

# DAQ-Middleware概論

千代浩司

高エネルギー加速器研究機構  
素粒子原子核研究所

ネットワーク読み出しモジュールで、接続すると  
データがくる場合には

```
nc 192.168.0.16 24 > datafile
```

nc - arbitrary TCP and UDP connections and  
listeners

```
nc 192.168.0.16 24 | histo_prog
```

```
nc 192.168.0.16 24 | tee datafile  
| histo_prog
```

# アウトライン

- DAQ-Middlewareの紹介
  - 実際に使用されているところの紹介
  - 開発体制
  - 性能測定
  - その他
- 
- DAQ-Middlewareホームページ  
<http://daqmw.kek.jp/>

# DAQ-Middlewareの 紹介

# DAQ-Middlewareとは (1)

- 汎用のネットワークベースデータ収集 (DAQ) ソフトウェアフレームワーク
  - 再利用性を考慮
  - DAQコンポーネントでいろいろな状況に対応
  - 全体の枠組は統一されている(フレームワーク)
- ターゲット
  - 中小規模実験
  - テストベッド (センサー、読み出しモジュール)

# DAQ-Middlewareで提供するもの

- データ収集パス
  - 複数のDAQコンポーネントでデータを集める



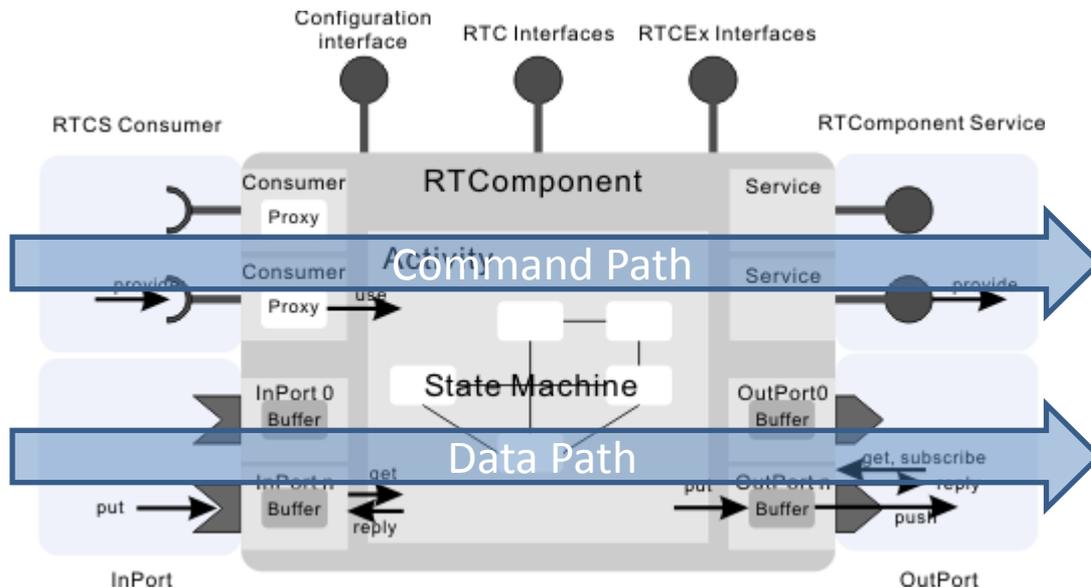
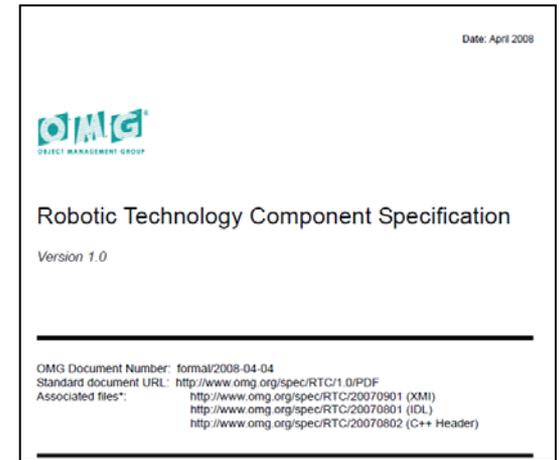
- ランコントロール
  - スタート、ストップ、状態遷移
- システムコンフィギュレーション
  - DAQコンポーネントの組み合わせを指定する
  - その他必要なパラメータを指定する

# 背景、構想

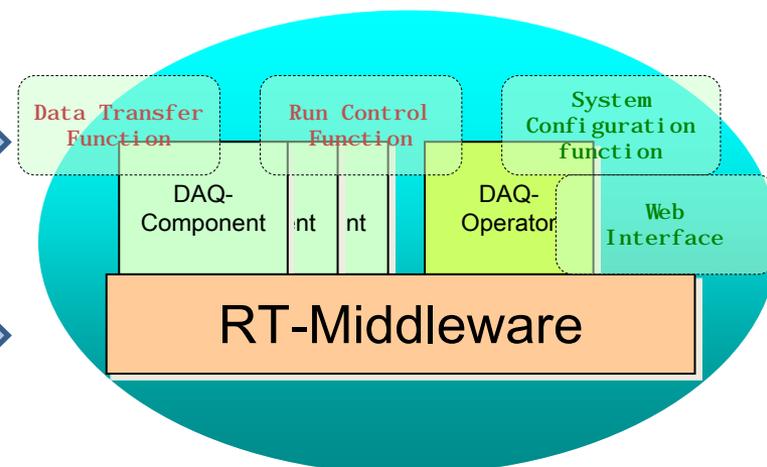
- 背景
  - 従来DAQシステムでのソフトウェアの再利用化はドライバ、ライブラリレベルでおこなわれてきた。
  - 扱うデータの増大、使う計算機の数が増えてきてDAQシステムを構築するのがむずかしくなってきた。
  - どんな実験にも対応できるように、抽象化、汎用化してしまうとデータ収集効率が落ちてしまう。
- 解決案
  - ドライバ、ライブラリとDAQシステムの間コンポーネントという中間層を作り、実験毎の違いを吸収、収集効率を確保し、
  - システムの枠組みは普遍であるDAQフレームワークを作ればよいのではないか？

# DAQ-Middleware (2)

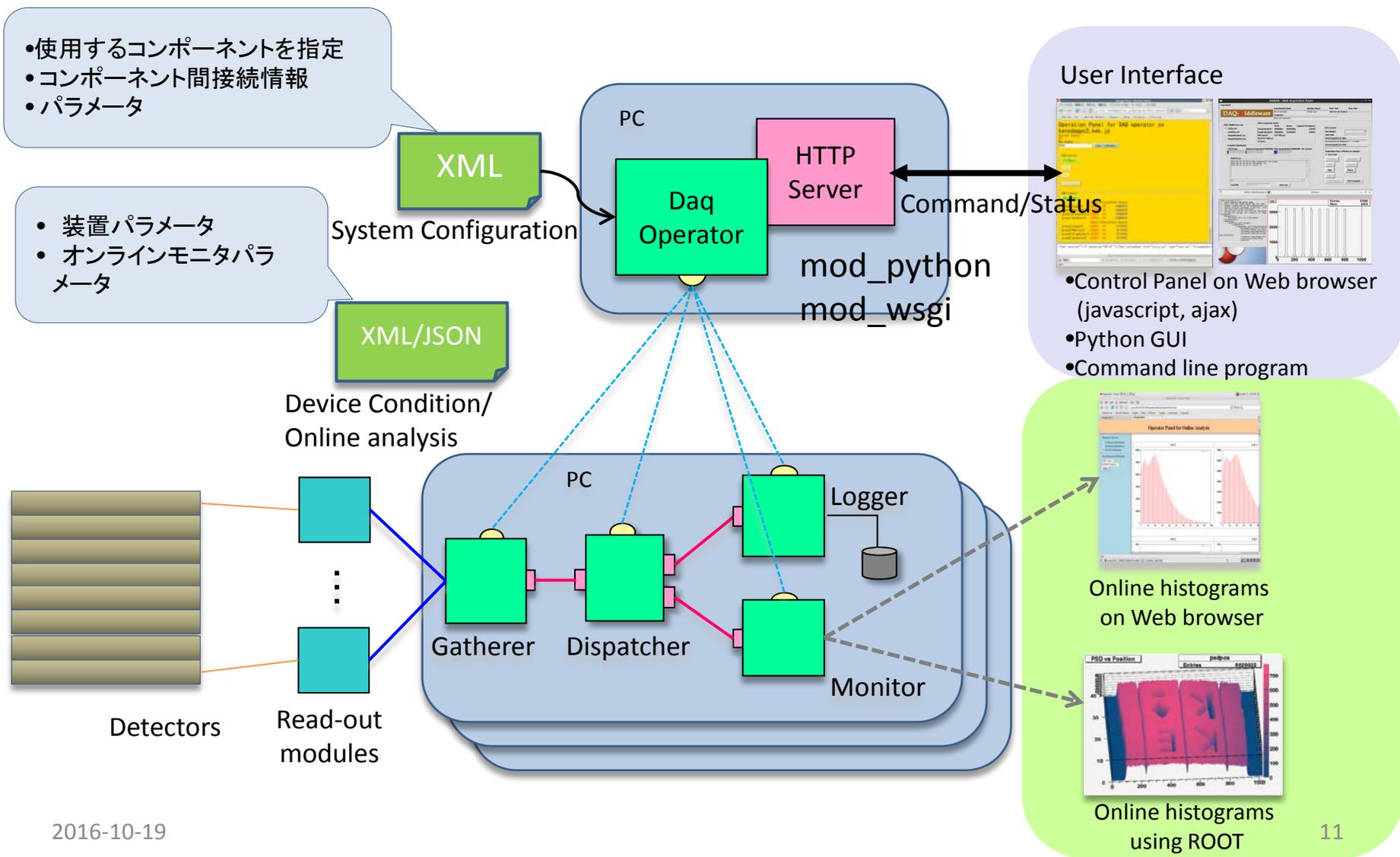
- RT(Robot Technology)-Middlewareをデータ収集用に拡張
- RT-Middleware
  - ネットワークロボットシステムの構築のためのソフトウェア共通プラットフォーム
  - 産総研知能システム研究部門・タスクインテリジェンス研究グループが開発
  - 複数のコンポーネントが通信してひとつの機能を実現する
  - そのソフトウェアコンポーネントの仕様は国際標準規格(OMG)
  - 我々は2006年から産総研と共同研究を行っている



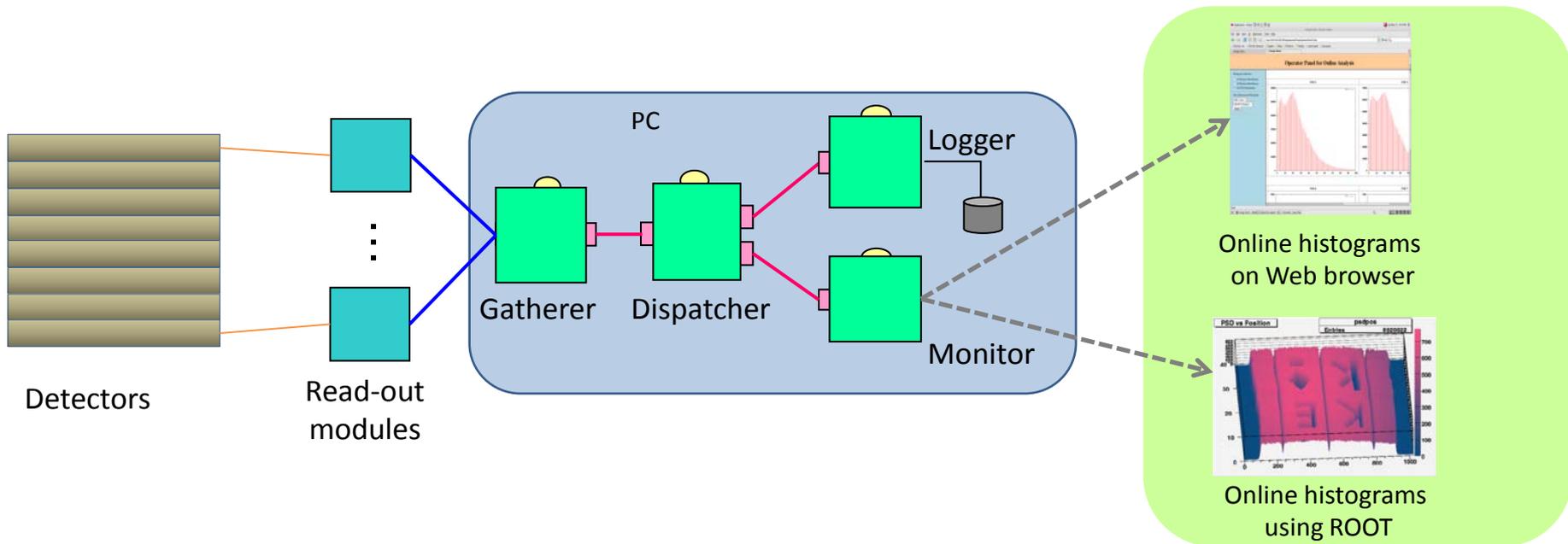
## DAQ-Middleware



# DAQ-Middleware構成図



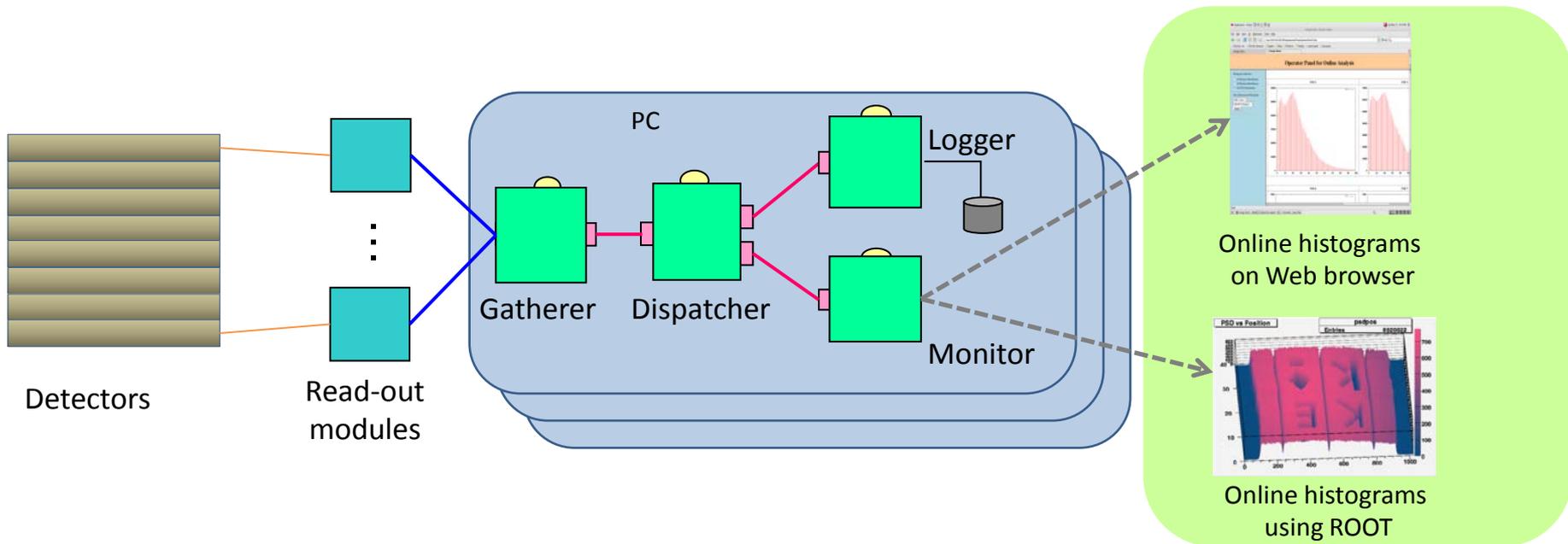
# データ収集パス



複数のDAQコンポーネントを組み合わせてデータ収集パスを作る。

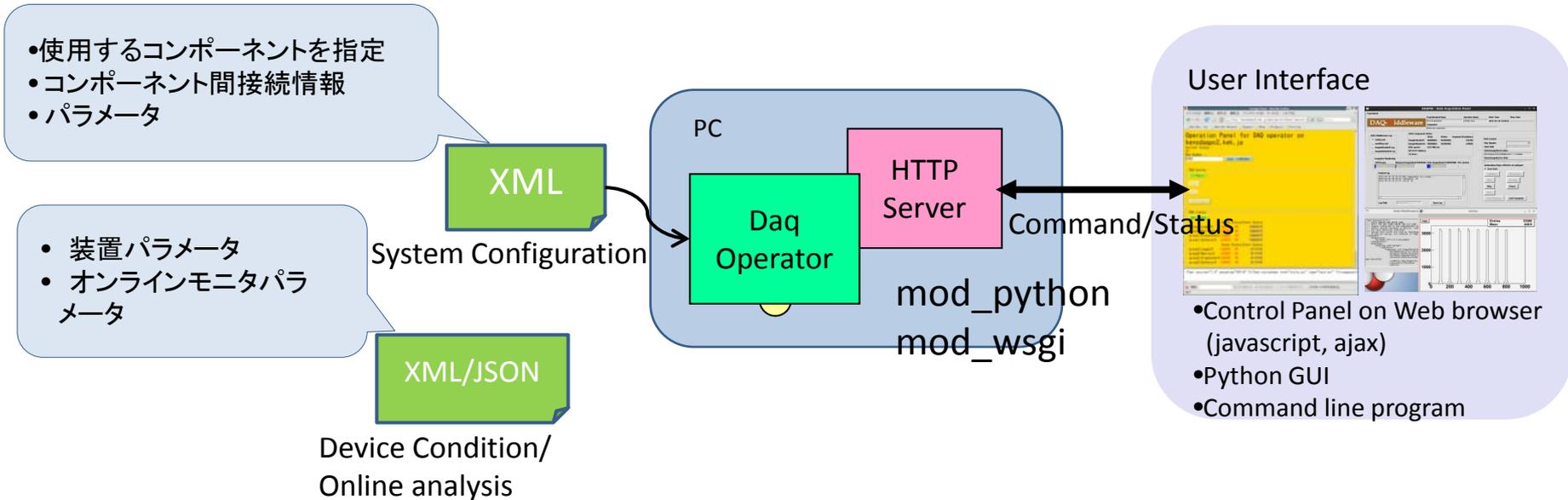
- (red line) DAQ-Middlewareで提供するパス(ネットワーク接続)
- (blue line) リードアウトモジュール - gatherer間はネットワークだったり  
その他だったりする(リードアウトモジュールによる)

# データ収集パス

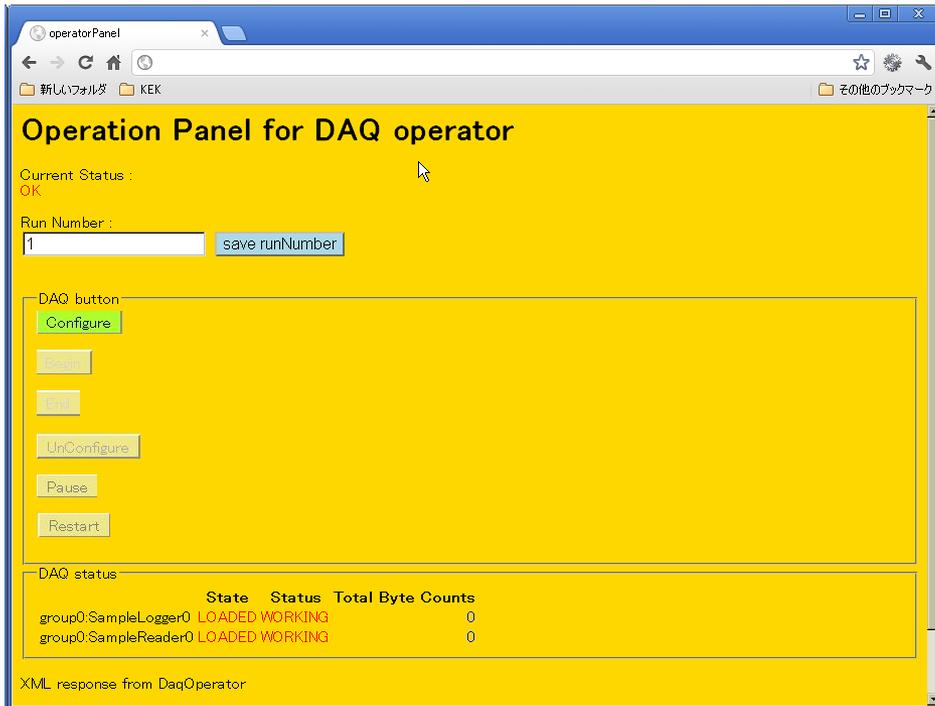


リードアウトモジュールが多い場合は複数セット用意することで対応する

# ランコントロール



- DaqOperator: DAQコンポーネントを統括する
- DaqOperatorに対する指示はhttpで行う
  - 既存のものがあるときはそれがhttpで通信するようにすれば使える



Operation Panel for DAQ operator

Current Status : OK

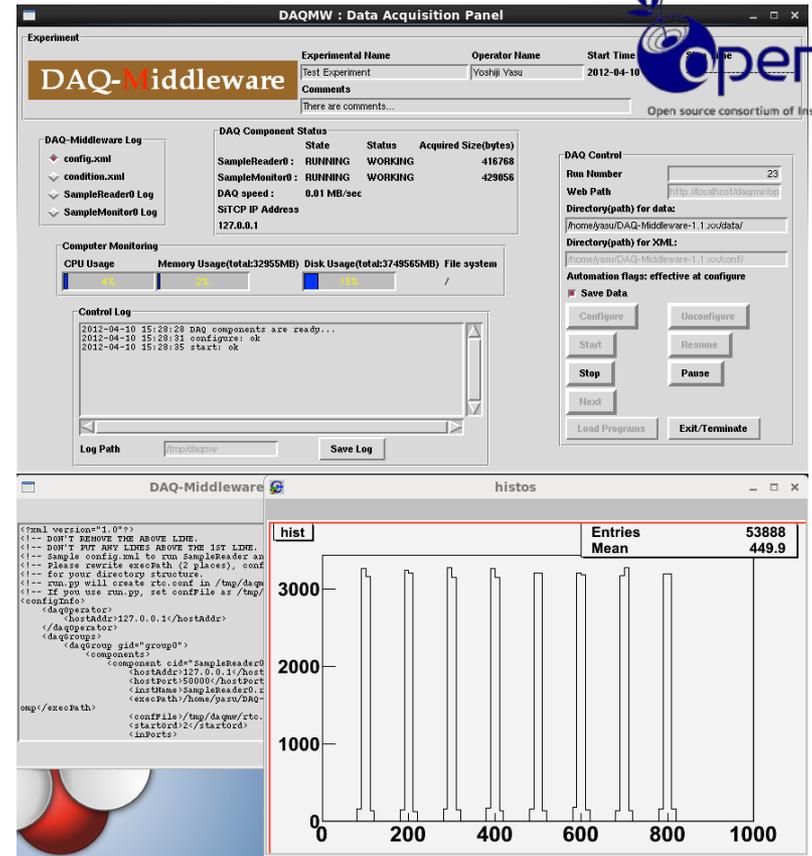
Run Number : 1

DAQ button

State	Status	Total	Byte Counts
group0.SampleLogger0	LOADED WORKING		0
group0.SampleReader0	LOADED WORKING		0

XML response from DaqOperator

```
% daqcom http://localhost/daqmw/operatorPanel/ -g state
[('g', 'state')] []
LOADED
% daqcom http://localhost/daqmw/operatorPanel/ -c
[('c', '')] []
<?xml version="1.0" encoding="UTF-8" ?><?xml-stylesheet href="style.xsl" type="text/xsl" ?><response><methodName>Params</methodName><returnValue><result><status>OK</status><code>0</code><className/><name/><methodName/><messageEng/><messageJpn/></result></returnValue></response>
% daqcom http://localhost/daqmw/operatorPanel/ -b 1
[('b', '1')] []
OK
<?xml version="1.0" encoding="UTF-8" ?><?xml-stylesheet href="style.xsl" type="text/xsl" ?><response><methodName>Begin</methodName><returnValue><result><status>OK</status><code>0</code><className/><name/><methodName/><messageEng/><messageJpn/></result></returnValue></response>
% █
```



DAQ-Middleware : Data Acquisition Panel

Experiment: Test Experiment, Operator Name: Yoshiji Yasu, Start Time: 2012-04-10

DAQ Component Status	State	Status	Acquired Size(bytes)
SampleReader0	RUNNING	WORKING	416768
SampleMonitor0	RUNNING	WORKING	429856

Computer Monitoring: CPU Usage, Memory Usage (total: 32955MB), Disk Usage (total: 3748565MB), File system

Control Log: 2012-04-10 15:28:28 DAQ components are ready... 2012-04-10 15:28:31 configure: ok 2012-04-10 15:28:35 start: ok

DAQ Control: Run Number: 23, Web Path: http://localhost/daqmw/

Automation flags: effective at configure

Save Data:  Save Data

Configure, Unconfigure, Start, Resume, Stop, Pause, Next, Load Programs, Exit/Terminate

DAQ-Middleware histos

Entries: 53888, Mean: 449.9

hist

3000  
2000  
1000  
0

0 200 400 600 800 1000

## ランコントロールインターフェイス

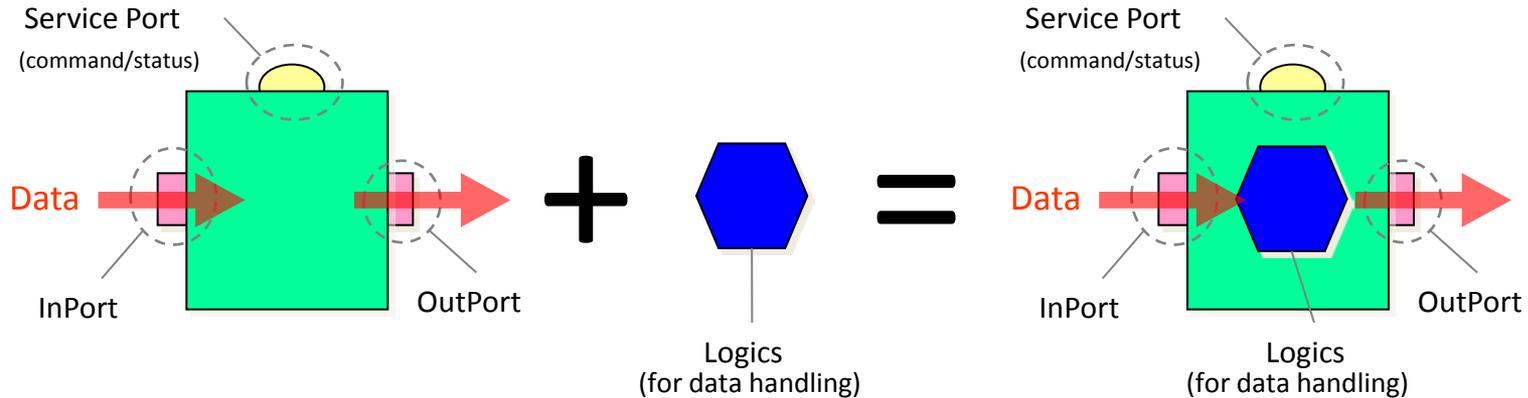
- Web browser UI
- python TK UI
- Linux command line

# システムコンフィギュレーション

## XMLで記述する

```
<configInfo>
  <daqOperator>
    <hostAddr>127.0.0.1</hostAddr>
  </daqOperator>
  <daqGroups>
    <daqGroup gid="group0">
      <components>
        <component cid="SampleReader0">
          <hostAddr>127.0.0.1</hostAddr>
          <hostPort>50000</hostPort>
          <instName>SampleReader0.rtc</instName>
          <execPath>/home/daq/MyDaq/SampleReader/SampleReaderComp</execPath>
          <confFile>/tmp/daqmw/rtc.conf</confFile>
          <startOrd>2</startOrd>
          <inPorts>
          </inPorts>
          <outPorts>
            <outPort>samplerreader_out</outPort>
          </outPorts>
          <params>
            <param pid="srcAddr">127.0.0.1</param>
            <param pid="srcPort">2222</param>
          </params>
        </component>
      </components>
    </daqGroup>
  </daqGroups>
</configInfo>
```

# DAQコンポーネント

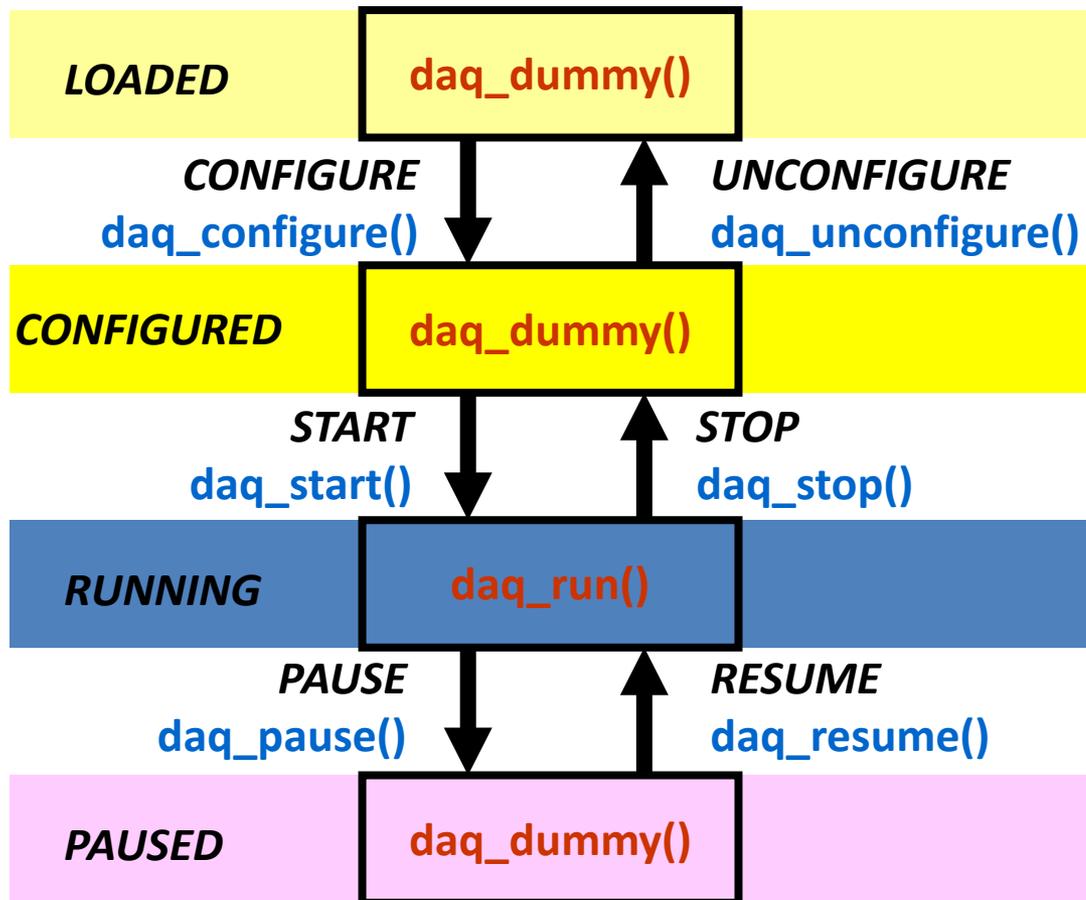


- DAQコンポーネントを組み合わせてDAQシステムを構築する。
- 上流からのデータを読むにはInPortを読む。
- データを下流に送るにはOutPortに書く。
- DAQコンポーネント間のデータ転送機能はDAQ-Middlewareが提供する
- ユーザーはコアロジックを実装することで新しいコンポーネントを作成できる。

コアロジックの例：

- リードアウトモジュールからのデータの読み取りロジック
- ヒストグラムの作成ロジック

# コンポーネント状態遷移



各状態(LOADED, CONFIGURED, RUNNING, PAUSED)にある間、対応する関数が繰り返し呼ばれる。

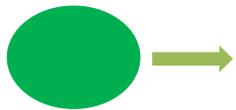
状態遷移するときは状態遷移関数が呼ばれる。

状態遷移できるようにするためには、daq\_run()等は永遠にそのなかでブロックしてはだめ。  
(例: Gathererのソケットプログラムでtimeoutつきにする必要がある)

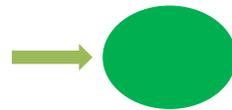
各関数を実装することでDAQコンポーネントを完成させる。

# コンポーネント間通信での分類

Source Type



Sink Type



Filter Type



Dispather Type

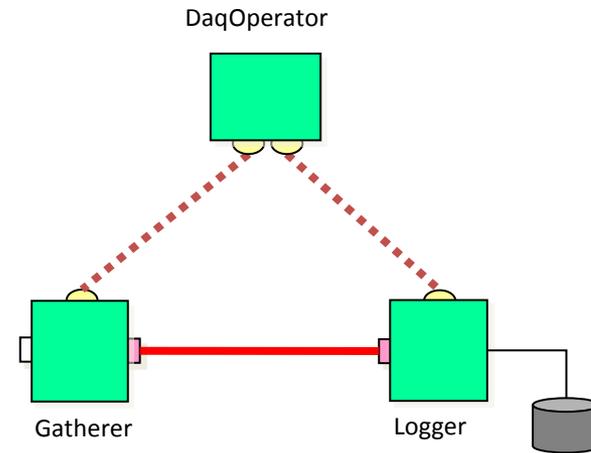
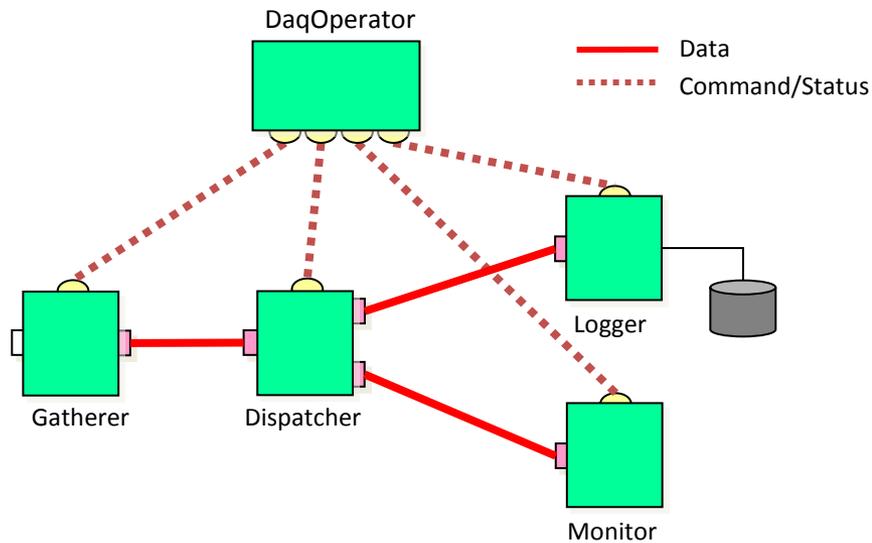


Merger Type

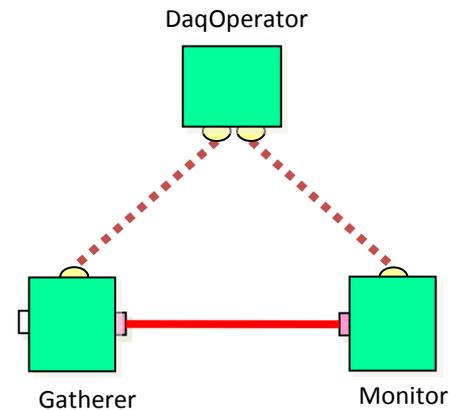
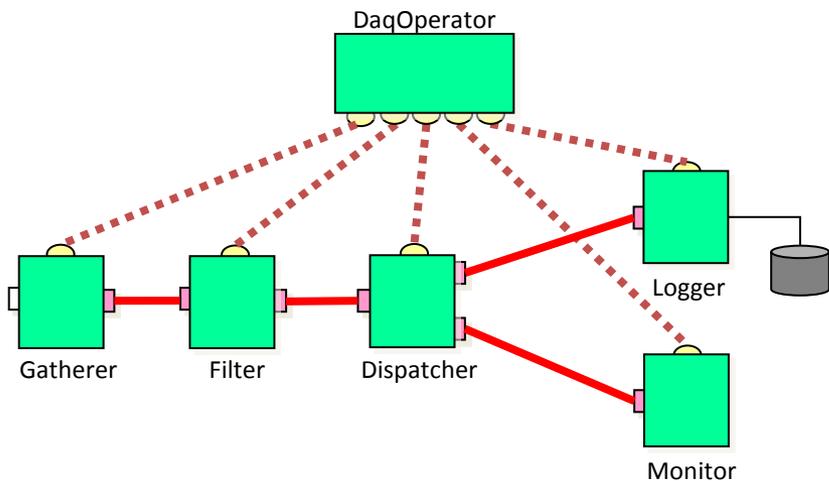


- Source Type (Gatherer)
- Sink Type (Logger, Monitor)
- Dispatcher Type

# DAQコンポーネント 構成例(1)



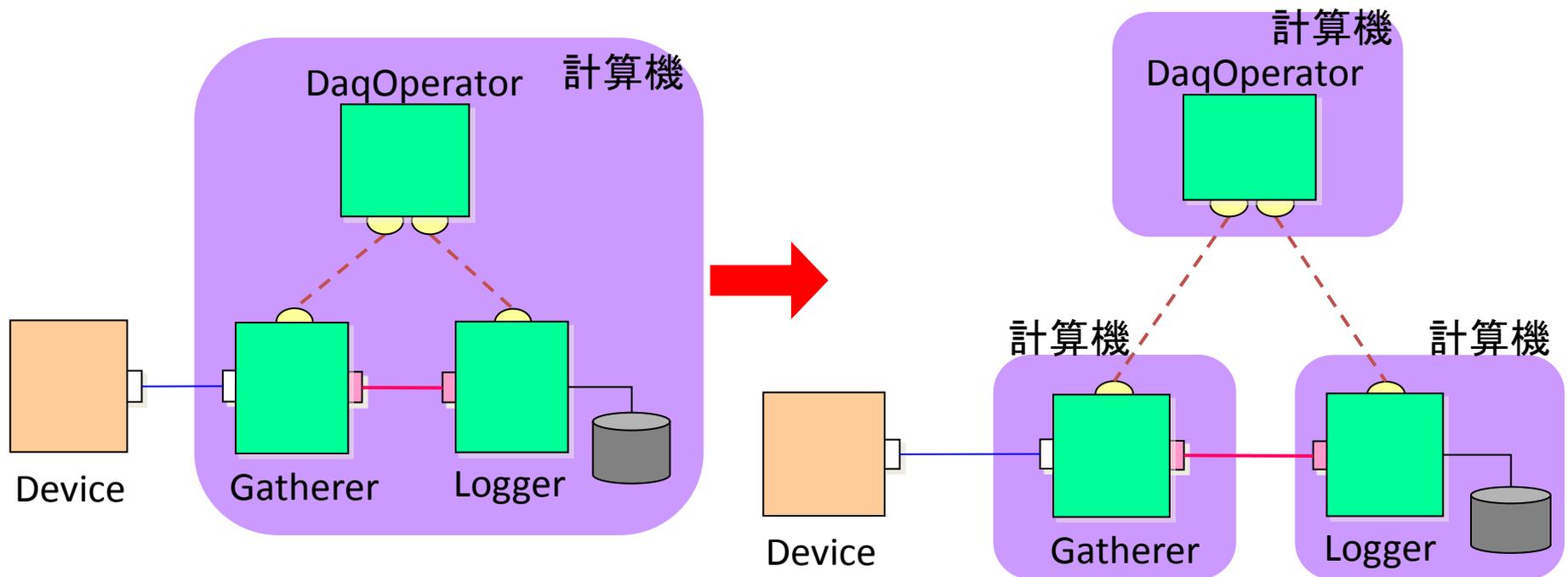
モニターなしでデータをディスクに  
セーブする



データセーブなしでオンライン  
モニターする

# DAQコンポーネント構成例 (2)

## ネットワーク透過性

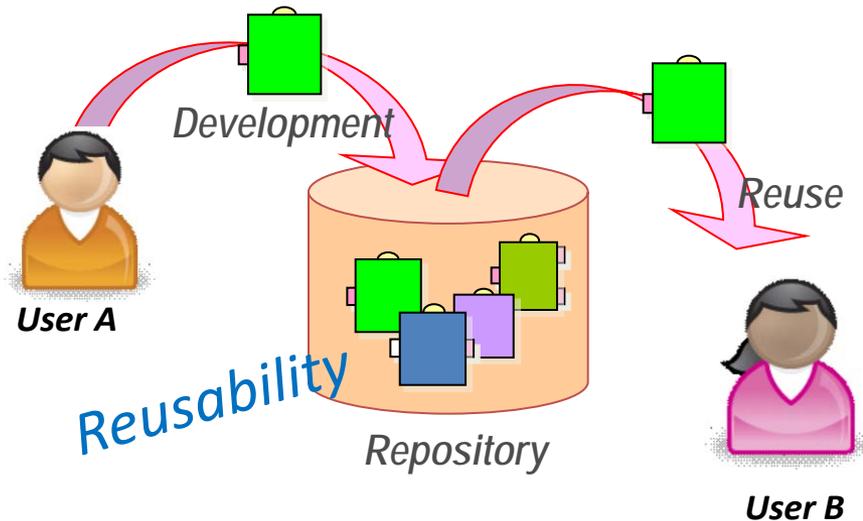
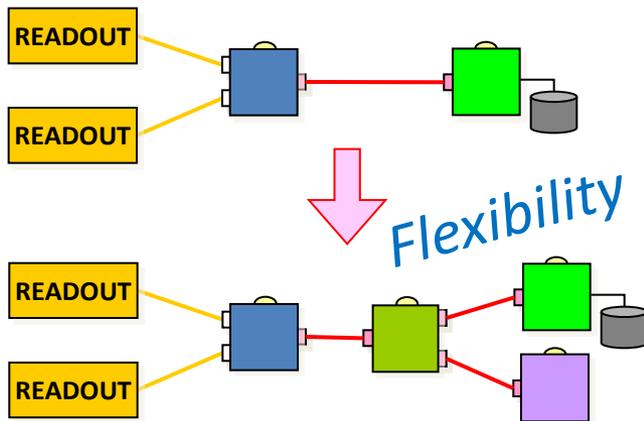
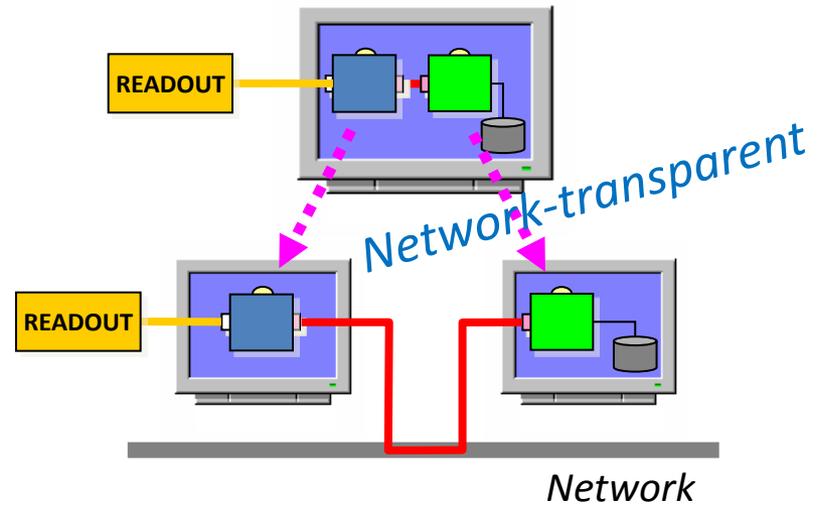
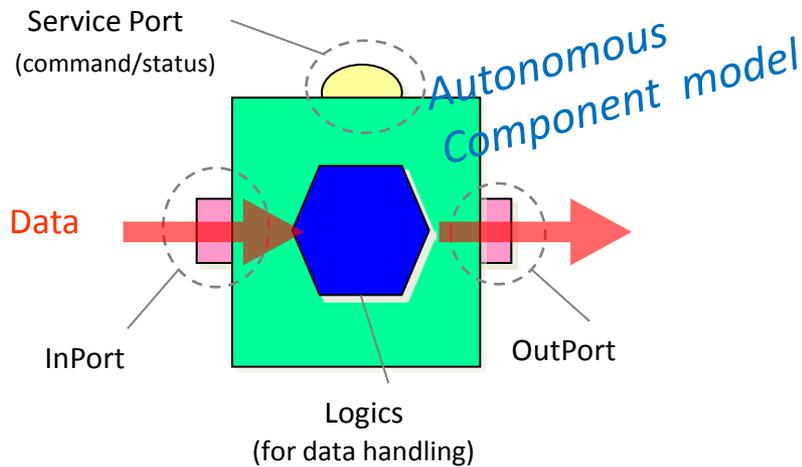


DAQ-Componentは、1台の計算機でもネットワーク分散環境でもシームレスな利用が可能

たとえばDAQシステム(PC)の負荷を分散させたい場合、計算機を追加してDAQ-Componentを移すだけで対応できる

CPUコアが複数ある現在はPC1台のほうがCPUキャッシュを使えて有利な場合もある。

# DAQコンポーネント特徴のまとめ



# データ収集システム

- データ収集システムで必要な事柄
  - データ読み出し、保存
  - 実験中のモニタリング
  - データ収集スタート、ストップ等のランコントロール
  - 周辺機器コントロール

# DAQ-Middleware

## 使用例

# 使用例

- 実験
  - J-PARC/MLF
  - DAQ system of Depth-resolved XMCD (X-ray Magnetic Circular Dichroism) experiments at Photon Factory (KEK IMSS, KEK IPNS)
  - CANDLES
- 実験(これから)
  - J-PARC Hadron E16 (High P)
  - SuperNEMO
- 検出器テストベッド
  - ILC FPCCD Vertex (KEK, 東北大学)
  - GEM (KEK 測定器開発室)
  - SOI (KEK 測定器開発室)
  - ADC\_SiTCP (Open-It)
  - J-PARC Hadron COMET CDC
  - J-PARCニュートリノ 液体アルゴンTPC ボード

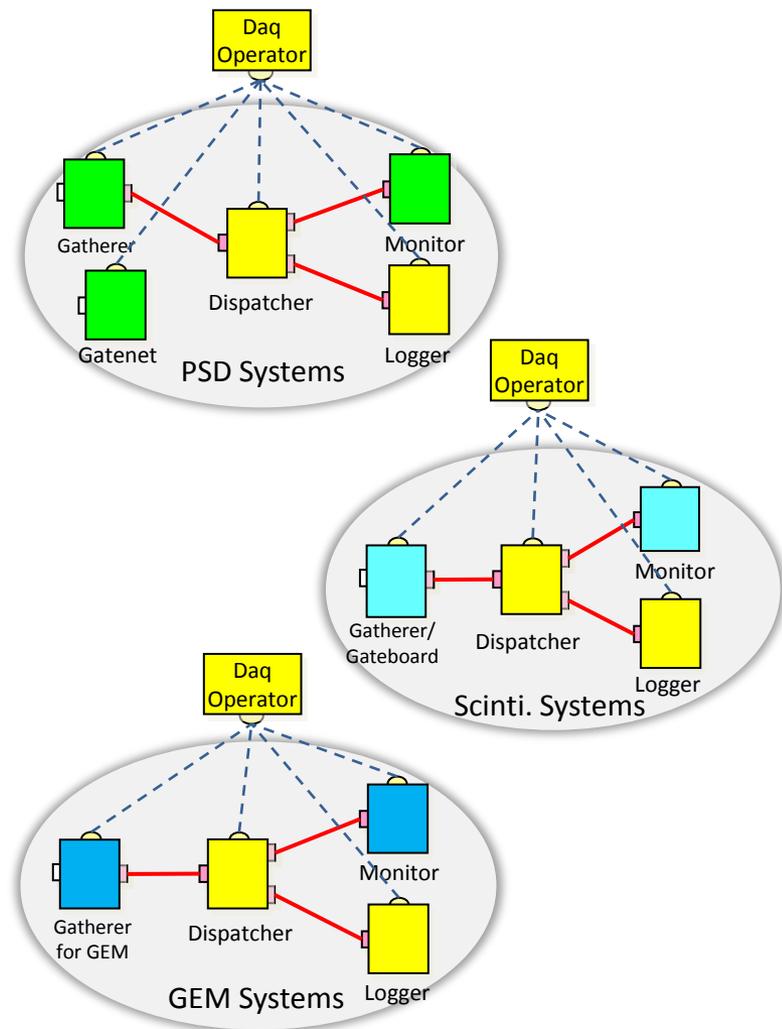
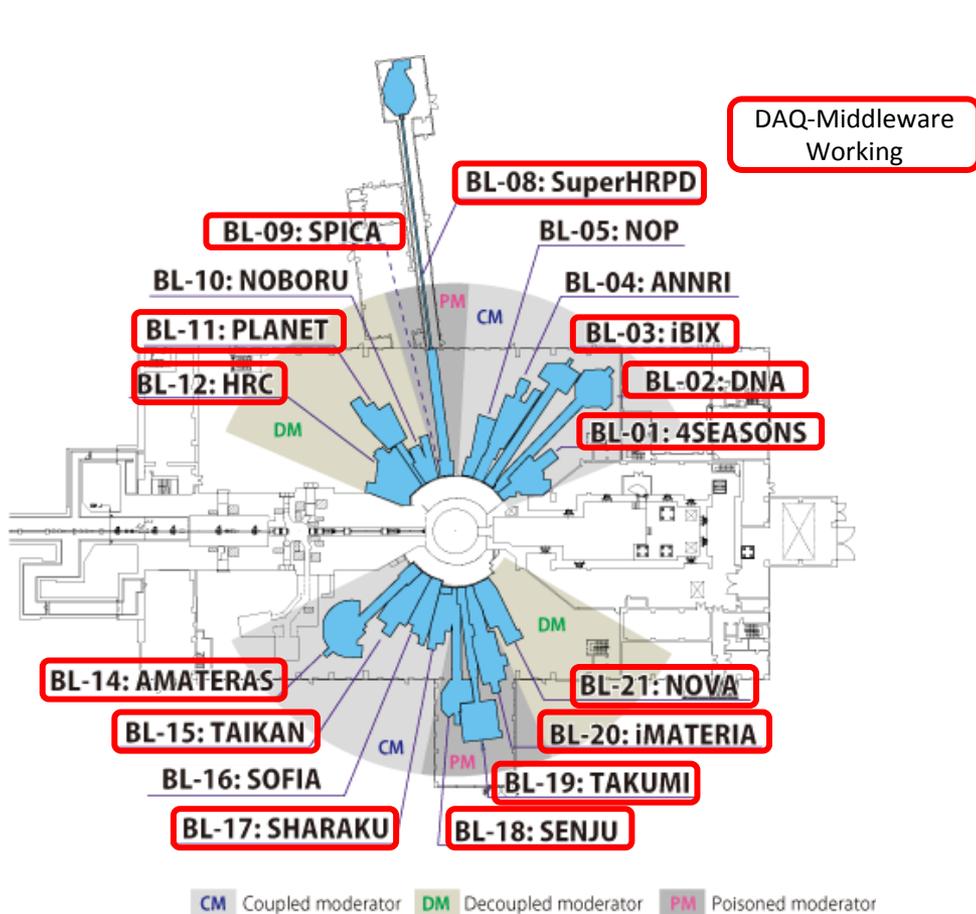
# J-PARC/MLFでの例

## Japan Proton Accelerator Research Complex



高エネルギー加速器研究機構 (KEK)、原子力研究開発機構 (JAEA) 共同運営

# J-PARC MLF中性子での使用状況



# J-PARC/MLF 中性子 検出器・リードアウトモジュール

- Position Sensitive Detector (PSD)
  - $^3\text{He}$  filled proportional counter
  - The most common neutron detector
- Photon-counting 2-D/1-D detector (Scinti)
- Gas Electron Multiplier (GEM)



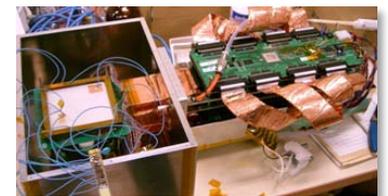
PSDs



2-D Scinti

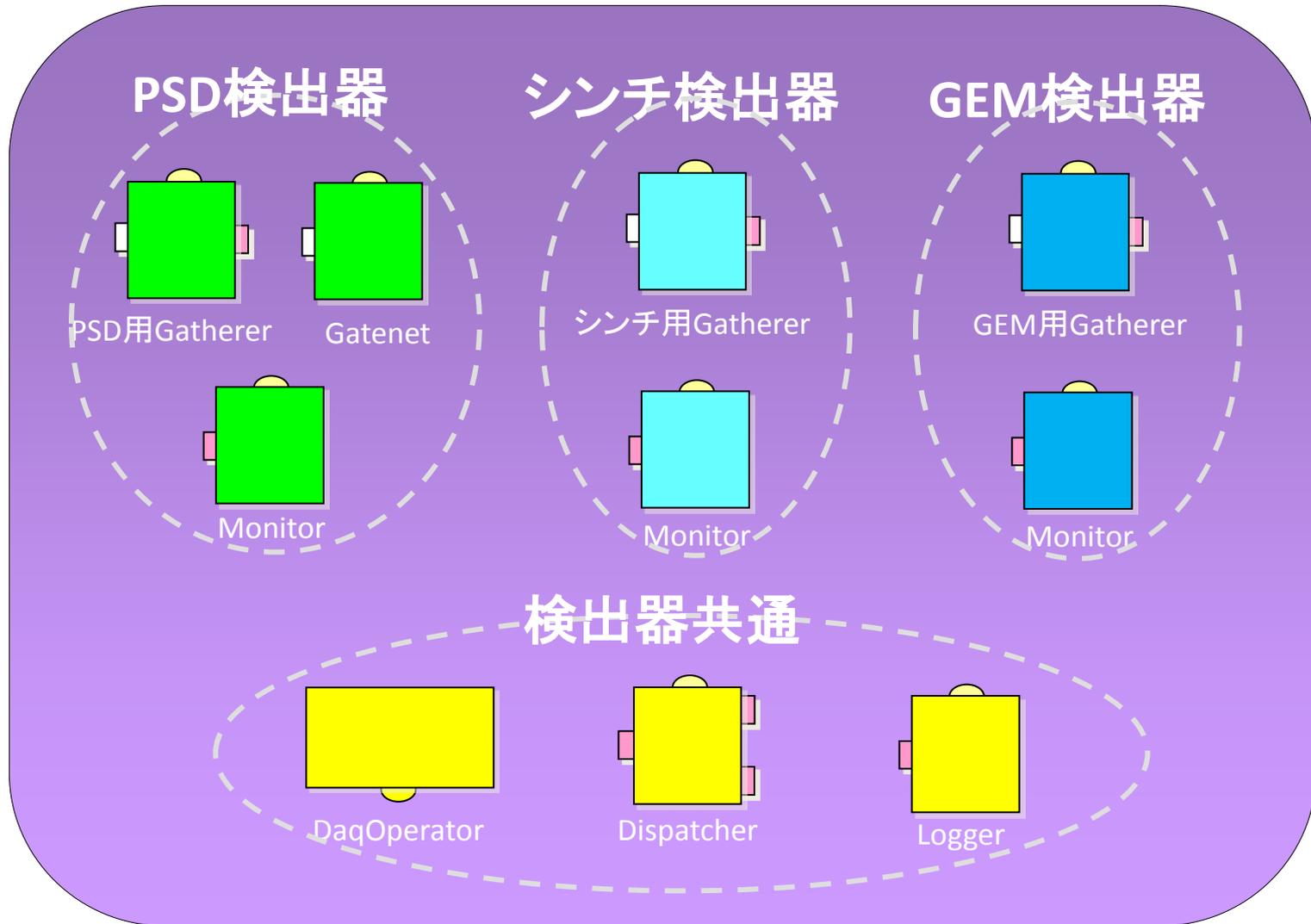


1-D Scinti



GEM

# MLF中性子用DAQコンポーネント群



# DAQ middleware

DAQ middleware is a standard tool for MLF in J-PARC.

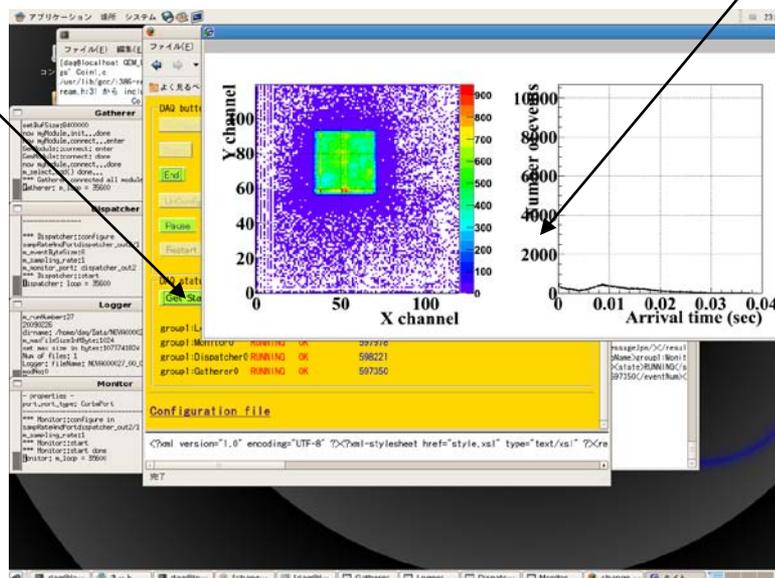
Users are able to take data without regard for the difference of detectors and to control the detectors from a web browser.

DAQ middleware is available as an online monitor.

Control panel in a web browser

The 2D image and the TOF distribution are updated every additional 100 events.

A screen shot during data taking



# ILC CCD Vertexでの状況

順調にデータがとれている。

下は担当の齊藤さんにいただいたスライド

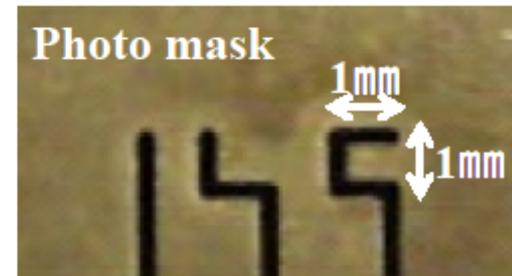
Tomoyuki Saito (Tohoku U.)

14

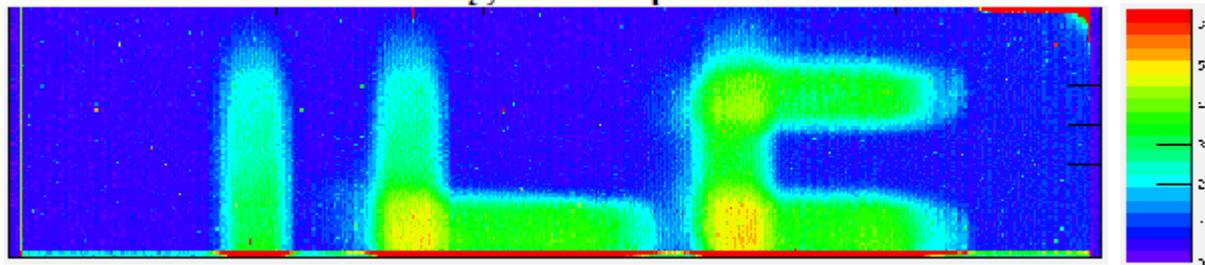
## CCD readout test : Test with LED

CCD is covered with the photo mask and radiated by LED light.

- Photo mask (made of brass)
  - ☞ Character size : 1mm × 1mm
  - ☞ Line width : 0.2 mm



CCD image with photomask



**Success in reading “ILC” image !**

Applications Places System Fri Jun 15, 7:51 AM DAQ User

### histos

Channel	Mean	Max
ch_0	4.186	2.154
ch_1	4.186	2.154
ch_2	3.186	2.154
ch_3	3.186	2.154
ch_4	2.186	2.154
ch_5	2.186	2.154
ch_6	2.186	2.154
ch_7	2.186	2.154
ch_8	2.186	2.154
ch_9	2.186	2.154
ch_10	2.186	2.154
ch_11	2.186	2.154
ch_12	2.186	2.154
ch_13	2.186	2.154
ch_14	2.186	2.154
ch_15	2.186	2.154

```

wp2:~/SiTCP_ADC
wp2.kek.jp.log Templates Videos
ME
data.bin
ster_set
conf
pAdcCoyer
p-adc-emulator
pAdcLogger
pAdcMonitor
pAdcReader
padc_reader_logger.xml

config.xml
d for XML schema
omp for DAQ-Operator

```

```

TCP_ADC/SitcpAdcMonitor

```

```

wp2:~/SiTCP_ADC
daq/SiTCP_ADC/SitcpAdcCoyer'
l'.
aq/SiTCP_ADC/SitcpAdcCoyer'

```

```

ME
data.bin
ster_set
conf
pAdcCoyer
p-adc-emulator
pAdcLogger
pAdcMonitor
pAdcReader
padc_reader_logger.xml

config.xml

```

```

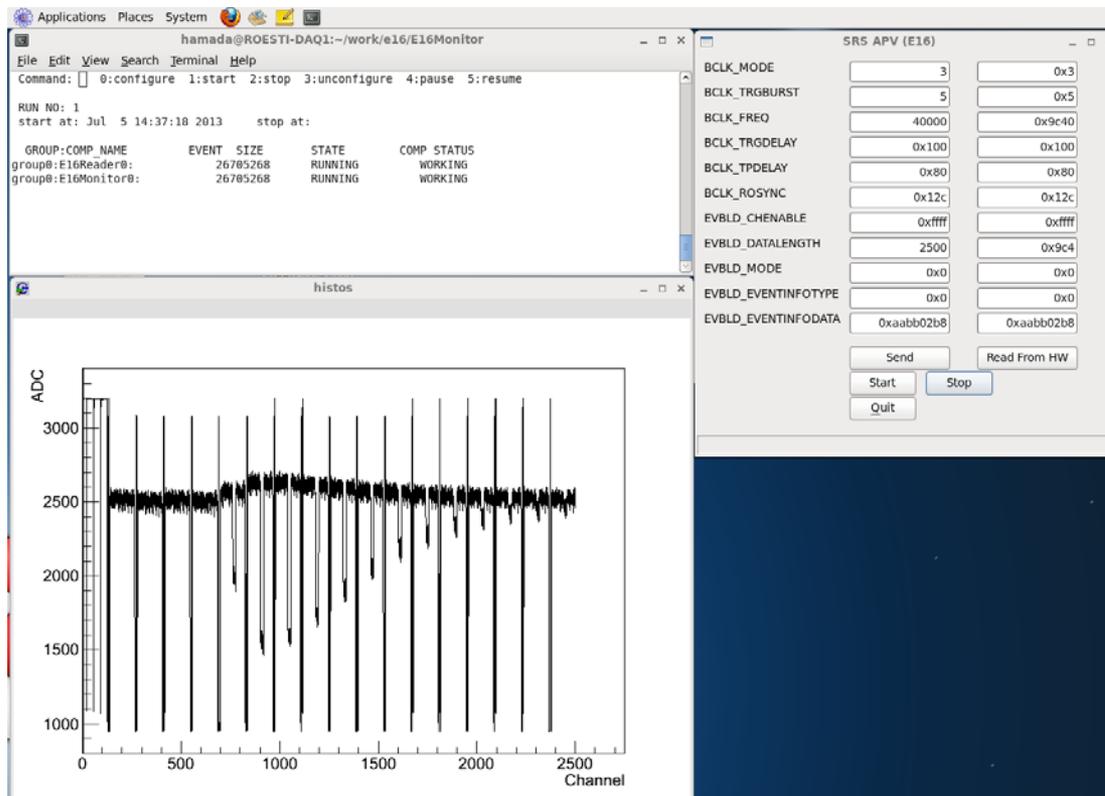
Use config file config.xml
Use /usr/share/daqmw/conf/config.xsd for XML schema
Use /usr/libexec/daqmw/DaqOperatorComp for DAQ-Operator
Conf file validated: config.xml
start new naming service... done
Local Comps booting... done
Now booting the DAQ-Operator... done
[daq@daqmwpc2 SiTCP_ADC]$

```

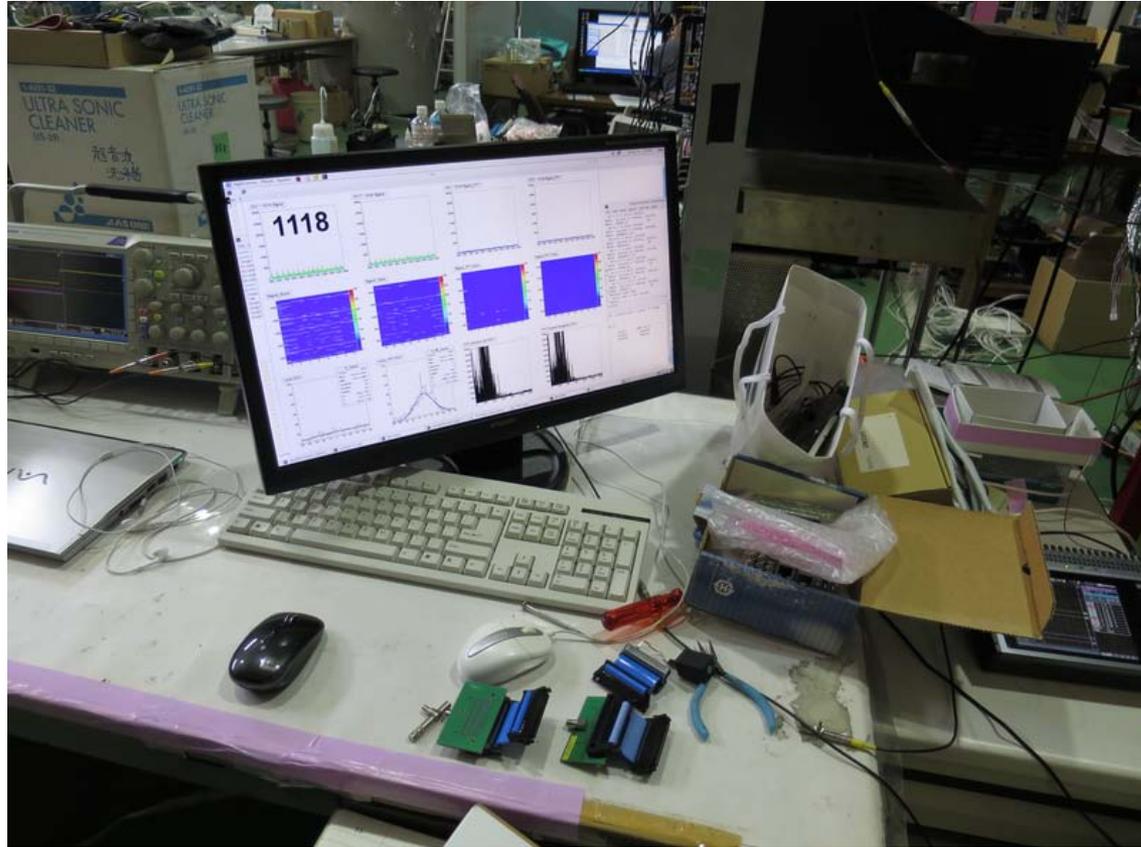
daq@daqmwpc2:~/SiT...    daq@daqmwpc2:~/SiT...    daq@daqmwpc2:~/SiT...

# J-PARC Hadron E16 (High Pt)

## DAQ-Middlewareテスト



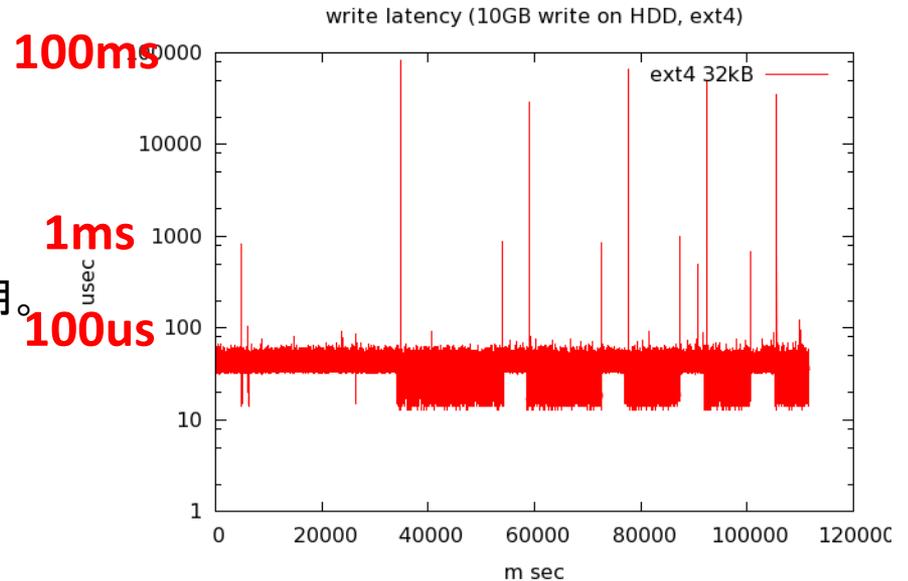
# J-PARC/ハドロン液体アルゴンTPC



# ディスクへの書き込み (1)

ハードディスク書き込み時間の測定  
 DAQ-Middlewareではなく普通のCで書いたプログラムで測定

- 最大スループット120MB/s のハードディスクを使用。
- 32kB書いて360us sleepするワークロードを定義。
- スループット88MB/sで10GB分書く。
- おのおの32kB書くのに要した時間を測定  
 (10GB/32kB = 327680回)









# ディスクへの書き込み (2)

## ハードディスク書き込み時間の測定

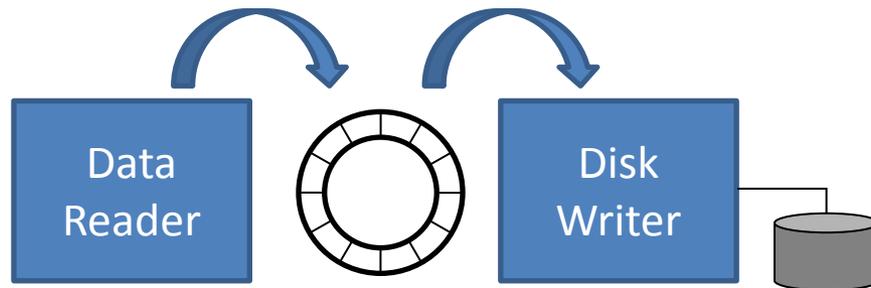
DAQ-Middlewareではなく普通のCで書いたプログラムで測定 **100ms**

- 32kB書いて360us sleepするワークロードを定義。
- 最大スループット120MB/s のハードディスクを使用。
- スループット88MB/sで10GB分書く。
- おおの32kB書くのに要した時間を測定 (10GB/32kB = 327680回)

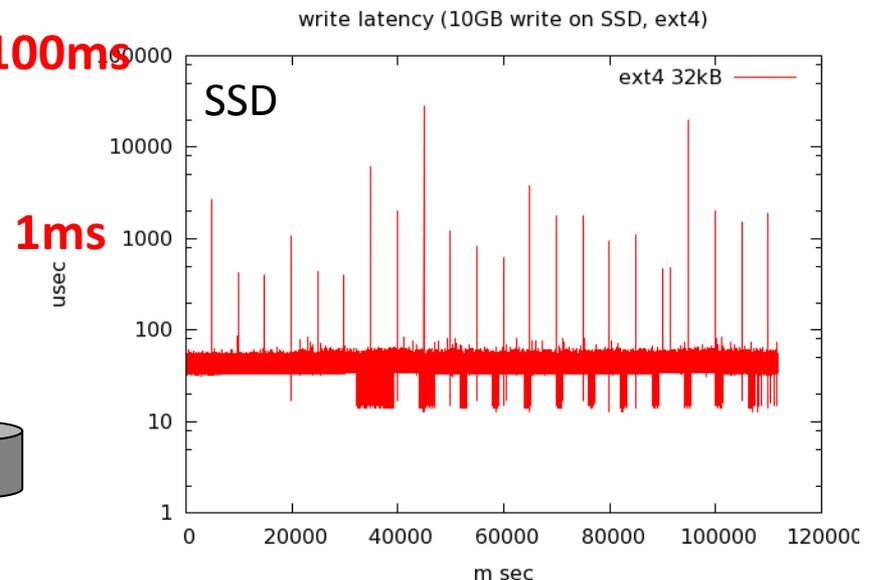
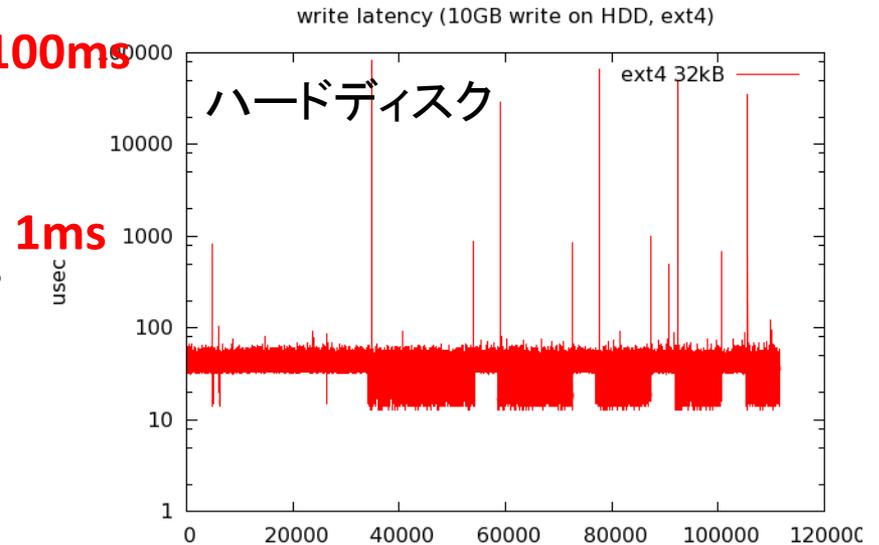
データを読む

データを書く

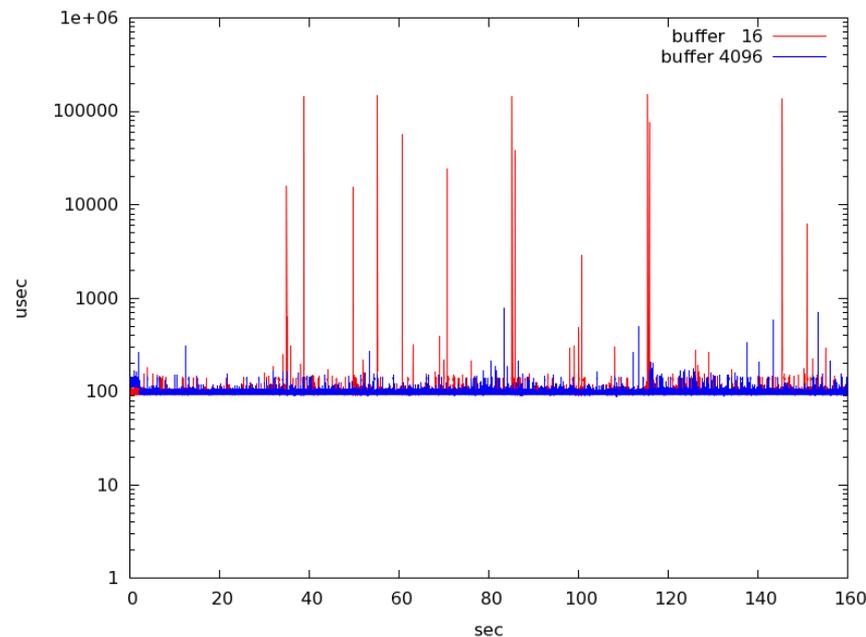
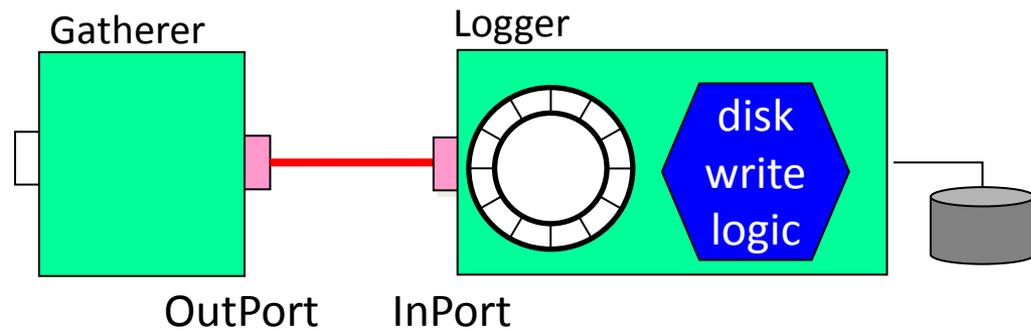
をシーケンシャルに実行するとデータを書くところで **100ms** 動作が止まることがあるのでバッファを設け、読み書きが独立に動作する必要がある。



2016-10-19



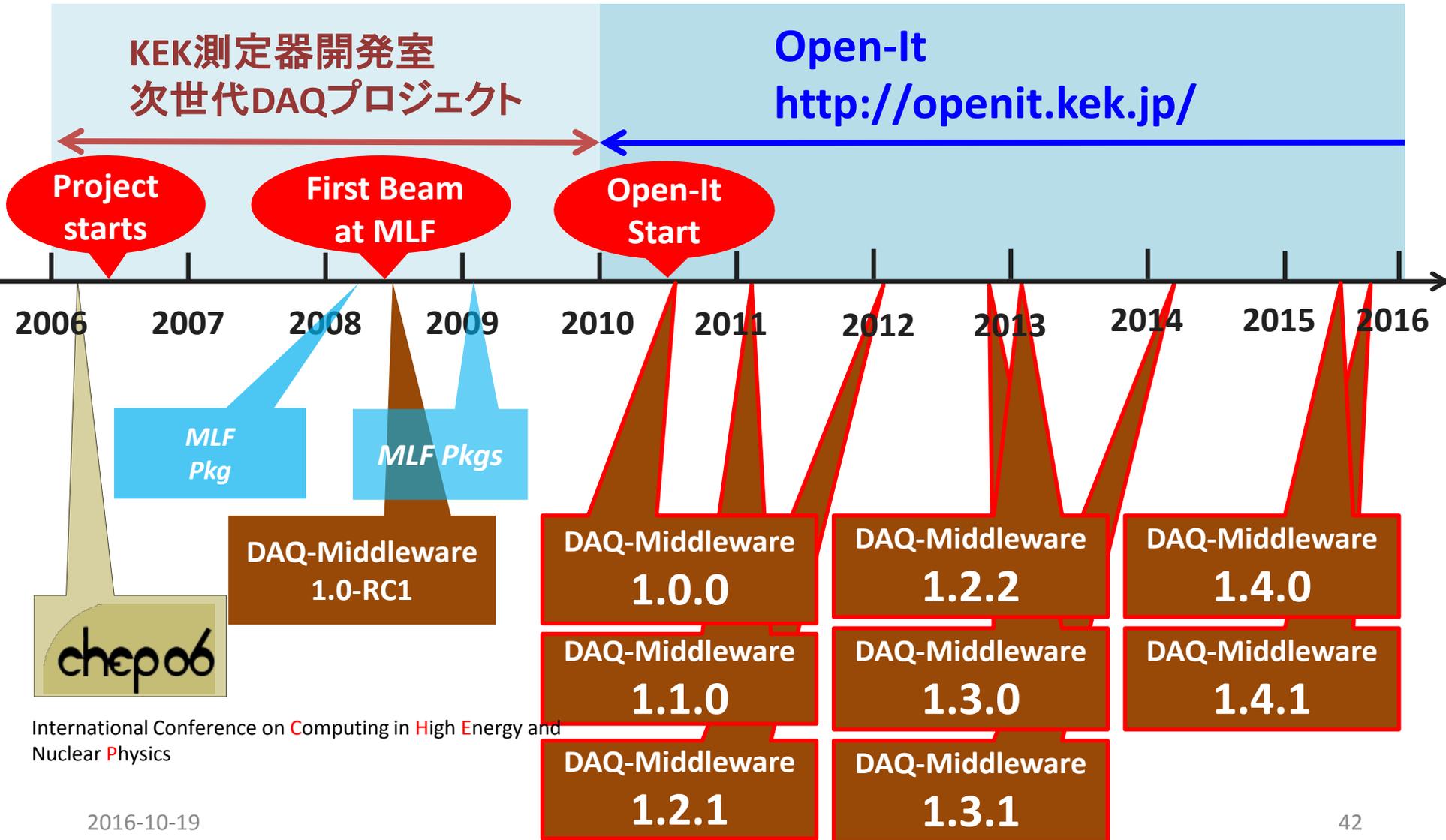
# ディスクへの書き込み (3)



- すでにバッファが用意されており遅延を吸収することが可能

# 開発体制など

# DAQ-Middlewareの歴史



# 1.0.0～1.4.0

- 1.0.0 MLF中性子依存のものを排除した最初のバージョン
- 1.1.0 Scientific Linux x86\_64 (64bit)サポート
- 1.2.0 Scientific Linux 6.xサポート (ソースコードではUbuntu、Debianをサポートしている)
- 1.2.1 - 1.2.2 Bug Fixes, Features
- 1.3.0 Buffering、Configuration GUI
- 1.4.0 Scientific Linux 7.xサポート

# 開発体制

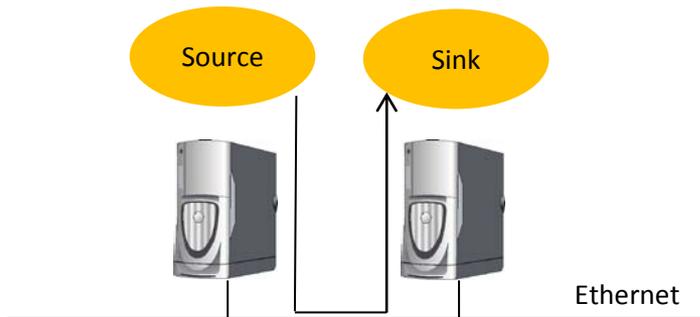
- 2010年4月 DAQ-Middleware Core グループ結成
- メンバー
  - 千代、濱田、井上 (KEK)
  - 長坂 (広島工業大学)
  - 味村 (大阪大学)
  - 神徳、安藤 (産業技術総合研究所)
  - 和田 ( (株) Bee Beans Technologies)
  - 仲吉(KEK, 2011年4月まで)、安(KEK, 2012年3月まで)、井上(KEK, 2015年3月まで)

# 学会発表、展示会等

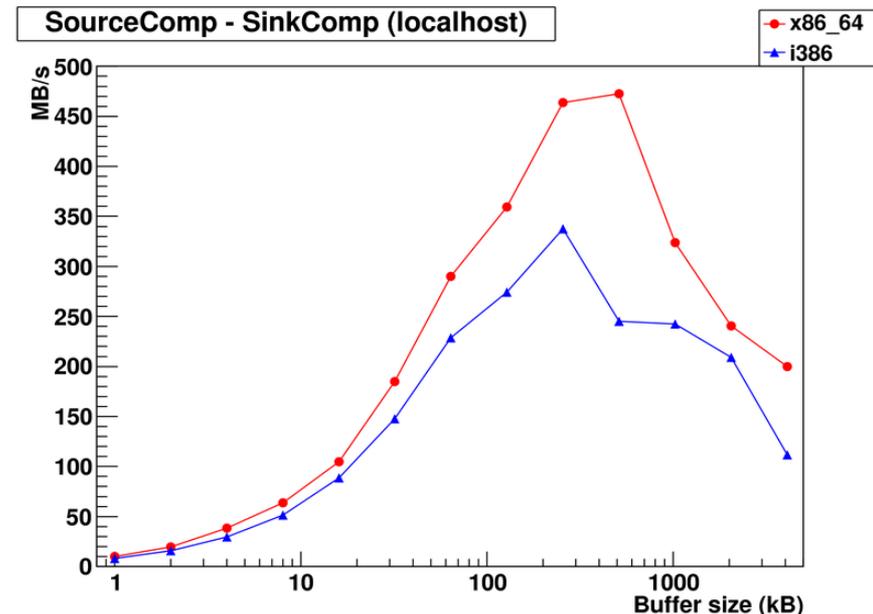
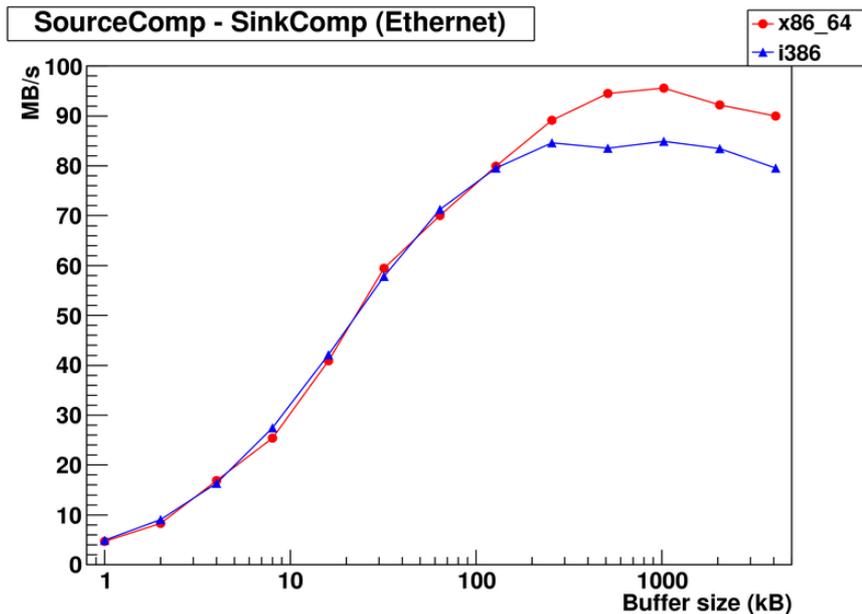
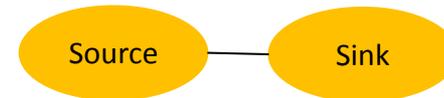
- CHEP (International Conference on Computing in High Energy and Nuclear Physics)
- Real Time Conference
- 物理学会
- 中性子科学会
- TXテクノロジーショーケース in つくば
- イノベーションジャパン
- 産総研オープンラボ

# 性能測定

# 転送速度テスト



Model: HP 8600 xw  
 CPU: Intel Xeon E5420 2.50GHz  
 NIC: Broadcom NetXtreme BCM5755  
 Scientific Linux 5.7 i386, x86\_64



# 開発に必要なプログラミング技能

- 言語: C++
  - 解説書例
    - 技術職員専門課程研修(平成22年度)データ処理のためのC++入門  
<http://www-lib.kek.jp/tiff/2010/1026/1026005.pdf> あるいは  
<http://daqmw.kek.jp/docs/fujii-cpp.pdf>
- リードアウトモジュールからのデータ読み出し
  - SiTCPリードアウトモジュールからのデータ読みだしならネットワークプログラム
  - ネットワークプログラムについては単純なものならDAQ-Middleware配布物としてライブラリがある
- モニターで使用するヒストグラムツール(ROOTなど)

# 開発環境

- DAQ-Middleware開発者提供のVMware Playerイメージを使う
- 自力
  - Scientific Linux (5.x, 6.x 32bit, 64bit, 7x 64bit) (CentOS 5.x, RHEL 5.x, 6.x, 7)用バイナリパッケージ、Ubuntu (2012.04 LTS 32bit, 64bit)バイナリパッケージがあるのでこれを使ってセットアップ
    - SL
      - `wget http://daqmw.kek.jp/src/daqmw-rpm`
      - `sh daqmw-rpm install`
    - Ubuntu
      - `wget http://daqmw.kek.jp/src/Ubuntu_daqmw`
      - `sh Ubuntu_daqmw install`
  - それ以外のOSにはソースからセットアップ
    - 依存物(OpenRTM-aist、omniORB)があるのでちょっとめんどくさい。

# ドキュメンテーション

- DAQ-Middleware 1.1.0 技術解説書  
(1.4.1でも有効)

<http://daqmw.kek.jp/docs/DAQ-Middleware-1.1.0-Tech.pdf>

- DAQ-Middleware 1.4.0開発マニュアル

<http://daqmw.kek.jp/docs/DAQ-Middleware-1.4.0-DevManual.pdf>

# ホームページ、サポート

- <http://daqmw.kek.jp/>
- サポートメールアドレス  
`daqmw-support@ml.post.kek.jp`