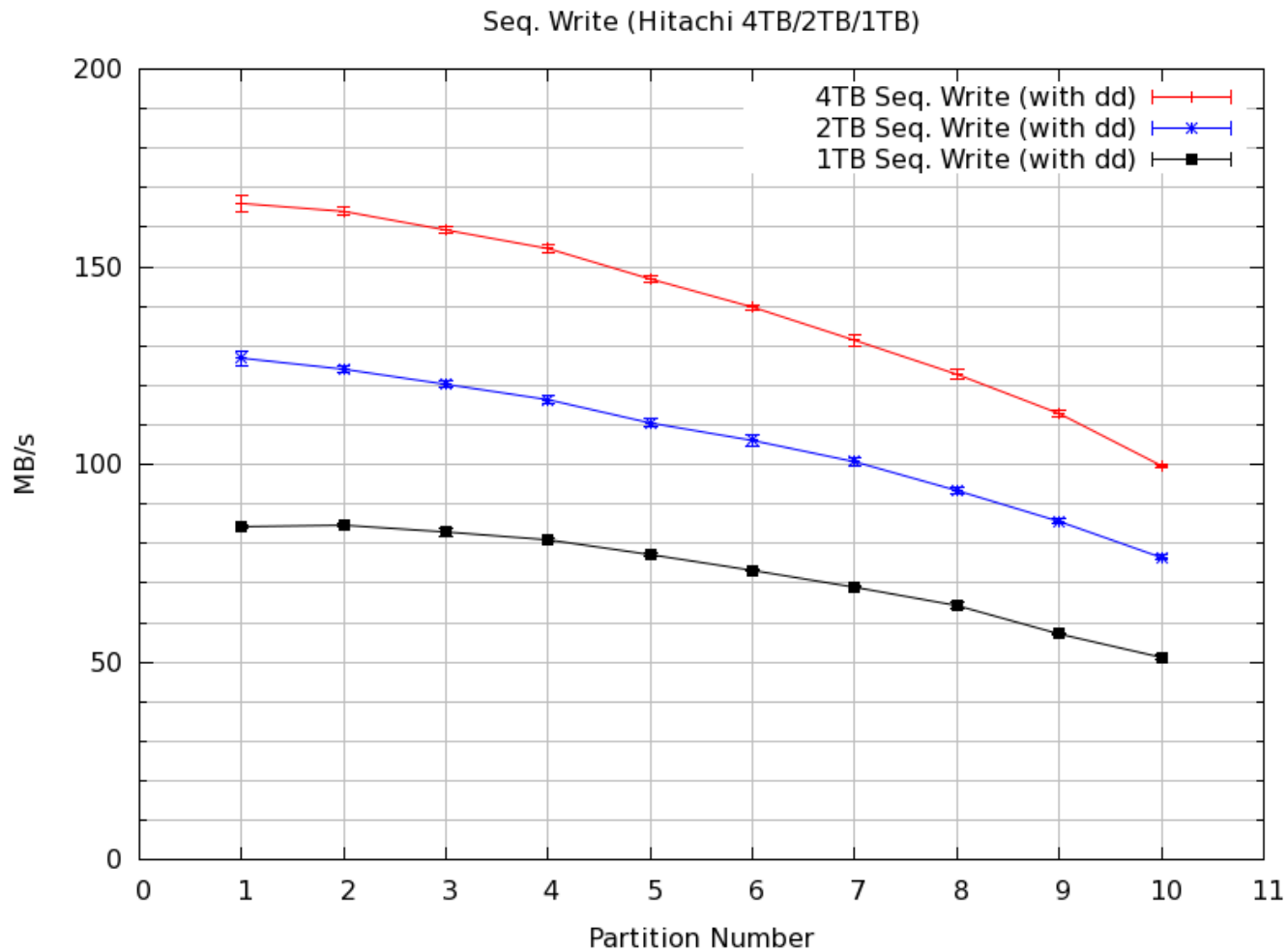
## PC3

CPU	Intel(R) Xeon(R) X5650 @ 2.67GHz 6Cores
Memory	24GB
Network	1Gbps x 5
OS	Scientific Linux 6.4
HDD	Hitachi HDS724040ALE6 4TB

CPU	Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz 6Cores
Memory	32GB
Network	1Gbps x 5
OS	Scientific Linux 6.6
HDD	Hitachi HDS724040ALE6 4TB

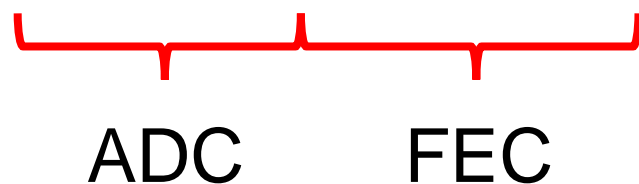
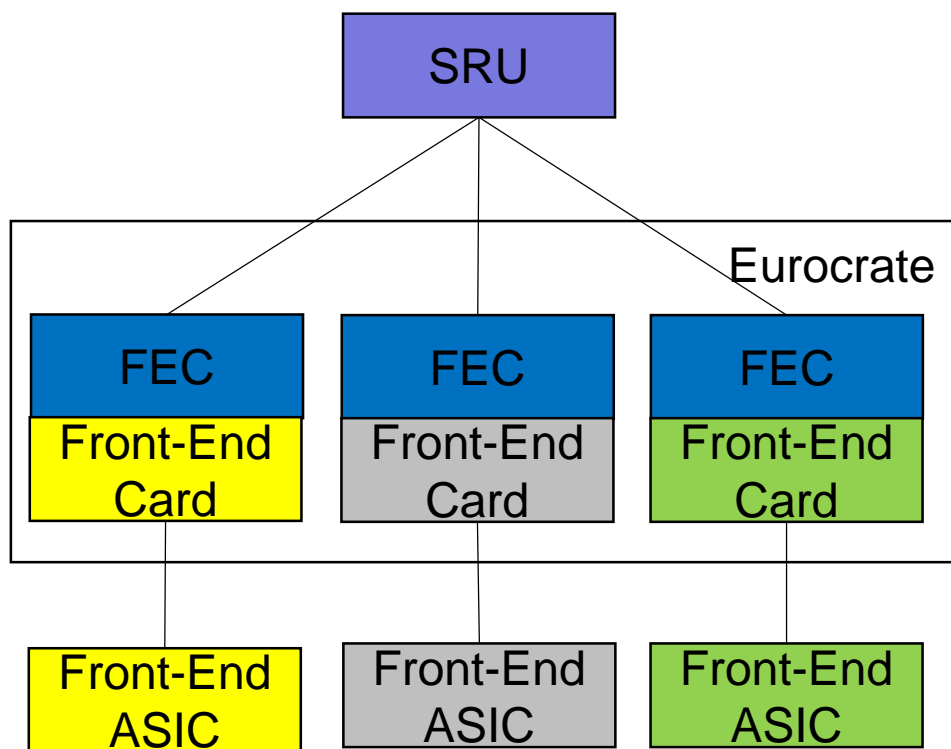
CPU	Intel(R) Xeon(R) CPU E3-1220 v3 @ 3.10GHz
Memory	8GB
Network	2Gbps
OS	Scientific Linux 6.6

# HDD write speed check



# SRS (Scalable Readout System)

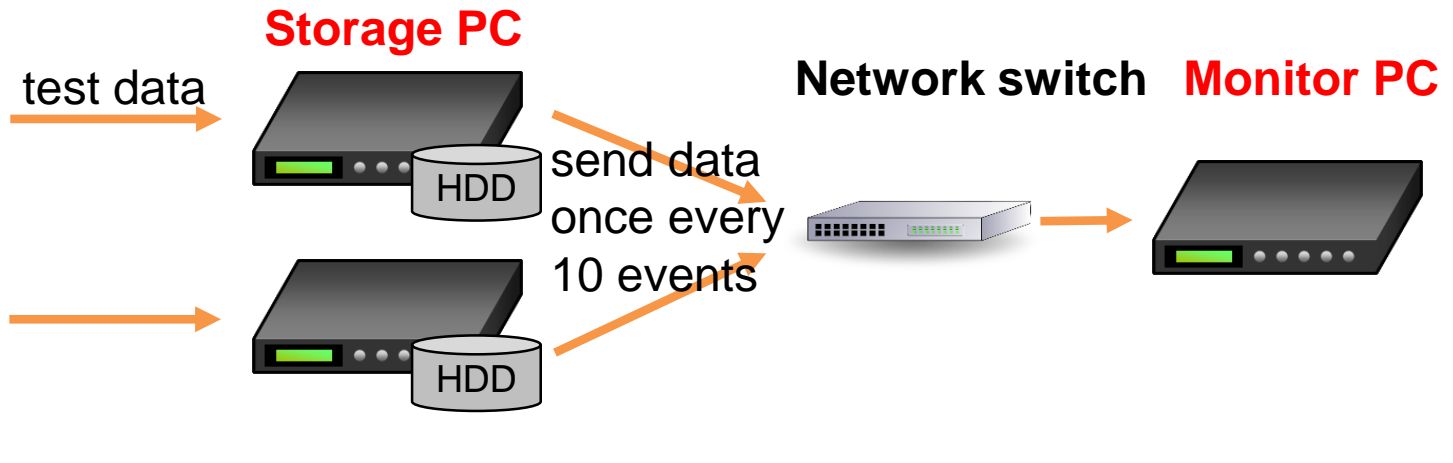
SRS is general purpose multi-channel readout system.  
User can choose variety of frontend chips.





# Evaluation by using SRS

**SRS  
ADC/EFC**



ADC/FEC send test data.

Maximum transfer speed of test data is 1Gbps + 1Gbps.

We configured that DAQ PC could process all of data.

# Storage PC Performance

CPU Intel(R) Xeon(R) X5650 @ 2.67GHz 6Cores  
Memory 24GB  
Network 1Gbps x 10  
OS Scientific Linux 6.4  
SSD Intel SSD520Series 240GB

Emulator

test data  
trigger rate = 2kHz  
1 event size = 14kB

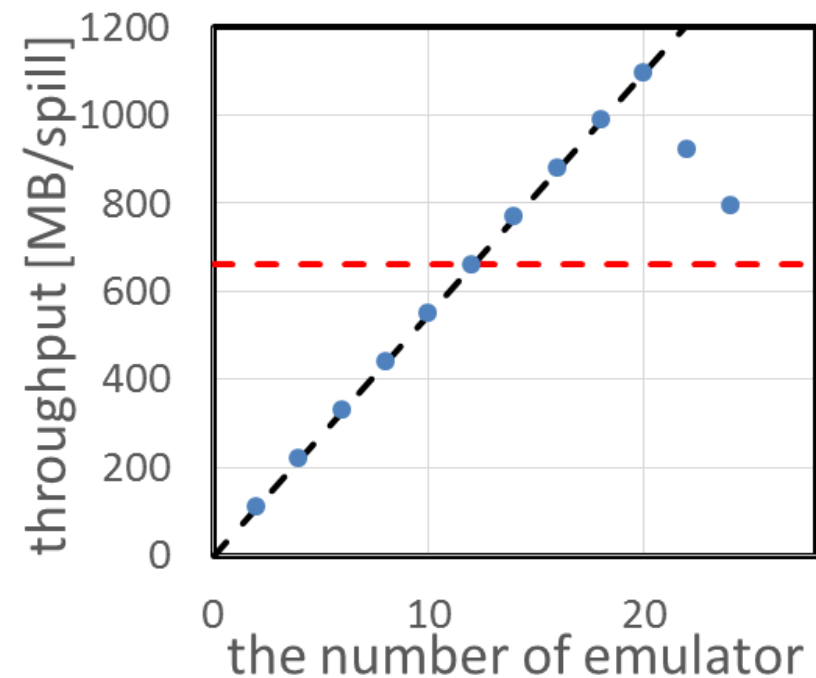
Storage PC

write data on SSD

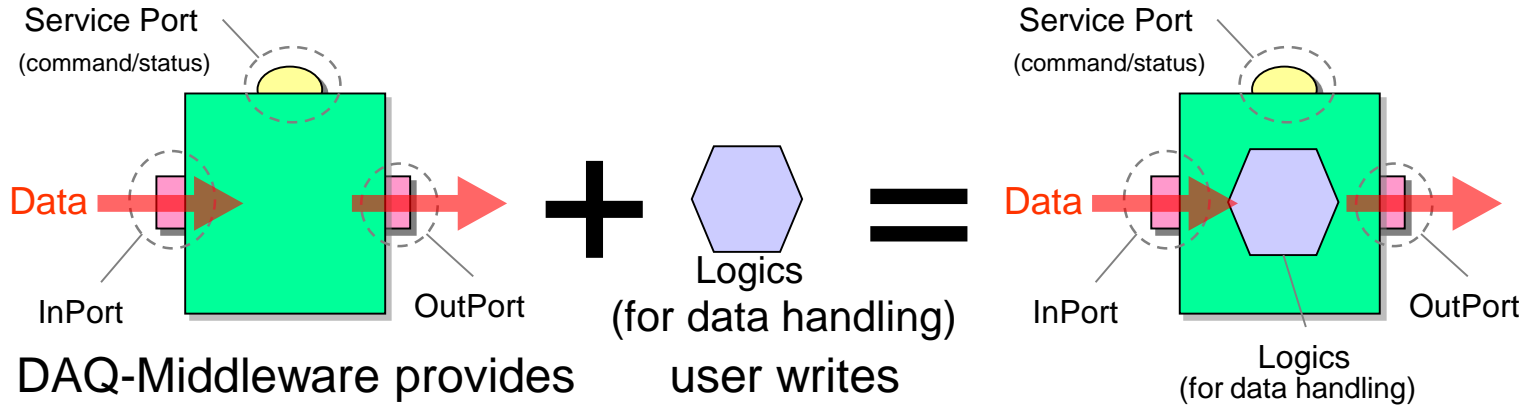
Test PC

only receive data

Throughput of 1 Storage PC is 1000MB/spill.



# DAQ Component & Configuration Example



## Examples of DAQ component combination

