

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

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

背景、構想

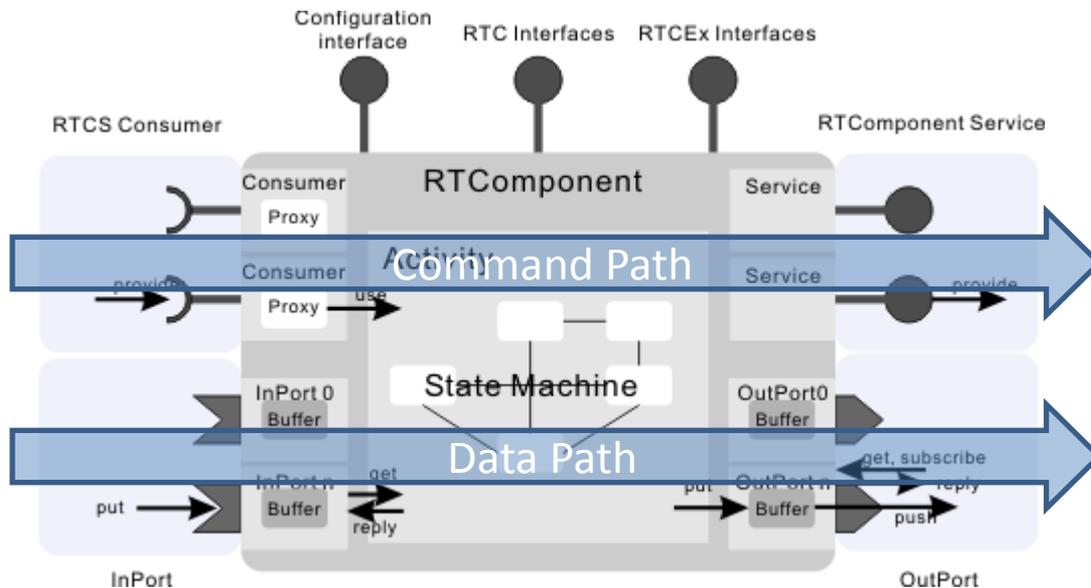
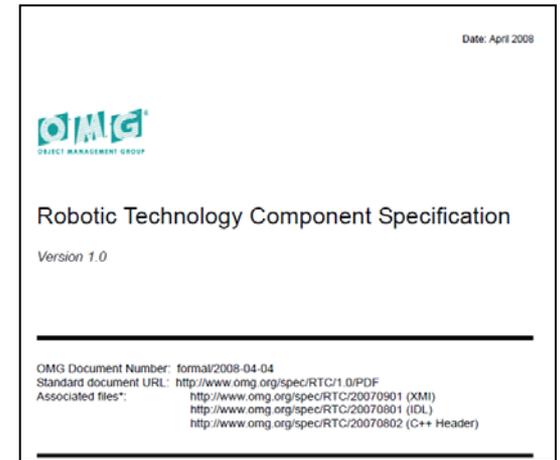
- 背景
 - 従来DAQシステムでのソフトウェアの再利用化はドライバ、ライブラリレベルでおこなわれてきた。
 - 扱うデータの増大、使う計算機の数が増えてきてDAQシステムを構築するのがむずかしくなってきた。
 - どんな実験にも対応できるように、抽象化、汎用化してしまうとデータ収集効率が落ちてしまう。
- 解決案
 - ドライバ、ライブラリとDAQシステムの間コンポーネントという中間層を作り、実験毎の違いを吸収、収集効率を確保し、
 - システムの枠組みは普遍であるDAQフレームワークを作ればよいのではないか？

DAQ-Middlewareとは (1)

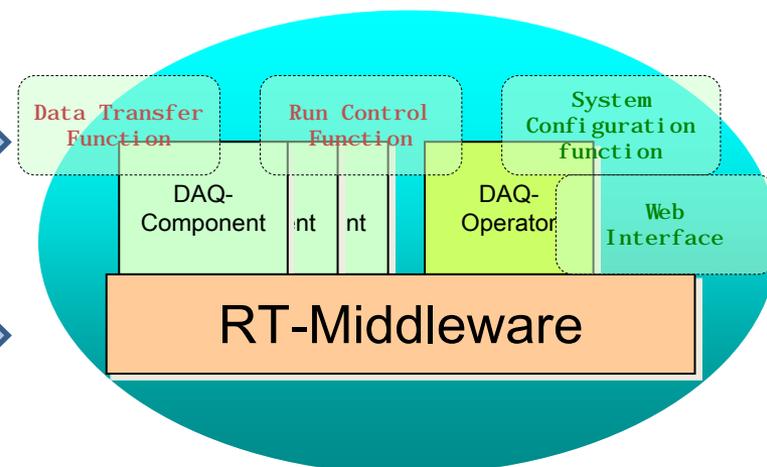
- 再利用が容易な、柔軟性がある汎用のネットワークベースデータ収集(DAQ)ソフトウェアフレームワーク
- ターゲット
 - 中小規模実験
 - テストベッド (測定器、エレクトロニクス等)

DAQ-Middleware (2)

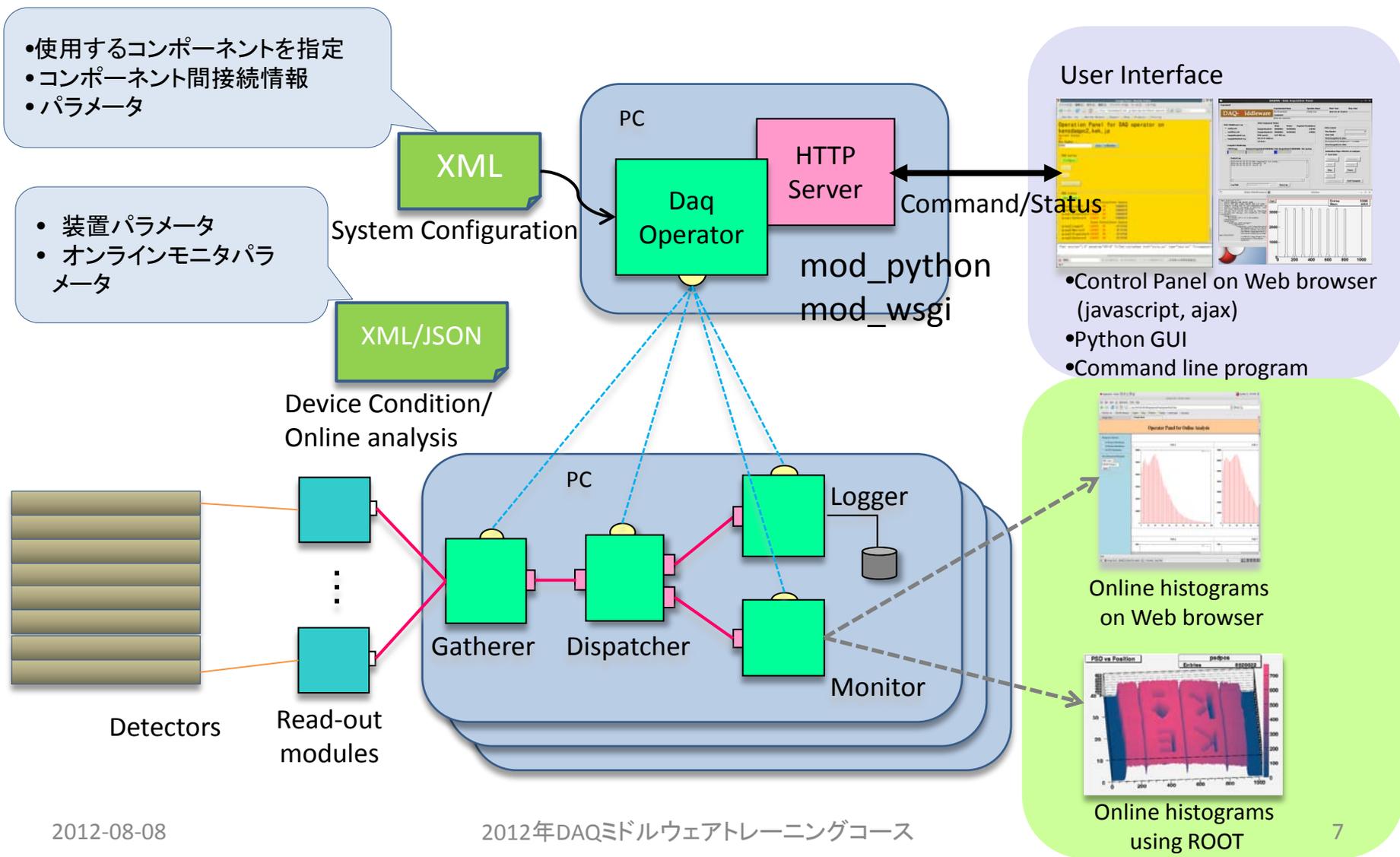
- RT(Robot Technology)-Middlewareをデータ収集用に拡張
- RT-Middleware
 - ネットワークロボットシステムの構築のためのソフトウェア共通プラットフォーム
 - 産総研知能システム研究部門・タスクインテリジェンス研究グループが開発
 - 複数のコンポーネントが通信してひとつの機能を実現する
 - そのソフトウェアコンポーネントの仕様は国際標準規格(OMG)
 - 我々は2006年から産総研と共同研究を行っている

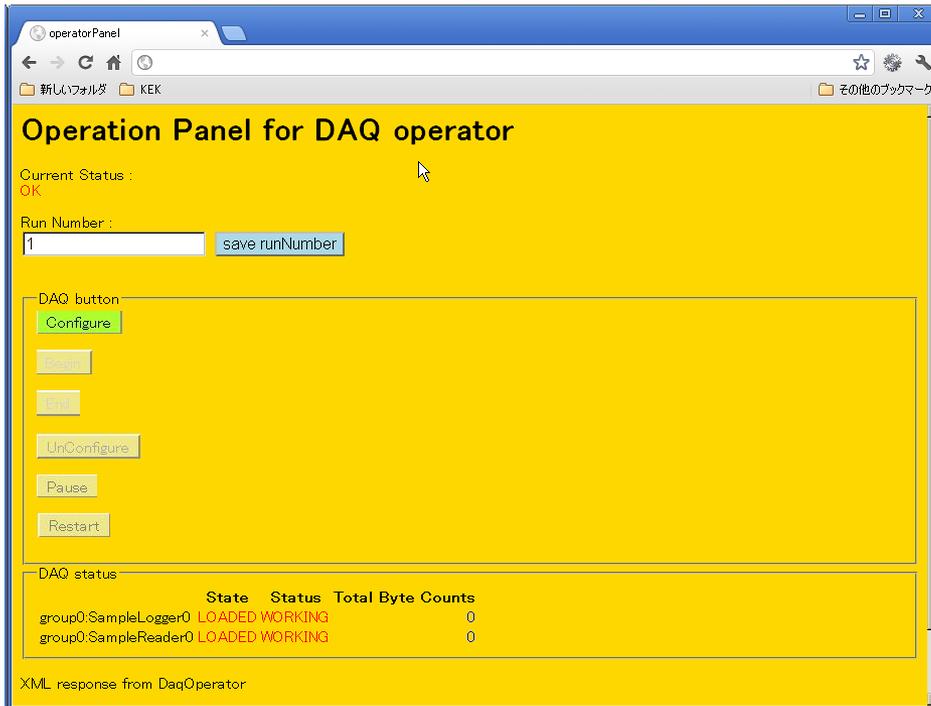


DAQ-Middleware

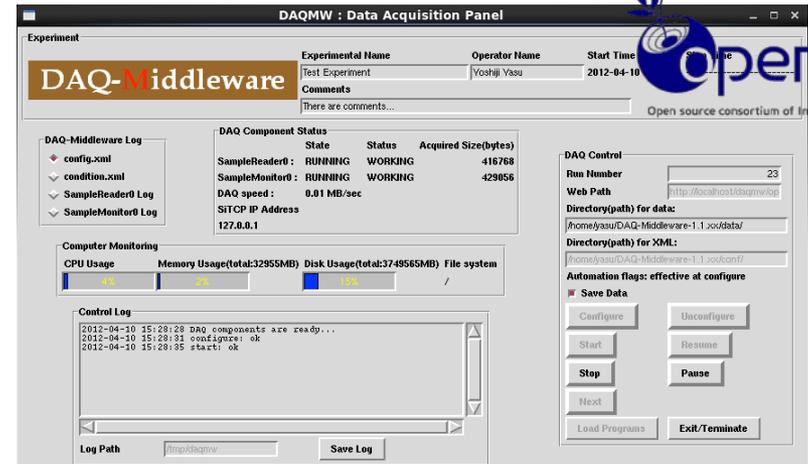


DAQ-Middleware構成図

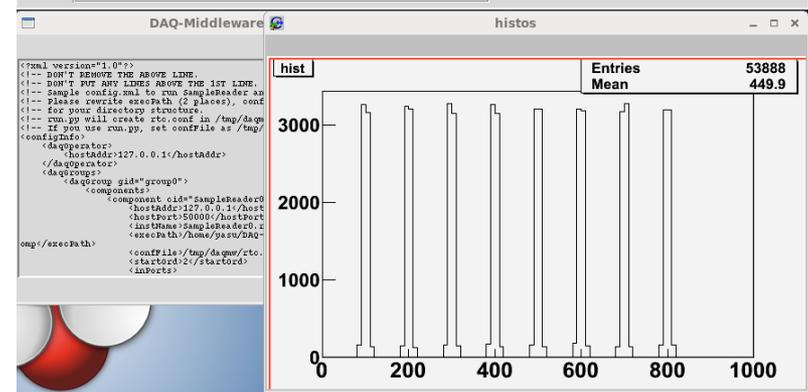




State	Status	Total	Byte Counts
group0.SampleLogger0	LOADED WORKING		0
group0.SampleReader0	LOADED WORKING		0



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



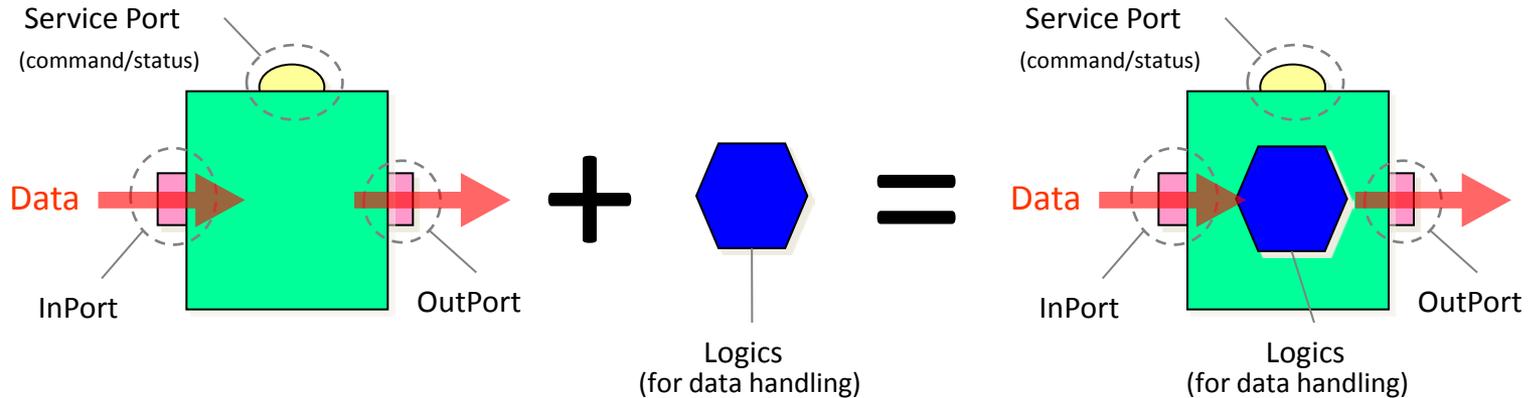
Entries	Mean
53888	449.9

```
% 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></className><name></name><methodName></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></className><name></name><methodName></methodName><messageEng></messageJpn></result></returnValue></response>
% █
```

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

DAQコンポーネント

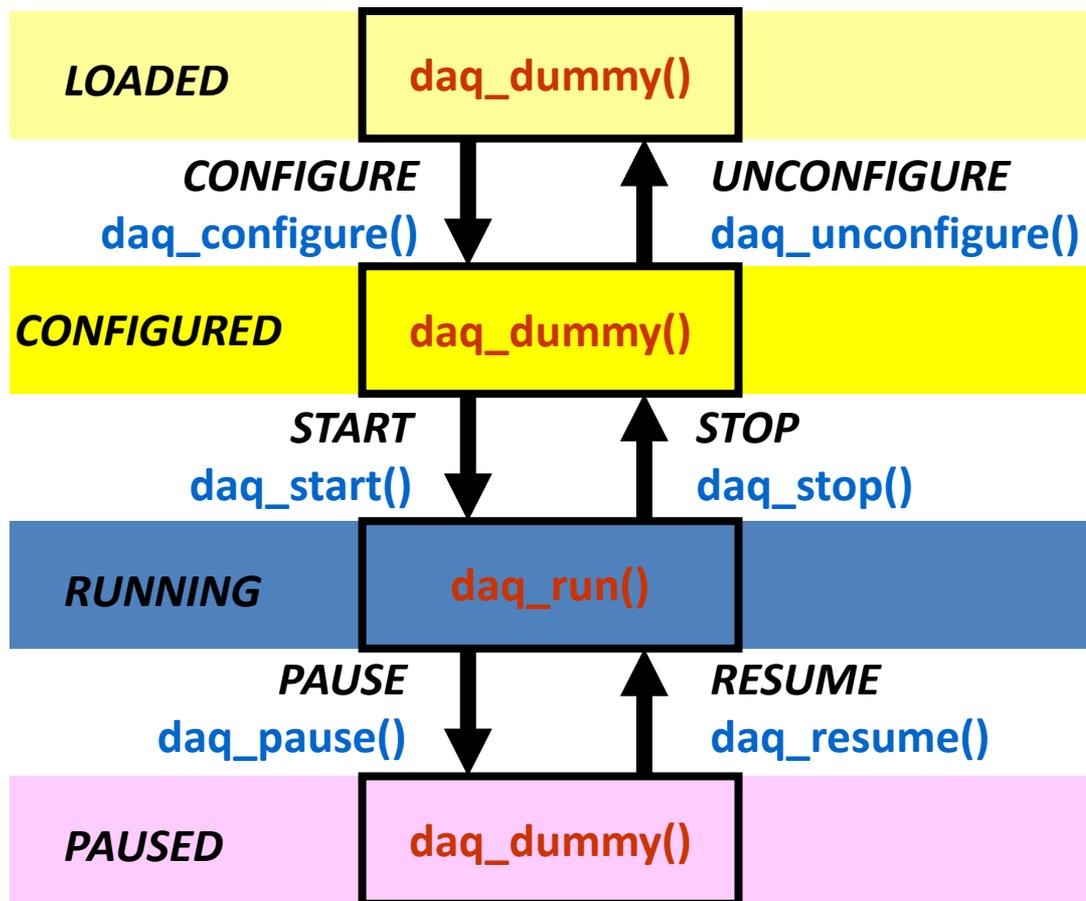


- DAQコンポーネントを組み合わせてDAQシステムを構築する
- データ転送機能、ランコントロール、システムコンフィギュレーション機能はDAQ-Middlewareで実装済み。
- データを下流に送るにはOutPortに書く。
- 上流からのデータを読むにはInPortを読む。
- ユーザーはコアロジックを実装することで新しいコンポーネントを作成できる。

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)

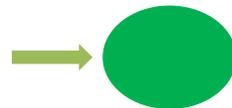
各関数を実装することでDAQコンポーネントを完成させる。

コンポーネント間通信での分類

Source Type



Sink Type



Filter Type



Dispather Type

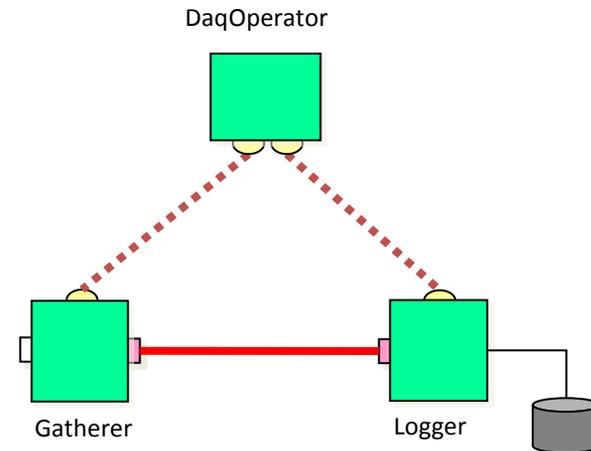
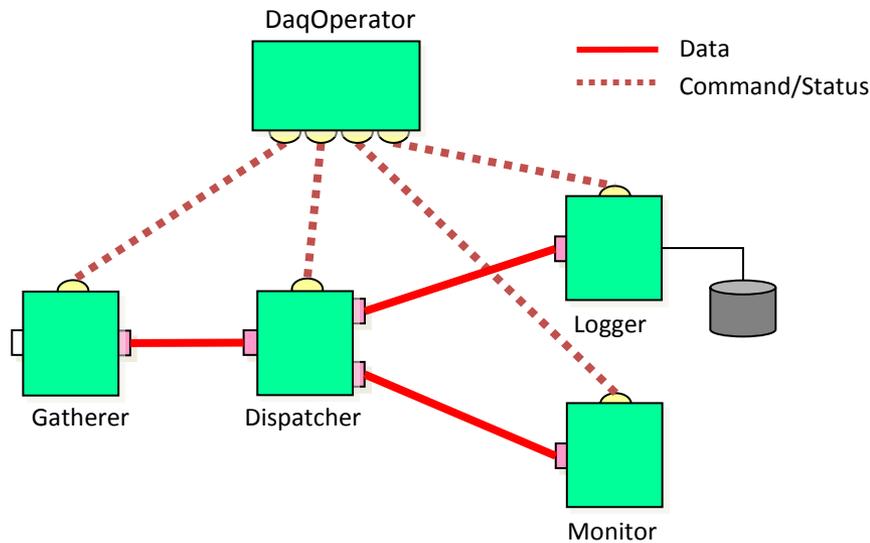


Merger Type

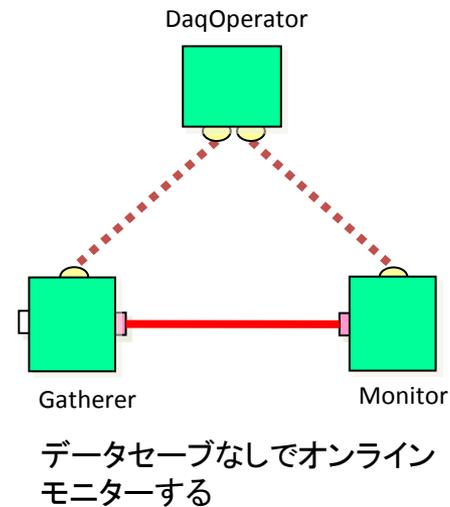
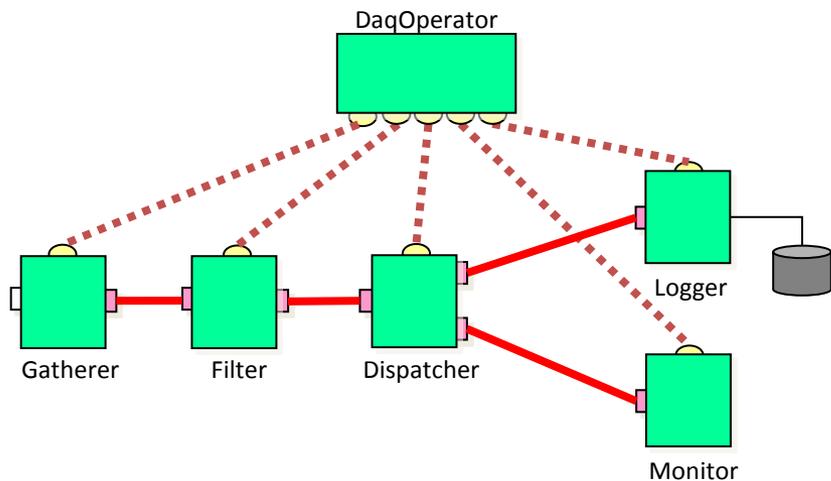


- Source Type (Gatherer)
- Sink Type (Logger, Monitor)
- Dispatcher Type

DAQコンポーネント 構成例(1)

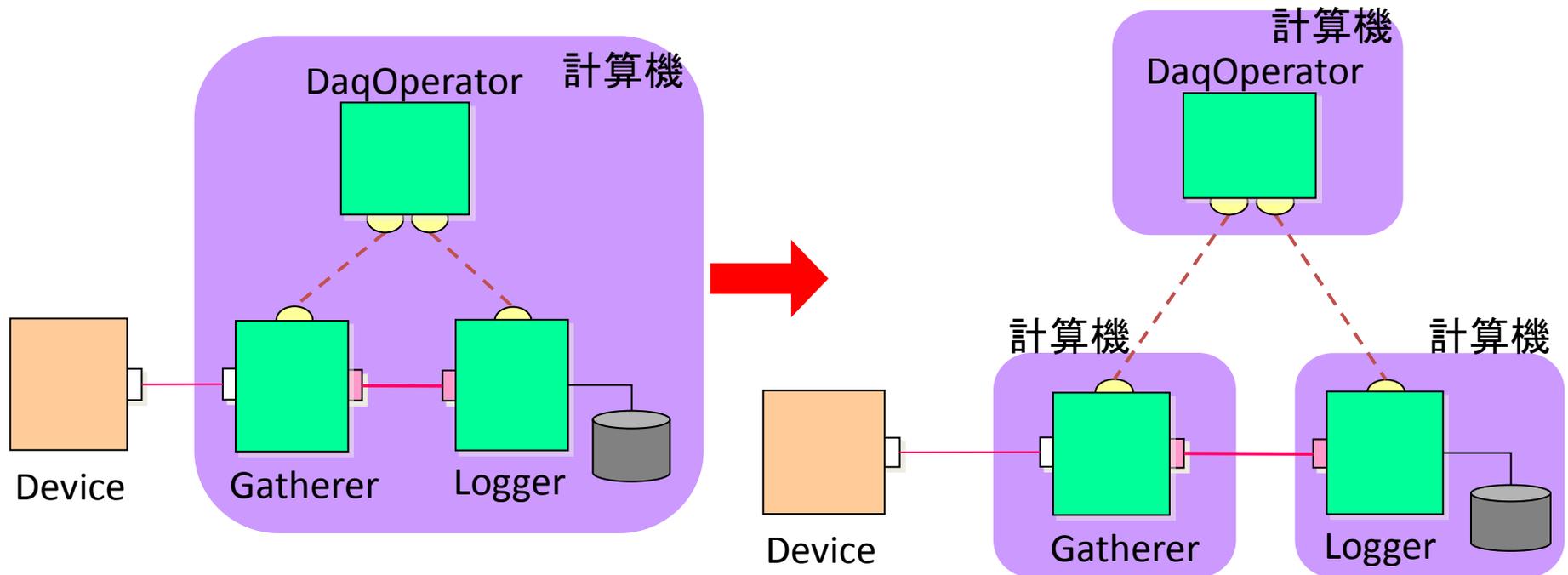


モニターなしでデータをディスクにセーブする



DAQコンポーネント構成例 (2)

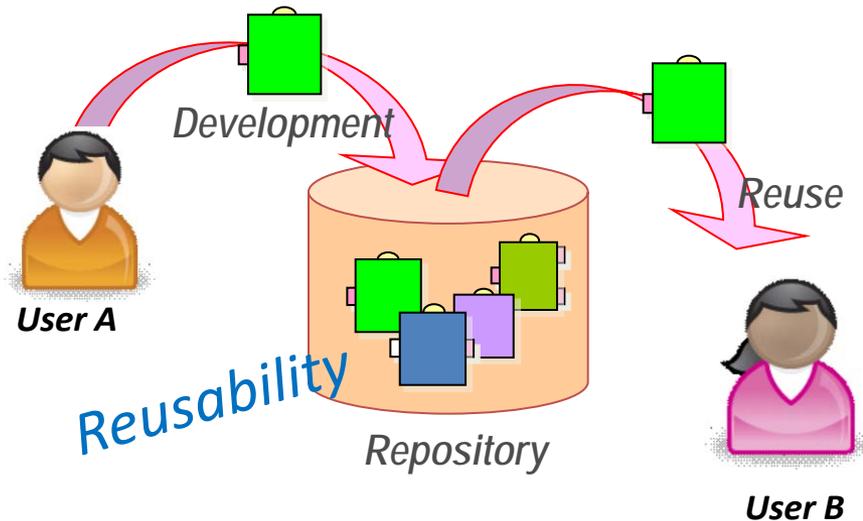
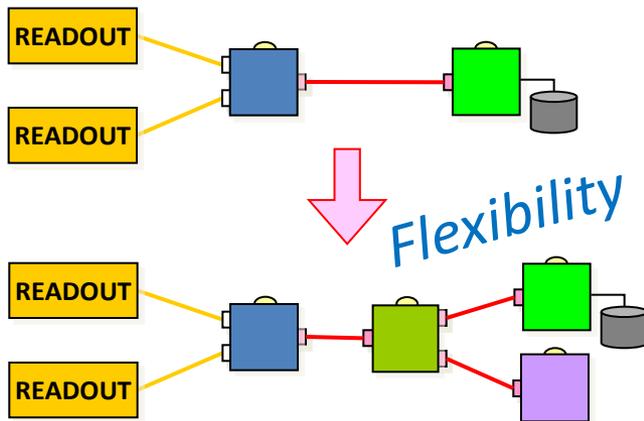
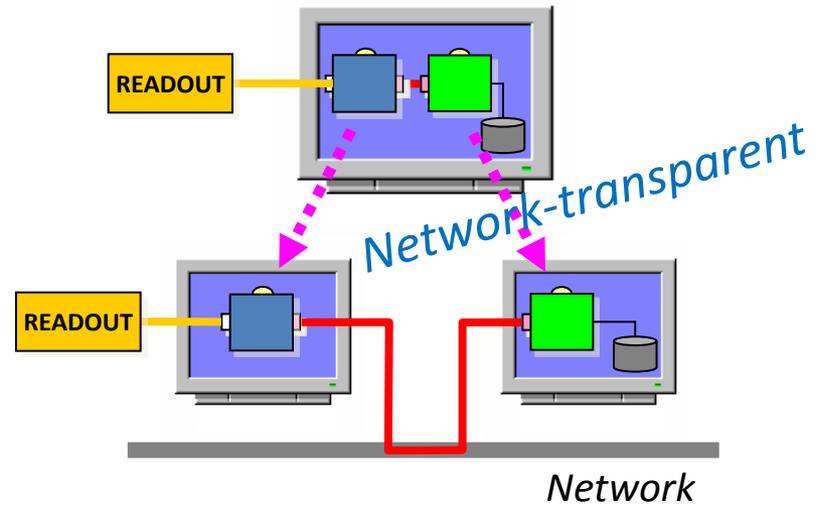
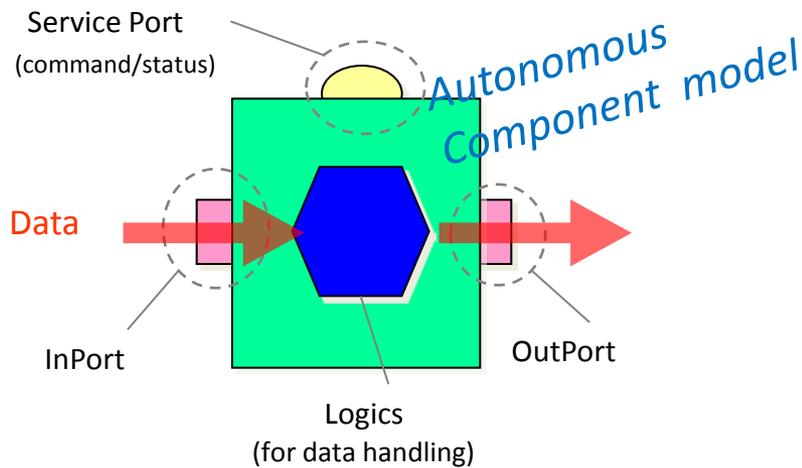
ネットワーク透過性



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

DAQコンポーネント特徴のまとめ



データ収集システム

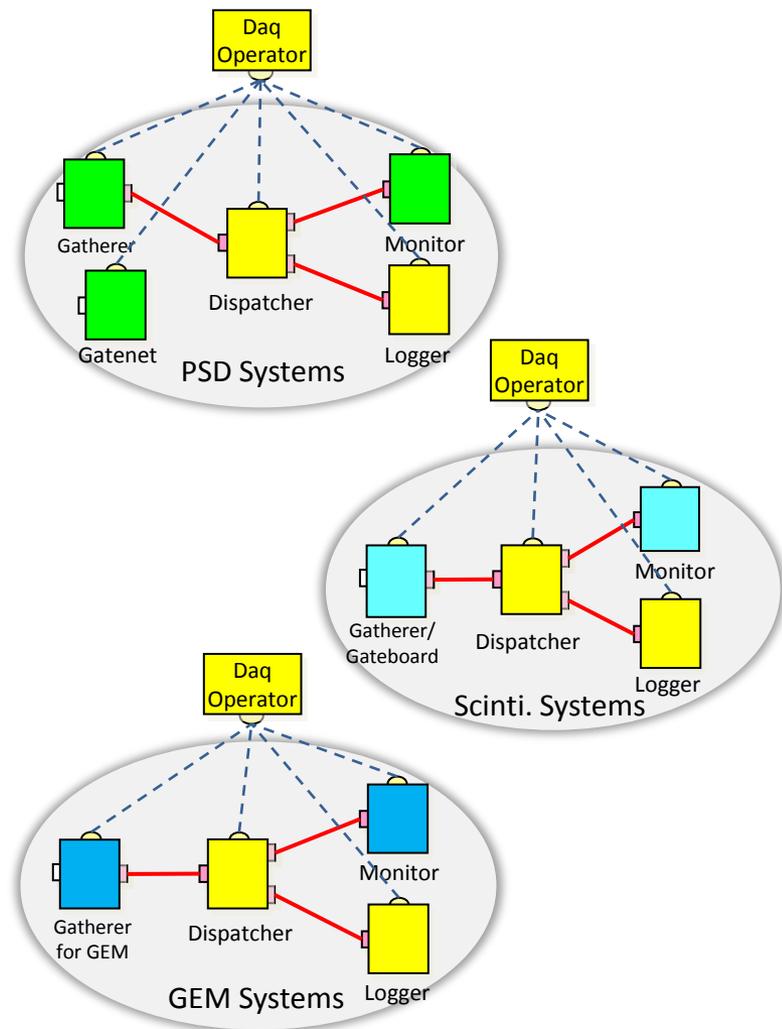
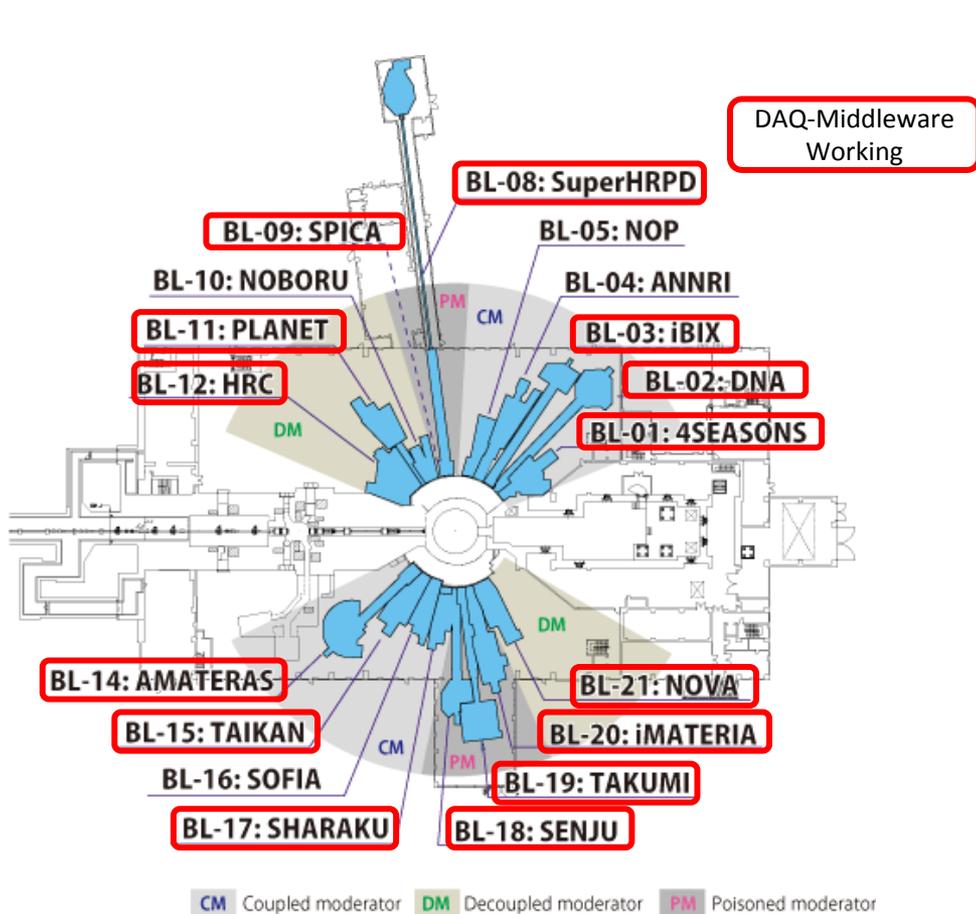
- データ収集システムで必要な事柄
 - データ読み出し、保存
 - 実験中のモニタリング
 - データ収集スタート、ストップ等のランコントロール
 - 周辺機器コントロール

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)
- 実験(検討中)
 - CANDLES
 - SuperNEMO
- 検出器テストベッド
 - ILC CCD Vertex (KEK, 東北大学)
 - GEM (KEK 測定器開発室)
 - SOI (KEK 測定器開発室)

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

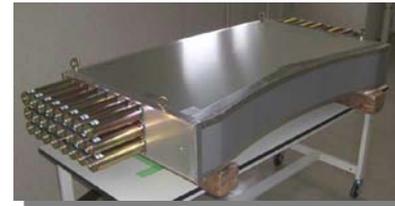
- Position Sensitive Detector (PSD)
 - ^3He filled proportional counter
 - The most common neutron detector
- Photon-counting 2-D/1-D detector (Scinti)
- Gas Electron Multiplier (GEM)



PSDs



2-D Scinti

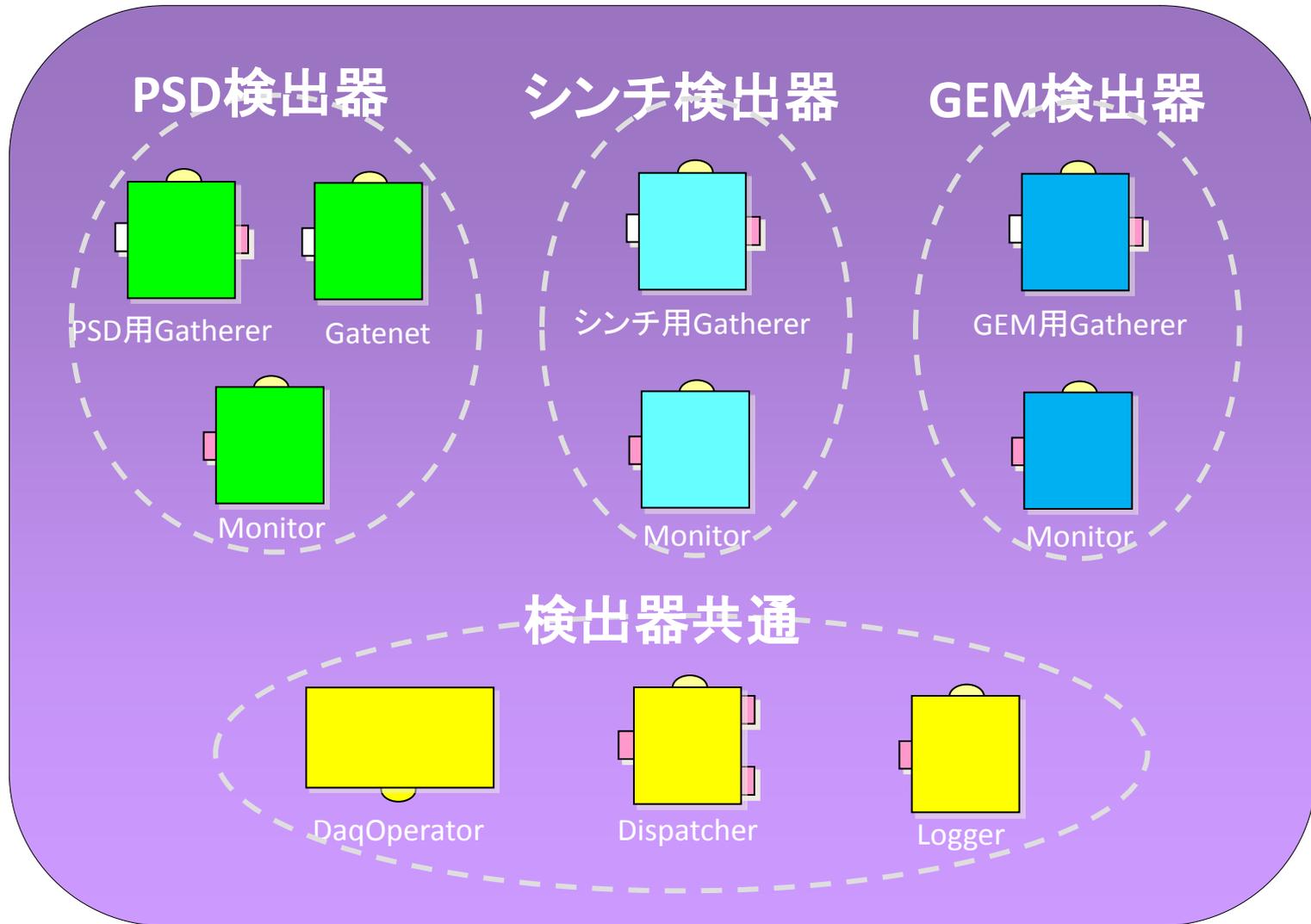


1-D Scinti



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.

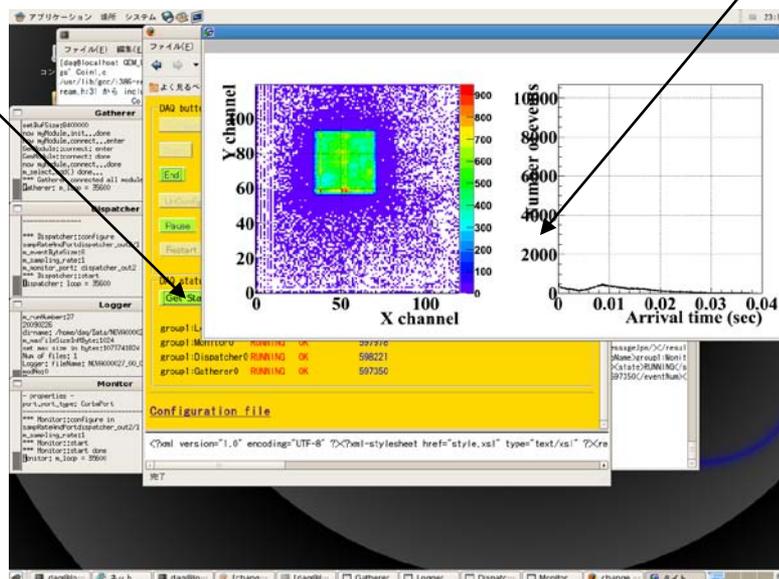
Users are able to take data without regard for the difference of detectors and to control the detectors from a web browser.

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

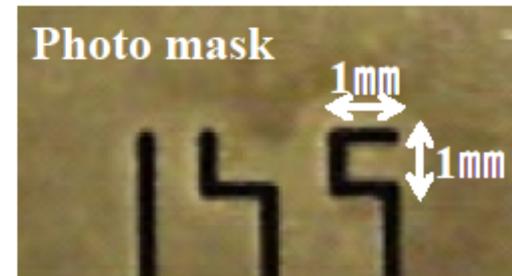
Tomoyuki Saito (Tohoku U.)

14

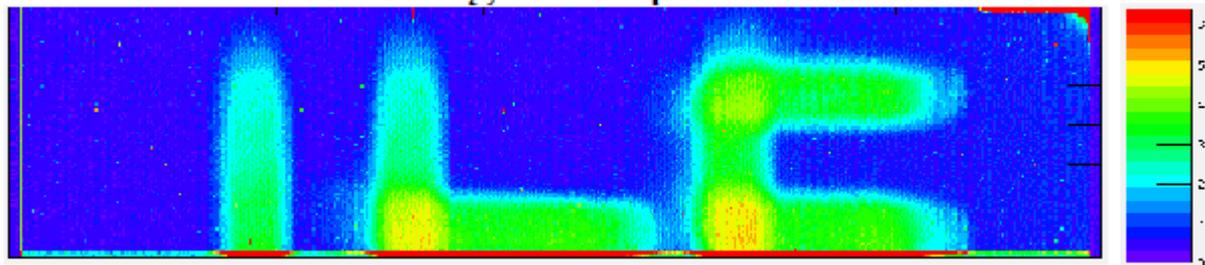
CCD readout test : Test with LED

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

The 'histos' window displays 16 histograms arranged in a 4x4 grid, labeled ch_0 through ch_15. Each histogram shows a distribution of data points with a peak around zero. The y-axis is logarithmic, ranging from 1 to 100. The x-axis ranges from -200 to 200. Each histogram includes a small box with statistics: Mean, StdDev, and Max.

The terminal window shows the following output:

```

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]$

```

開発体制など

1.0.0～1.2.0

- 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)

開発体制

- 2010年4月 DAQ-Middleware Core グループ
結成
- メンバー
 - 千代、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(2011年4月まで)、安(2012年3月まで)

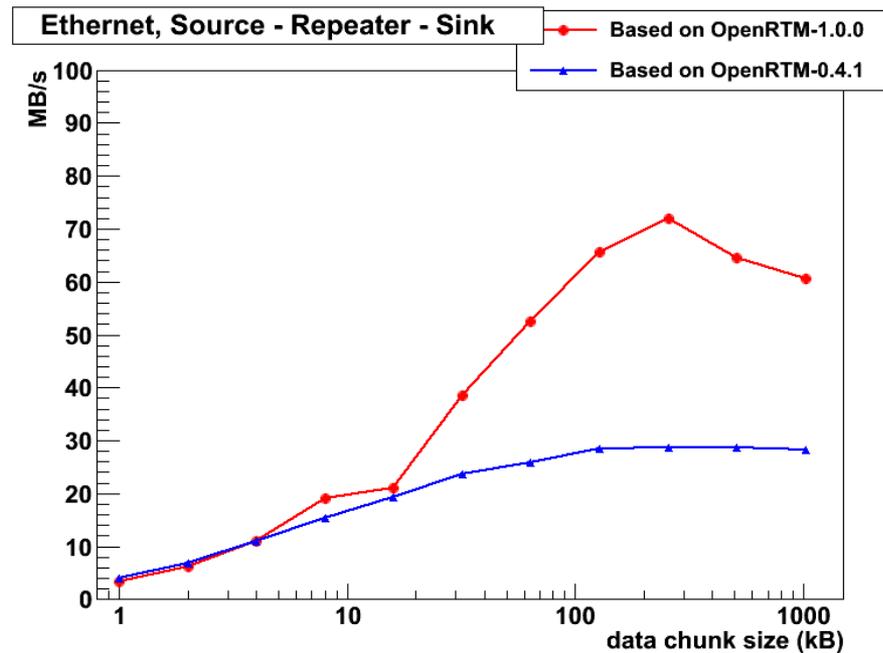
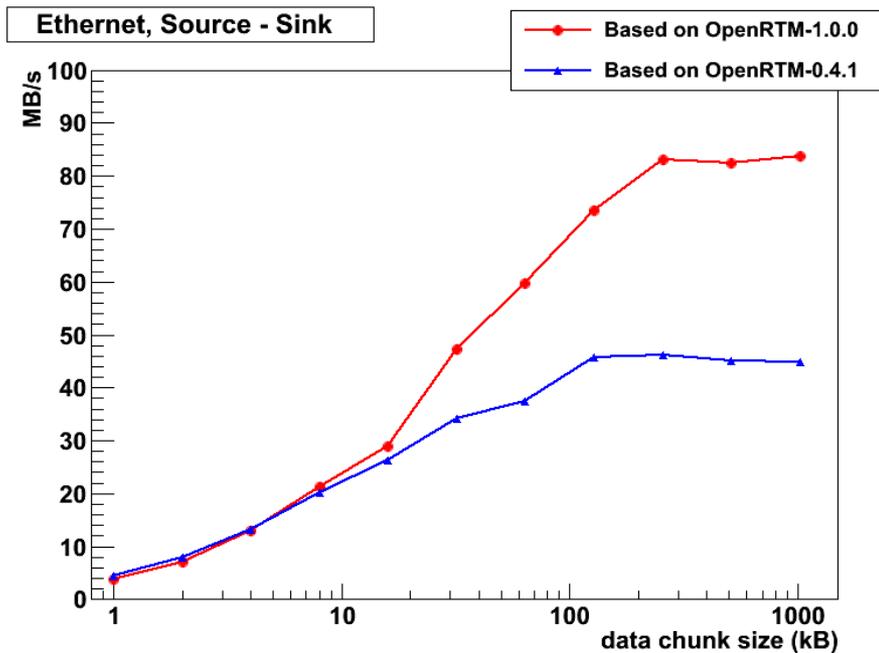
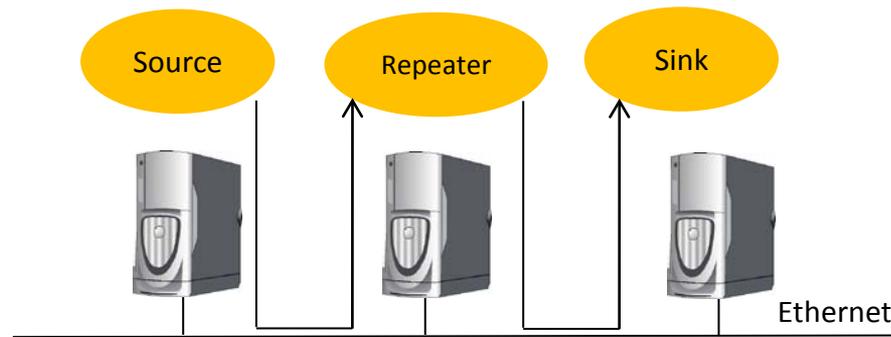
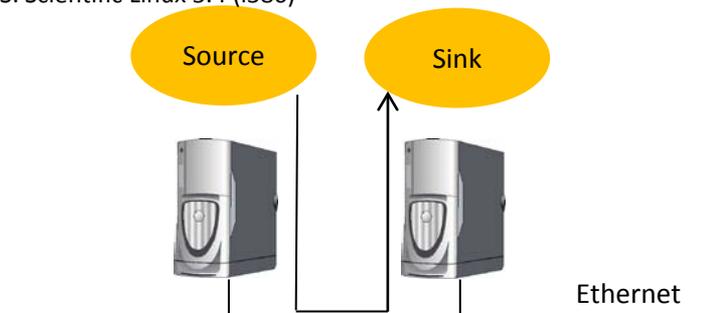
学会発表、展示会等

- CHEP (International Conference on Computing in High Energy and Nuclear Physics)
- 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

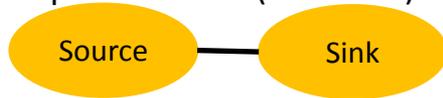
性能測定 (Ethernet)

Model : Dell PowerEdge SC1430
CPU : Intel Xeon 5120 @ 1.86GHz 2 Cores × 2
Memory: 2GB
NIC: Intel Pro 1000 PCI/e (1GbE)
OS: Scientific Linux 5.4 (i386)



性能測定 (Loopback)

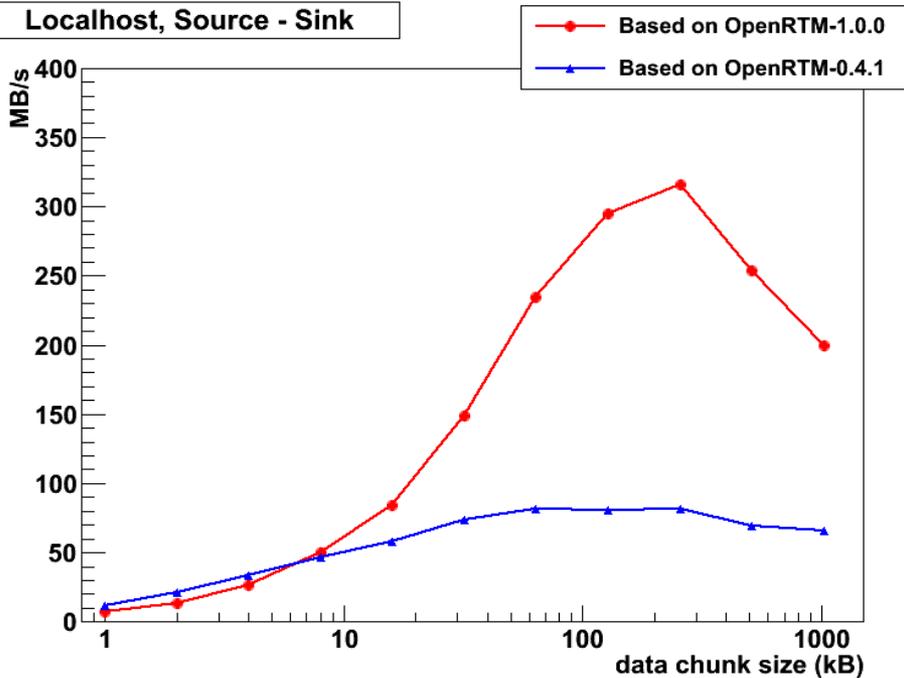
Run on one multi core CPU PC.
Each components communicate via
loopback device (localhost).



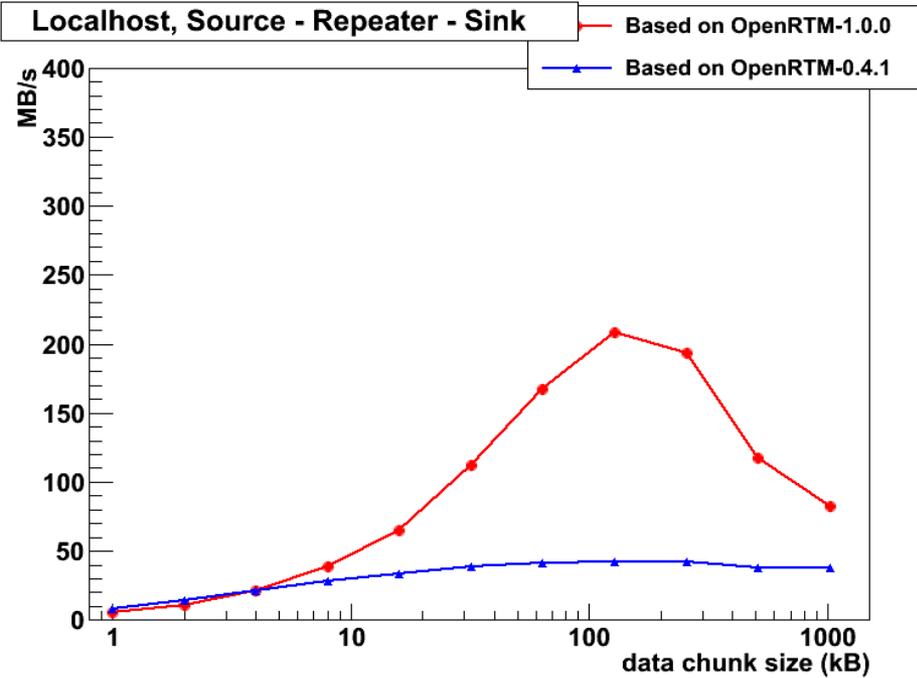
Same as left but put a Relay component
between two components.

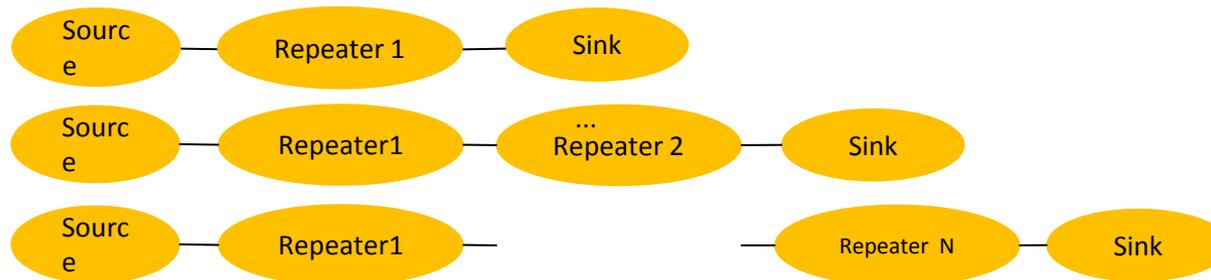


Localhost, Source - Sink

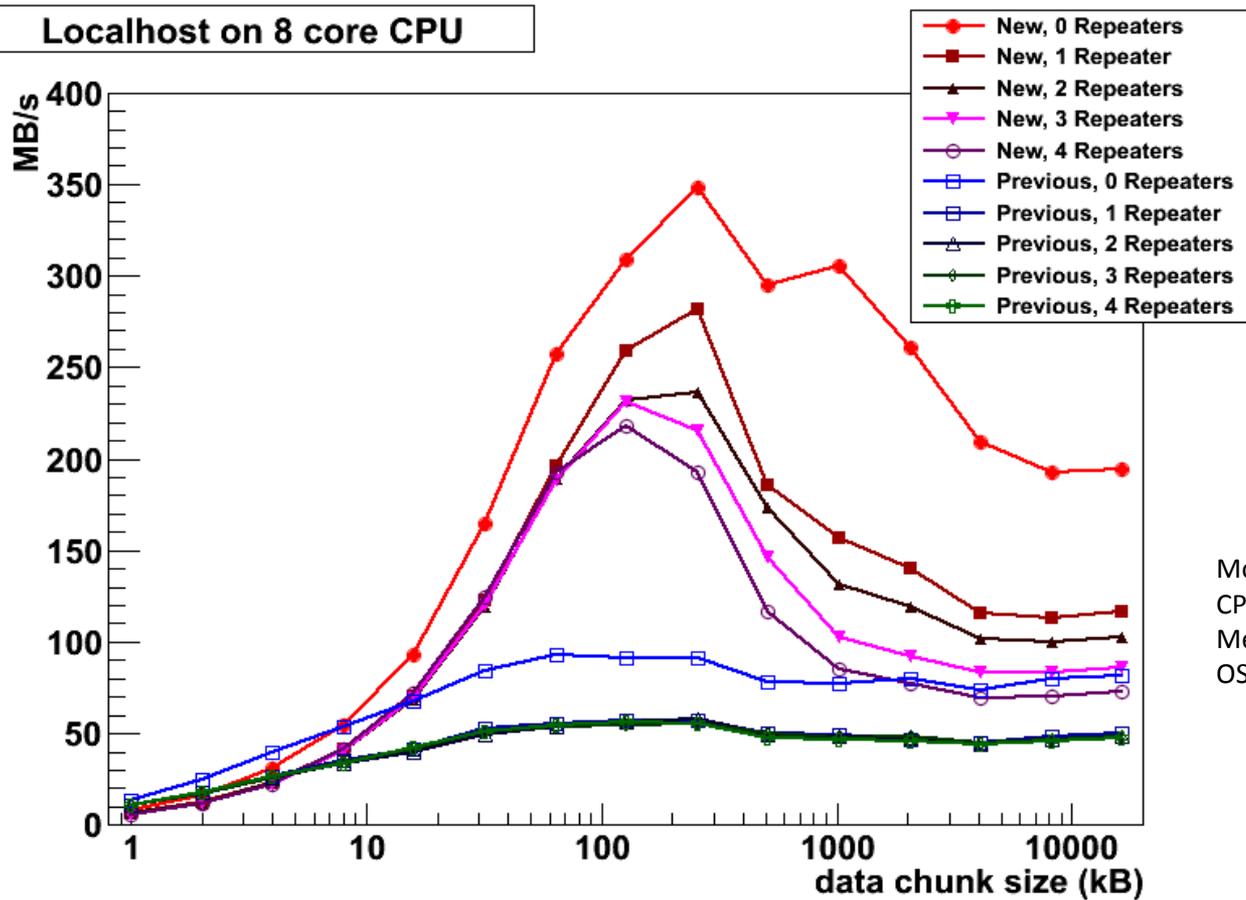


Localhost, Source - Repeater - Sink





Localhost on 8 core CPU



Model : HP xw8600
 CPU: Intel Xeon 5420 @ 2.50 GHz 4 Cores × 2
 Memory: 8GB
 OS: Scientific Linux 5.4 (i386)

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>
- ソケットを使ったネットワークプログラミング
 - SiTCPリードアウトモジュールからのデータ読みだしに必要
 - 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 32bit, 64bit) (CentOS 5.x, RHEL 5.x)用RPMがあるのでこれを使ってセットアップ
 - `wget http://daqmw.kek.jp/src/daqmw-rpm`
 - `sh daqmw-rpm install`
 - SL 5.x 以外で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

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

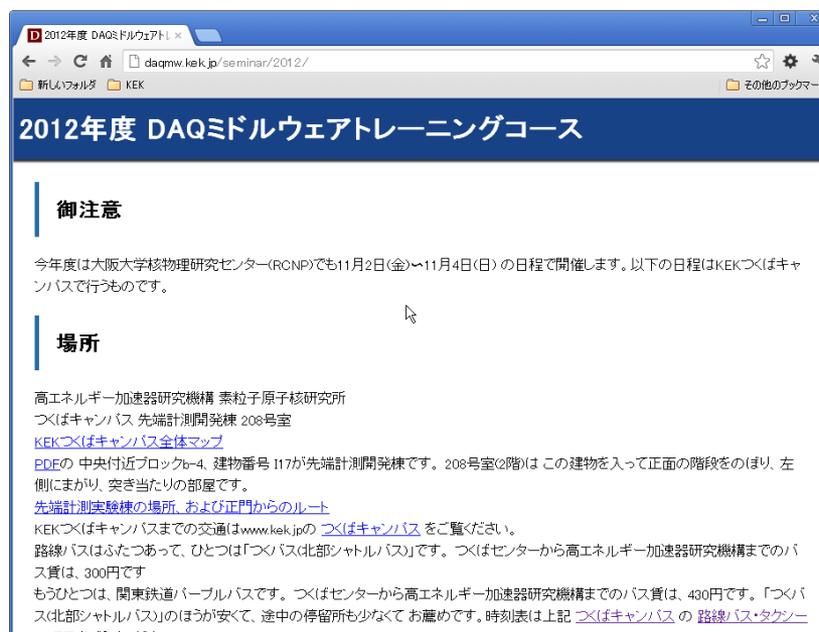
- DAQ-Middleware 1.2.0開発マニュアル

<http://daqmw.kek.jp/docs/DAQ-Middleware-1.2.0-DevManual.pdf>

DAQ-Middlewareトレーニングコース

- 毎年夏に開催
- 今年はKEK外でも開催

11月2日～4日 大阪大学核物理研究センター



2012年度 DAQミドルウェアトレーニングコース

御注意

今年度は大阪大学核物理研究センター(RCPN)でも11月2日(金)～11月4日(日)の日程で開催します。以下の日程はKEKつくばキャンパスで行うものです。

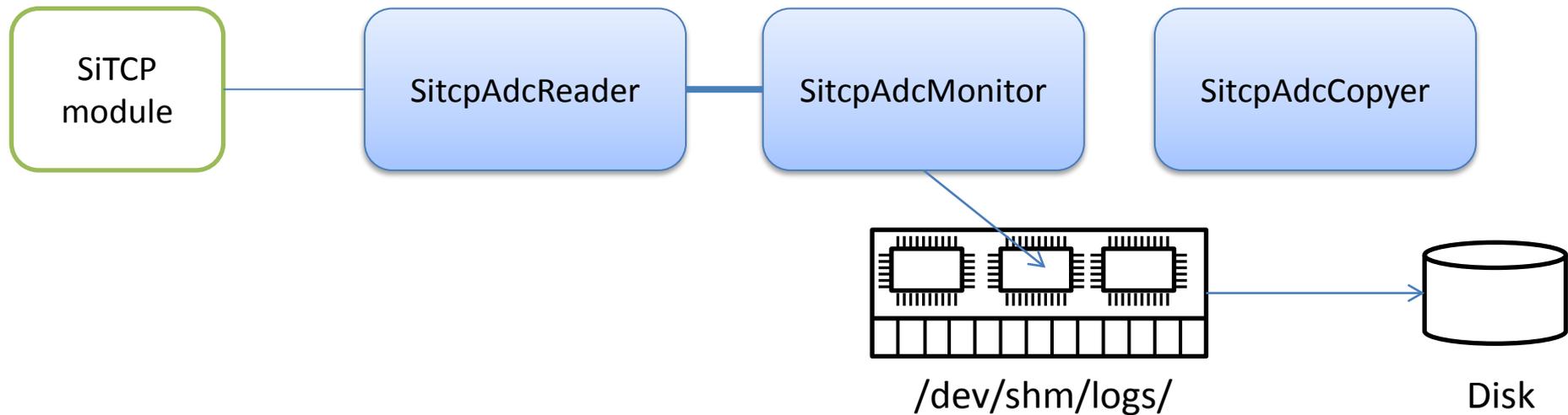
場所

高エネルギー加速器研究機構 素粒子原子核研究所
つくばキャンパス 先端計測開発棟 208号室
[KEKつくばキャンパス全体マップ](#)
PDEの中央付近ブロックb-4、建物番号 117が先端計測開発棟です。208号室(2階)はこの建物に入って正面の階段をのぼり、左側にまわり、突き当たりの部屋です。
[先端計測実験棟の場所、および正門からのルート](#)
KEKつくばキャンパスまでの交通は[www.kek.jp](#)の [つくばキャンパス](#) をご覧ください。
路線バス(はふたつあって、ひとつは「つくばバス(北部シャトルバス)」です。つくばセンターから高エネルギー加速器研究機構までのバス賃は、300円です
もうひとつは、関東鉄道バープルバスです。つくばセンターから高エネルギー加速器研究機構までのバス賃は、430円です。「つくばバス(北部シャトルバス)」のほうが安くて、途中の停留所も少なくおすすめです。時刻表は上記 [つくばキャンパスの路線バス・タクシー](#) の項目をご覧ください。

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`

最近行っているセンサーテストでの構成



ROOTでヒストグラムモード

- 1回の読み出しで4096チャンネル分のデータがくるセンサーのテスト実験
- イベントデータとして保存する必要はない
- 4096個のROOT TH1Fヒストグラムを作成(ヒストグラムモードでデータ収集)
- メモリー上にヒストグラムデータがあるので停電すると消える
- ときどきディスクに書きだすか

ROOT TFile

- ヒストグラムデータの保存にROOTファイルを使う
- `root -l adc_1234.root`
`root> ADC_1234->Draw()`でヒストグラムが書ける
- ファイルサイズを一定にするためには
`histogram.Write("", TObject::kOverwrite)`
する必要がある
- `kOverwrite`を指定すると`Write()`のたびに`fsync`する
- SL 6.2 ext4ファイルシステムでは`fsync`に30ミリ秒程度かかる

ROOT fsync

```
% strace -o fsync.log -tt -T ./sample
% grep fsync fsync.log
15:01:40.569934 fsync(7)           = 0 <0.039274>
15:01:40.609934 fsync(7)           = 0 <0.032640>
15:01:40.643034 fsync(7)           = 0 <0.032965>
```

- 1個のヒストグラムを書くのに0.1秒程度かかる。
- 4096個のヒストグラムだと400秒

回避方法

- 当座の回避方法としてディスクにかかずに /dev/shm/ (tmpfs) に書く。
 - tmpfs: swap領域でバックアップされたメモリー上のファイルシステム。
システムのメモリーがひっ迫してくると/dev/shm/に置いたファイルはswap領域に退避させられ、メモリーを必要としているプロセスにまわされる。
 - tmpfsなら450マイクロ秒で書ける
- 停電になると消える(振り出しにもどる)
- ときどき/dev/shm/に書いたROOTファイルをディスクに書くDAQコンポーネントを作成