

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

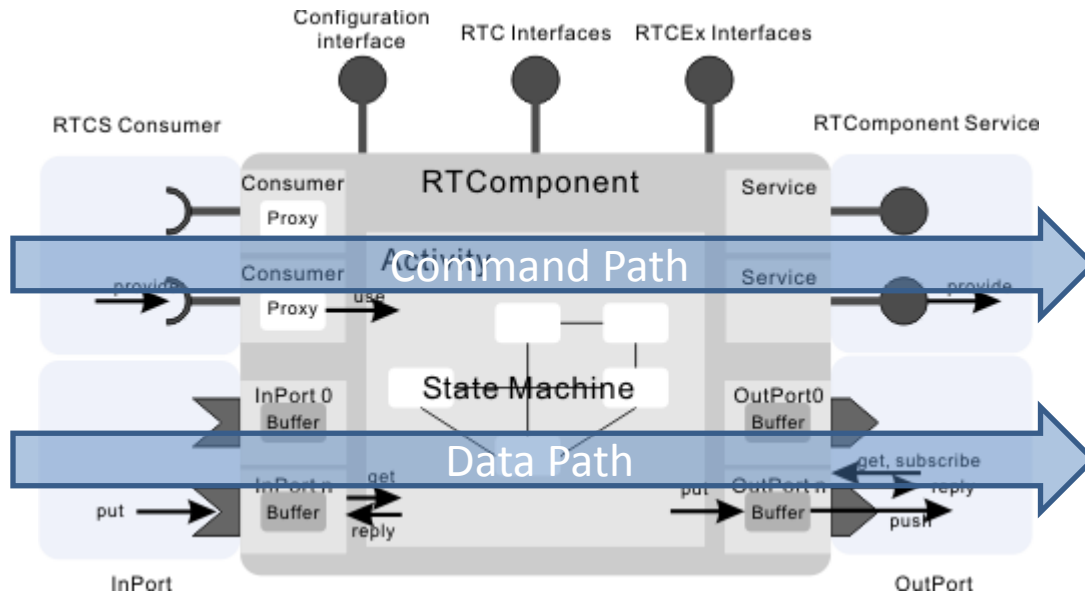
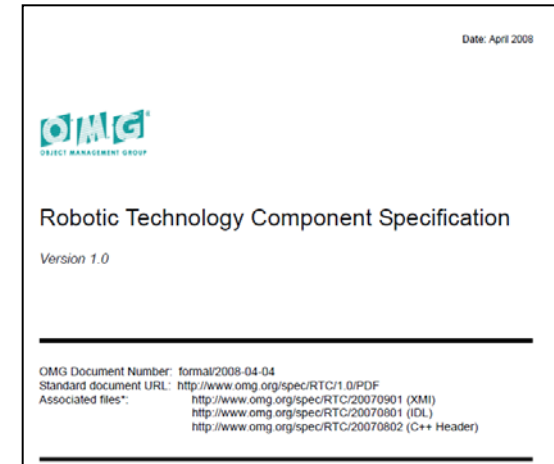
- 背景
 - 従来DAQシステムでのソフトウェアの再利用化はドライバ、ライブラリレベルでおこなわれてきた。
 - 扱うデータの増大、使う計算機の数が増えてきてDAQシステムを構築するのがむずかしくなってきた。
- 解決案？
 - どんな実験にも対応できるように、抽象化、汎用化してしまうとデータ収集効率が落ちてしまう。
- 解決案
 - ドライバ、ライブラリとDAQシステムの間にコンポーネントという中間層を作り、実験毎の違いを吸収、収集効率を確保し、
 - システムの枠組みは普遍であるDAQフレームワークを作ればよいのではないか？

DAQ-Middleware (1)

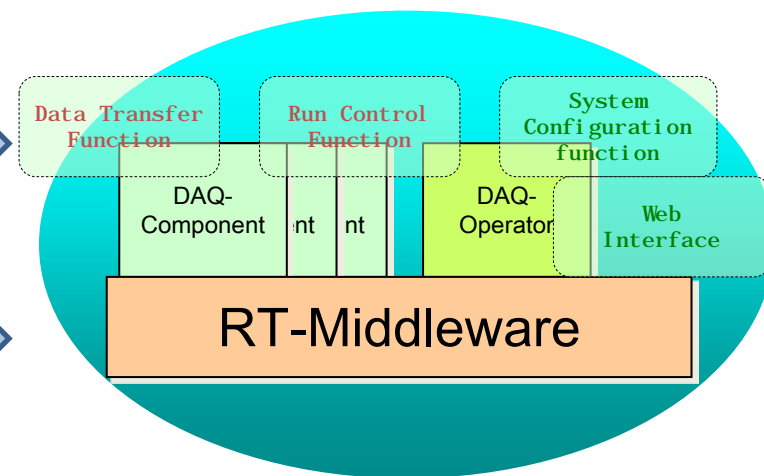
- 目的
 - 再利用可能な
 - 柔軟な
 - 容易に構築可能なDAQシステム
- 特徴
 - コンポーネントベース
 - ネットワークベース
 - フレームワーク

DAQ-Middleware (2)

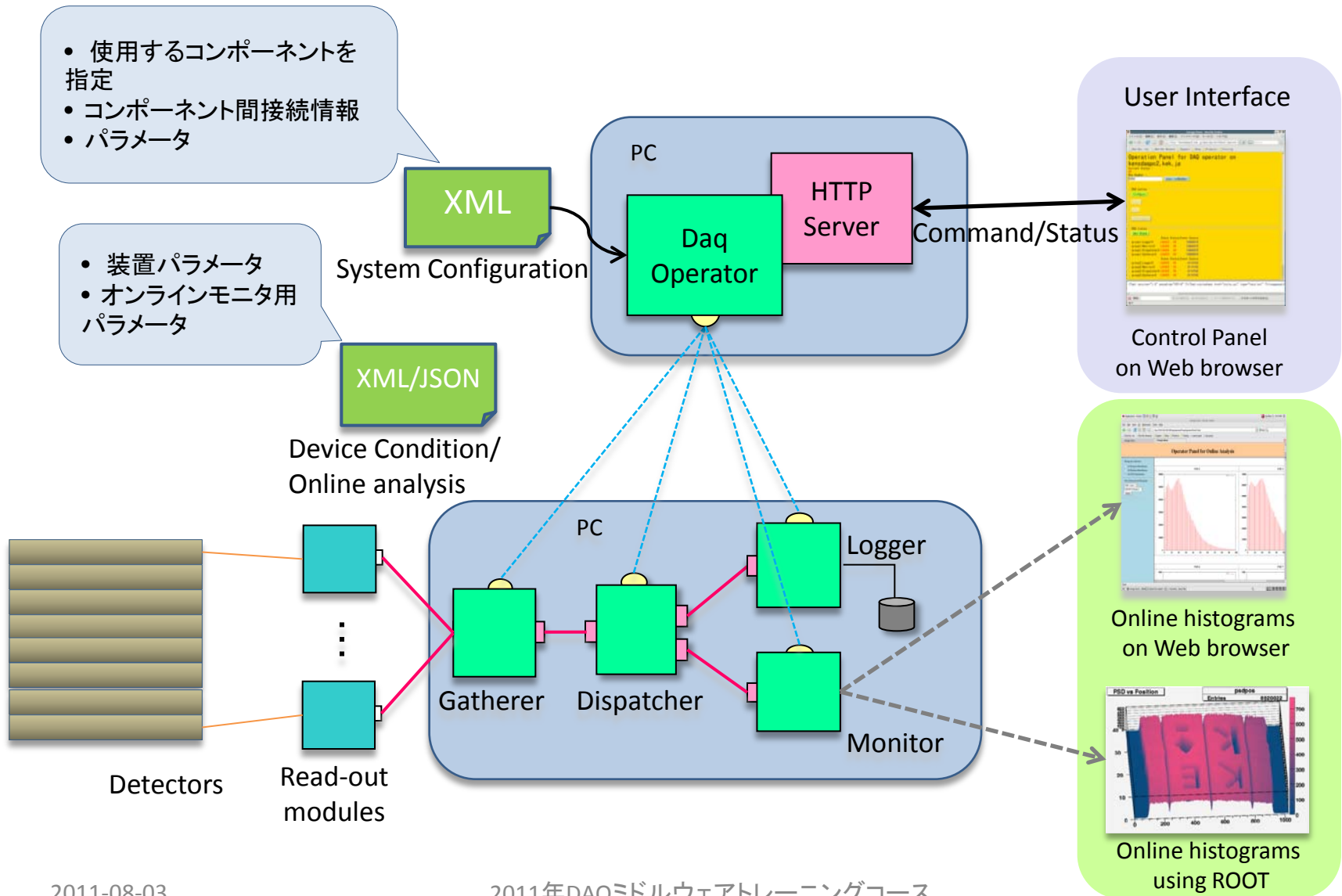
- RT(Robot Technology)-Middlewareをデータ収集用に拡張
- RT-Middleware
 - ネットワークロボットシステムの構築のためのソフトウェア共通プラットフォーム
 - 産総研知能システム研究部門・タスクインテリジェンス研究グループが開発
 - 複数のコンポーネントが通信してひとつの機能を実現する
 - そのソフトウェアコンポーネントの仕様は国際標準規格(OMG)
 - 我々は2006年から産総研と共同研究を行っている



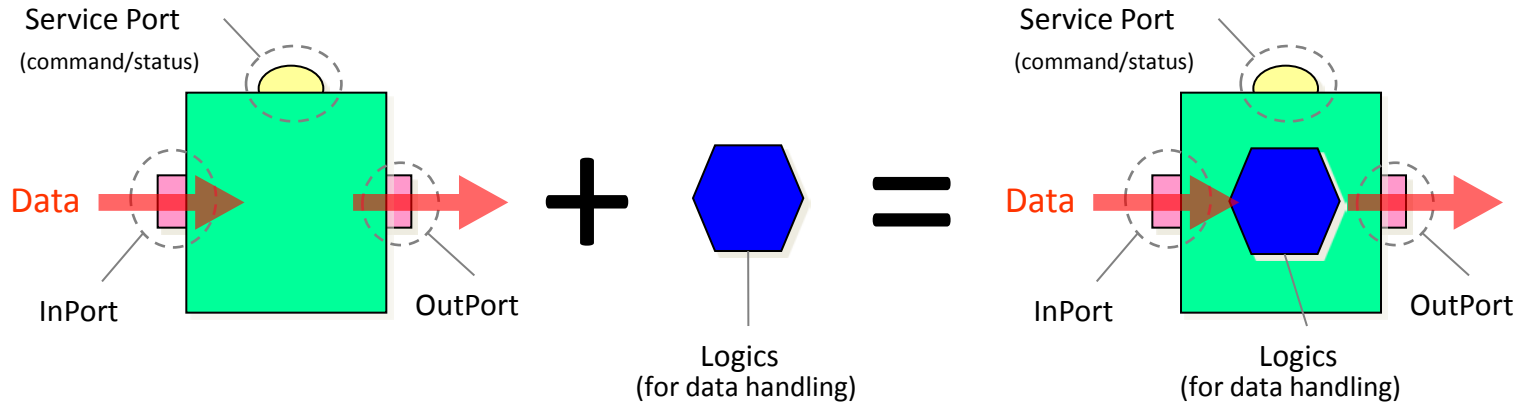
DAQ-Middleware



基本DAQモデル

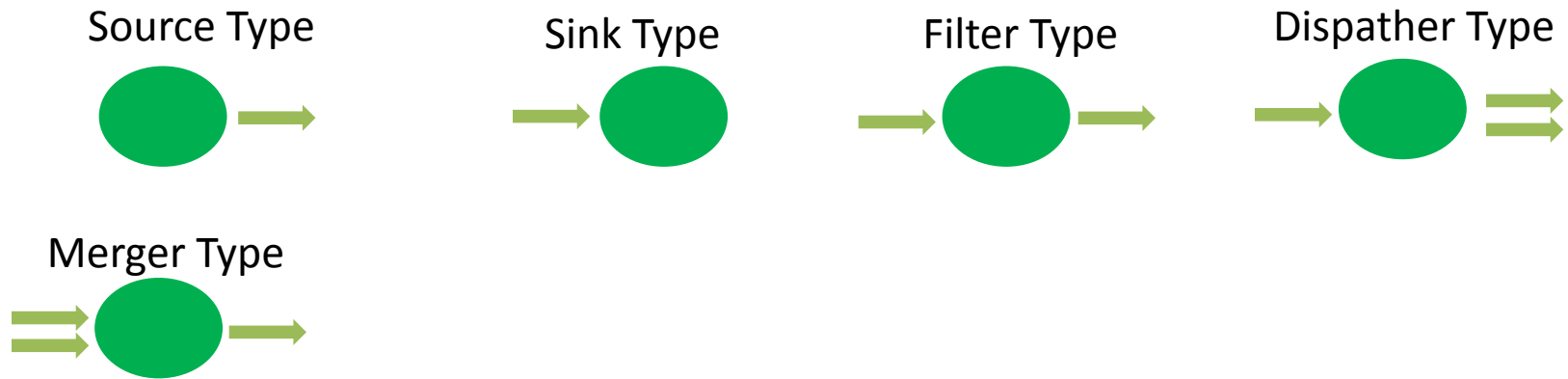


DAQコンポーネント



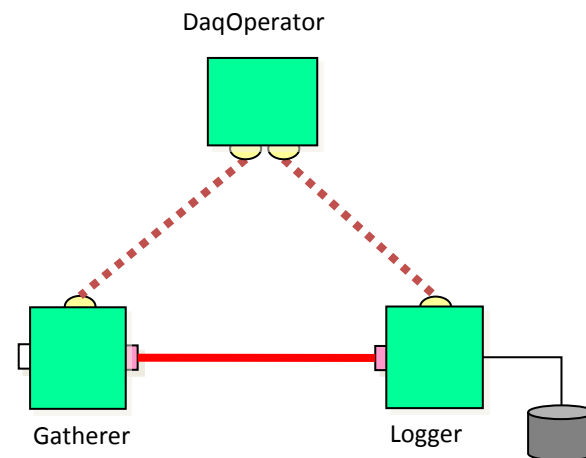
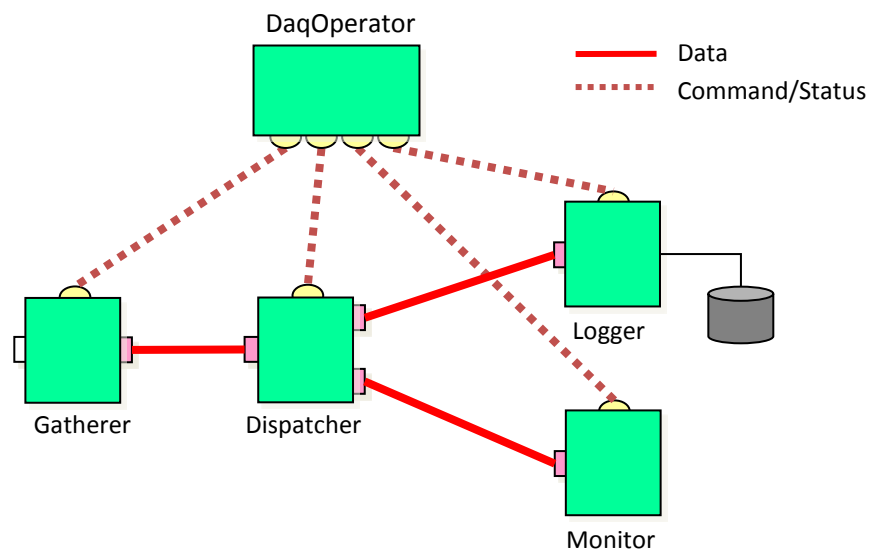
- DAQコンポーネントを組み合わせてDAQシステムを構築する
- データ転送機能、ランコントロール、システムコンフィギュレーション機能はDAQ-Middlewareで実装済み。
- データを下流に送るにはOutPortに書く。
- 上流からのデータを読むにはInPortを読む。
- ユーザーはコアロジックを実装することで新しいコンポーネントを作成できる。
コアロジックの例：
 - リードアウトモジュールからのデータの読み取りロジック
 - ヒストグラムの作成ロジック

コンポーネント間通信での分類

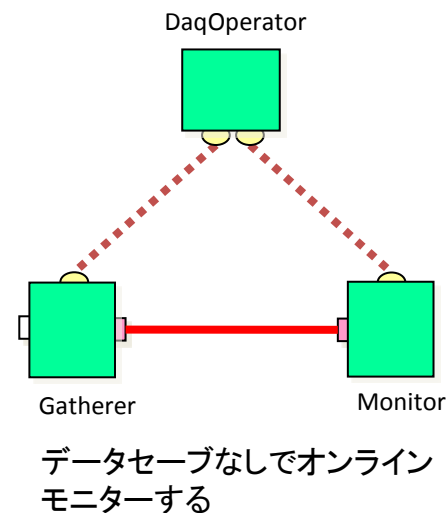
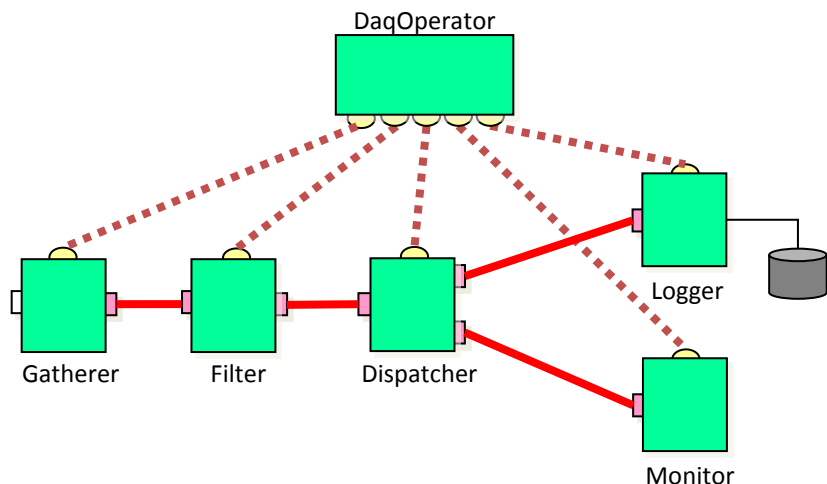


- Source Type (Gatherer)
- Sink Type (Logger, Monitor)
- Dispatcher Type

DAQコンポーネント構成例(1)



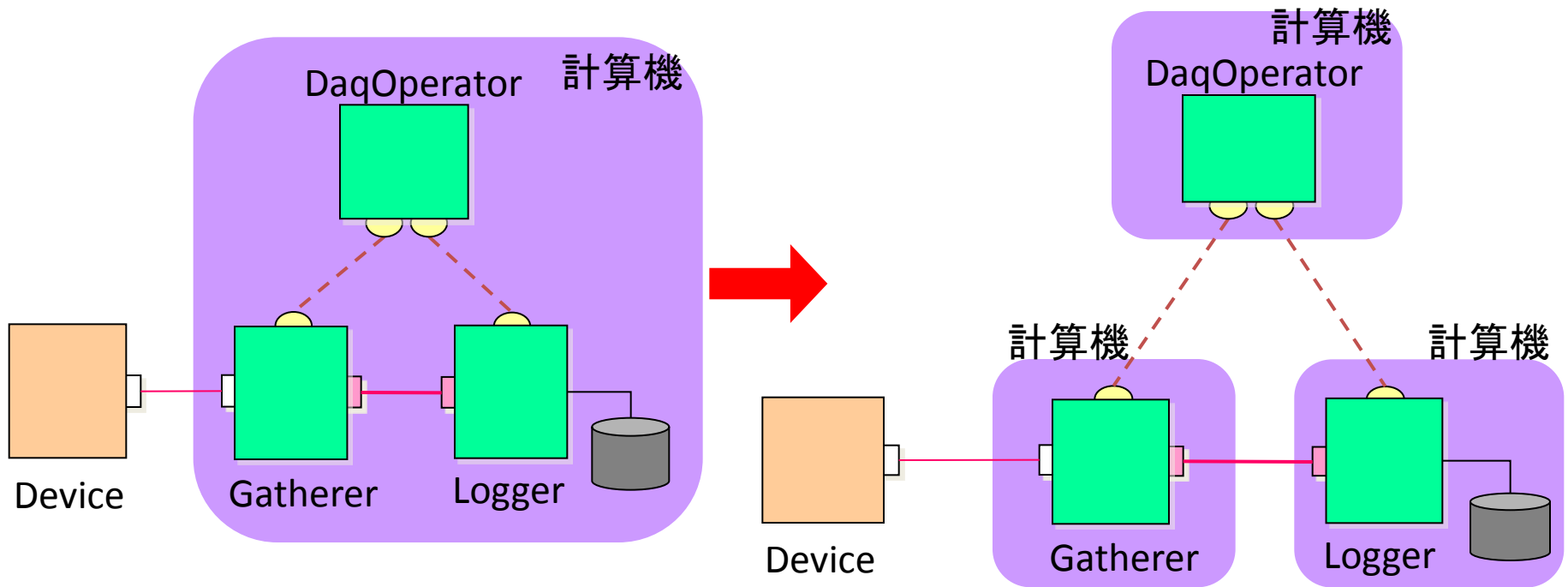
モニターなしでデータをディスクに
セーブする



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)

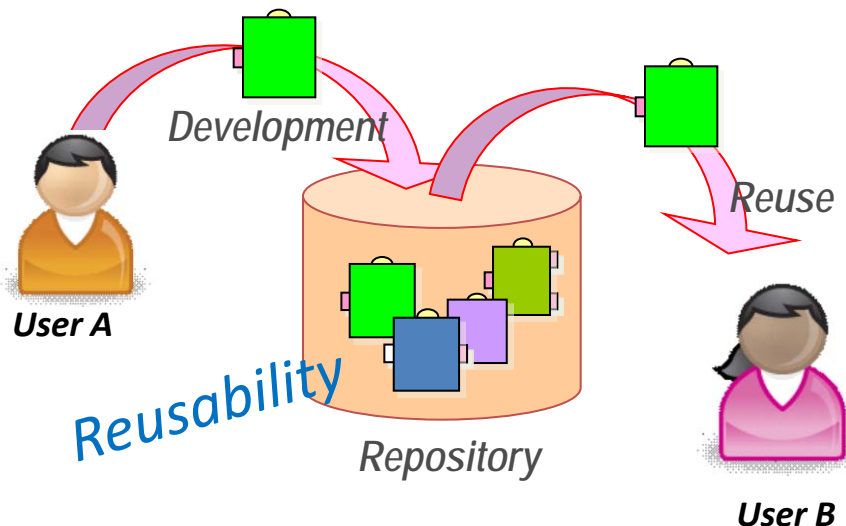
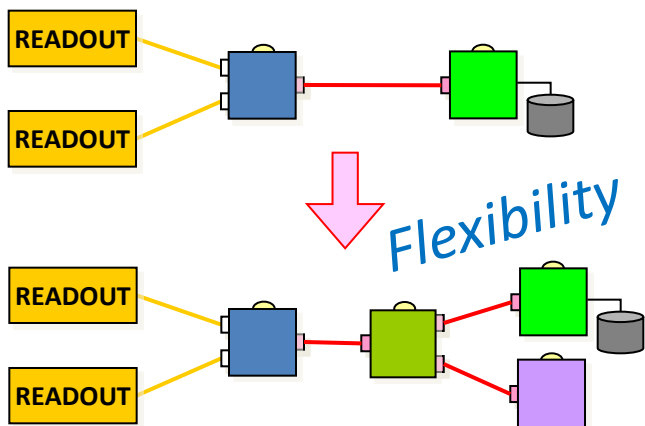
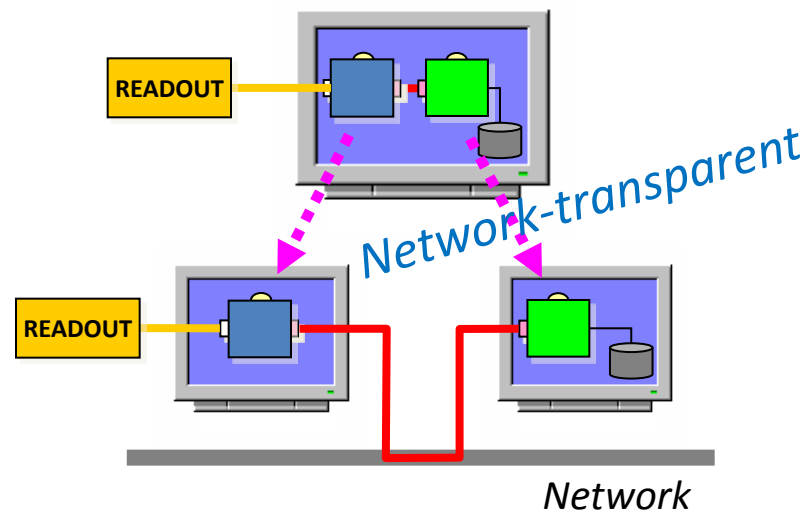
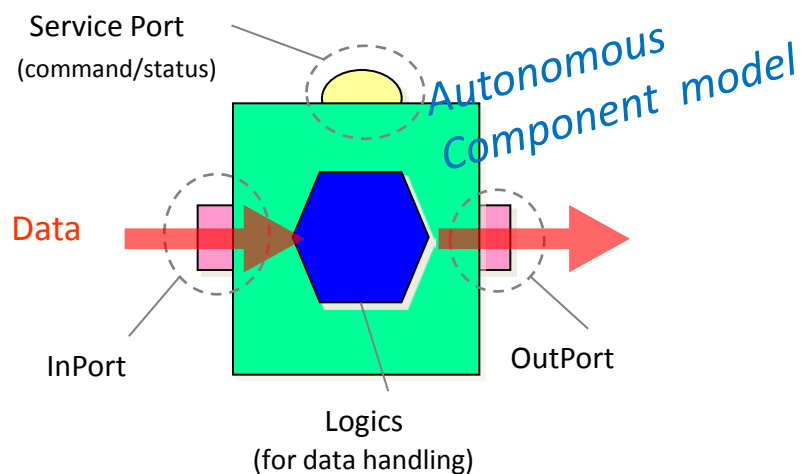
ネットワーク透過性



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

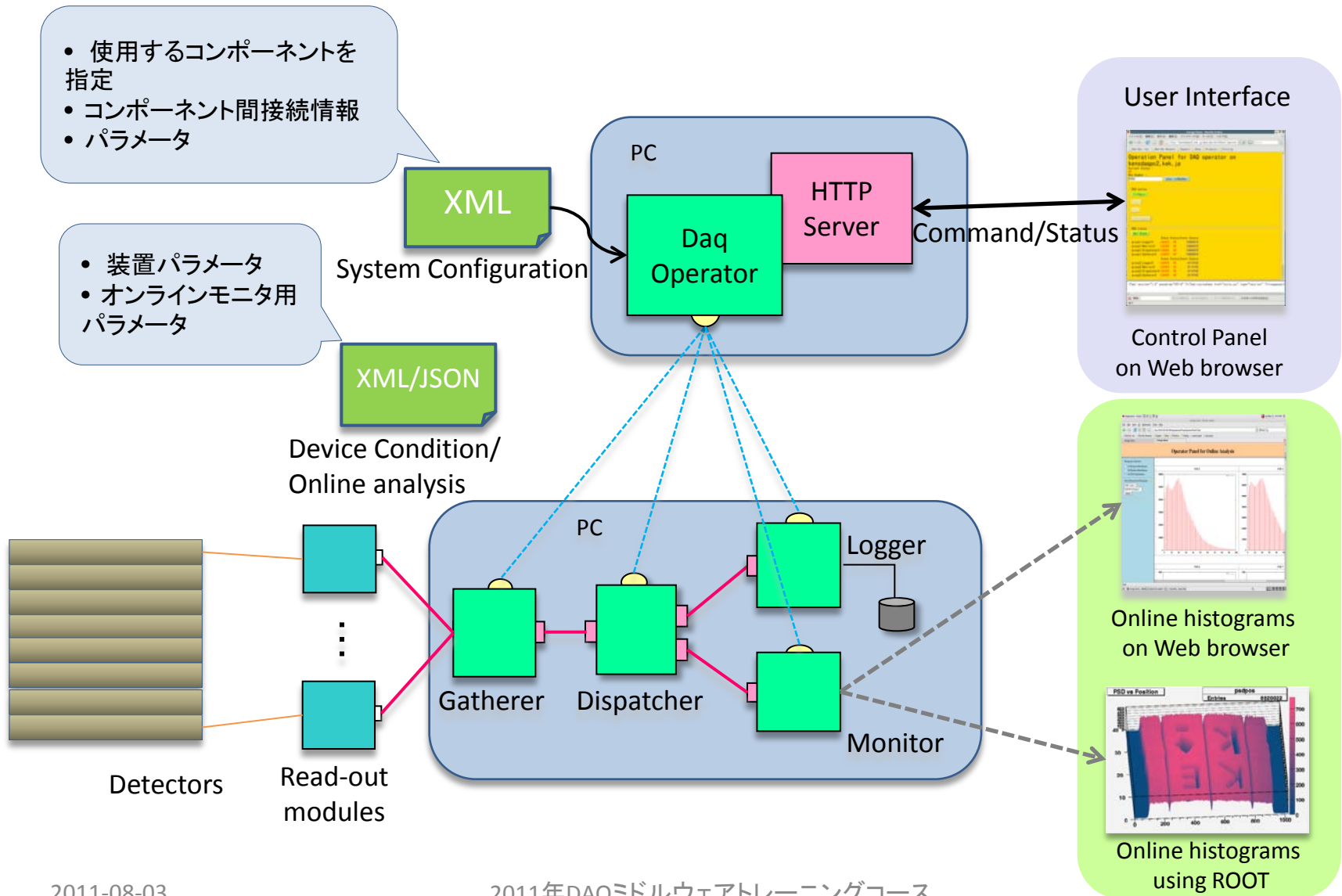
DAQコンポーネント特徴のまとめ



DaqOperator

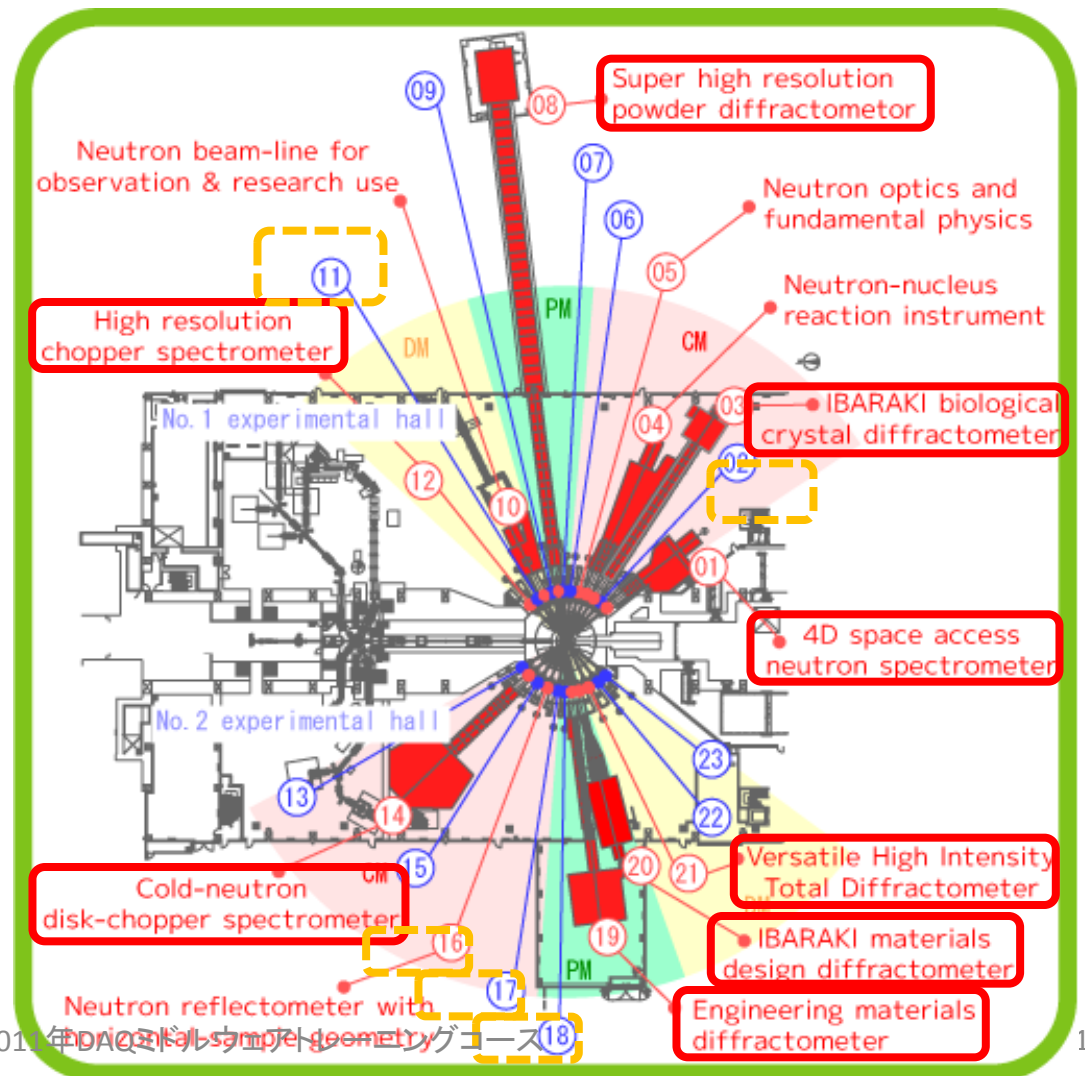
- DAQコンポーネント統括者
 - システムの把握はコンフィギュレーションXMLファイルを読んで行う。
- DAQコンポーネントに対して接続、スタート、ストップ等の指示をだす。
 - 指示を出すときにはすでにDAQコンポーネントはプロセスとして起動している必要がある
 - DaqOperatorが起動するわけではない。
 - DAQコンポーネントの起動方法
 - ローカル計算機ならスクリプト内部 (run.py -cl)
 - ネットワーク経由でのブート(xinetdを使う等)

基本DAQモデル



DAQ-Middleware 使用例

J-PARC MLF中性子実験での 使用状況



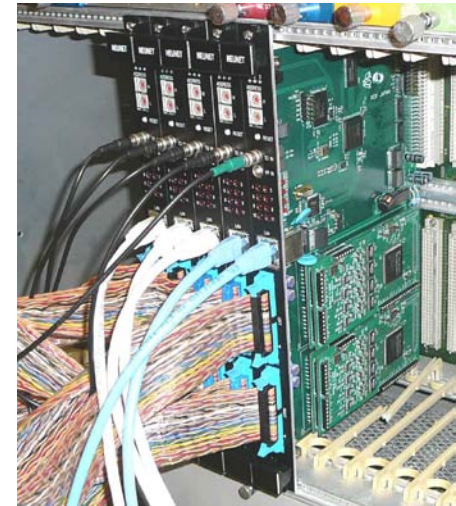
working

will be introduced

2010年6月現在

J-PARC/MLF 中性子 検出器・リードアウトモジュール

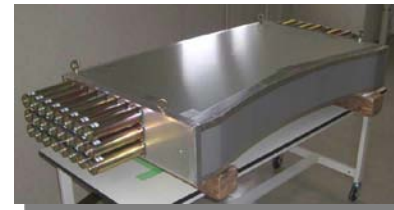
- Position Sensitive Detector (PSD)
 - ^3He filled proportional counter
 - The most common neutron detector
- Photon-counting 2-D/1-D detector (Scinti)
- Gas Electron Multiplier (GEM)
- SiTCP-based Readout modules
 - SiTCP: a hardware-based TCP processor



PSDs



2-D Scinti

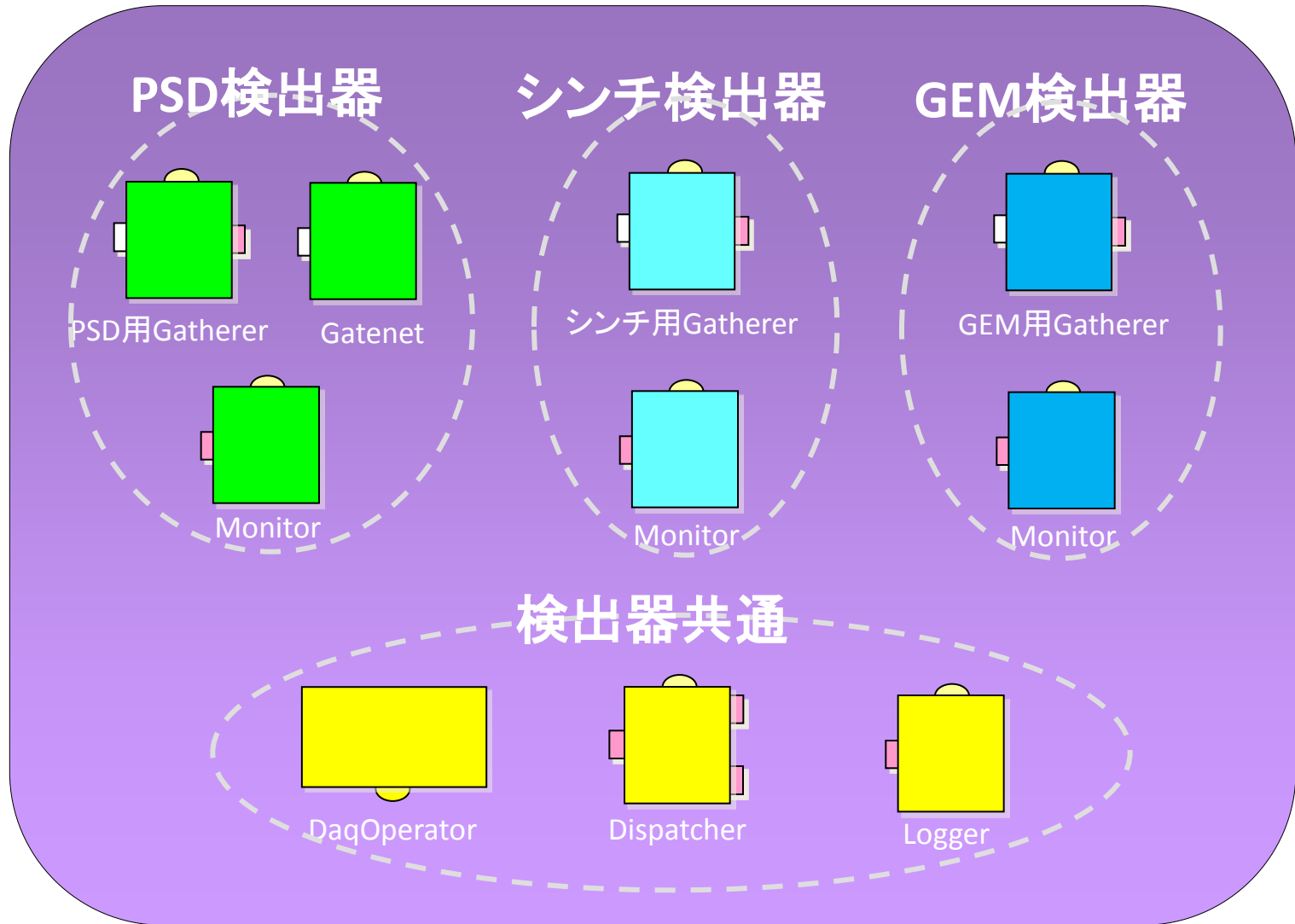


1-D Scinti



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.

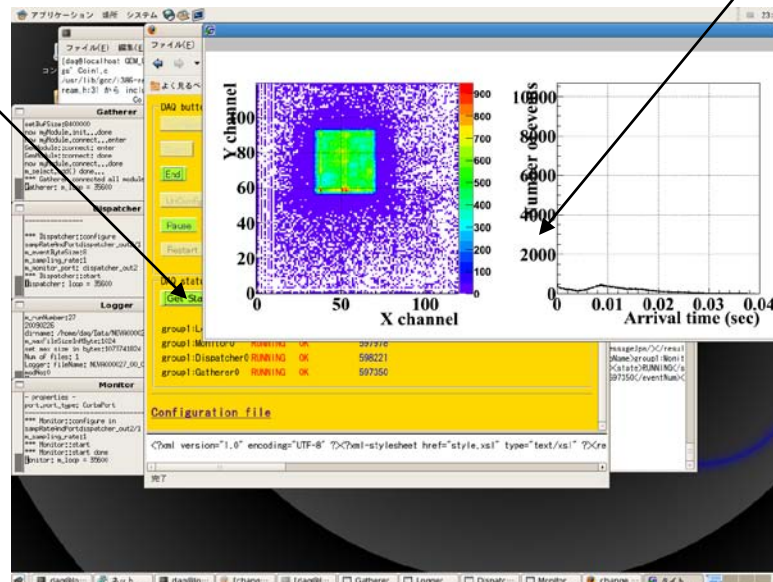
Users are able to take data without regard for the difference of detectors and to control the detectors from a web browser.

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

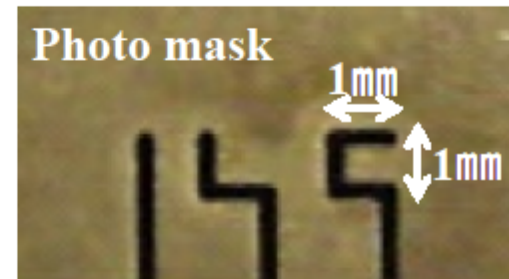
Tomoyuki Saito (Tohoku U.)

14

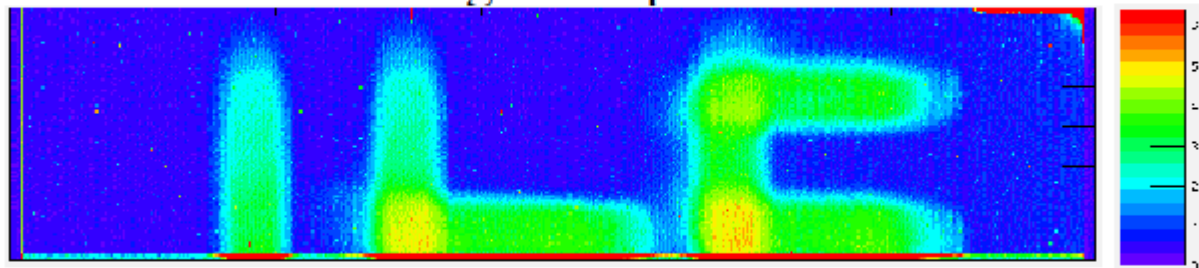
CCD readout test : Test with LED

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 !

福井大学

• CAMACのマルチチャンネル読み出し用DAQ-Middlewareの開発 (日本物理学会北陸支部)

CAMACのマルチチャンネル読み出し用DAQ-Middlewareの開発
福井大学工 大阪大学理^A KEK^B
前田翔平 玉川洋一 小川 泉 味村周平^A 能町正治^A 仲吉一男^B
Development of DAQ-Middleware for multichannel reading of CAMAC
Fac.of Eng., Univ. of Fukui Fac.of.Sci Osaka Univ^A KEK^B
S.Maeda Y.Tamagawa I.Ogawa S.Ajimura^A M.Nomachi^A K.Nakayoshi^B

DAQ(Data Acquisition)ミドルウェアは、ネットワーク分散環境でデータ収集用ソフトウェアを容易に構築するためのソフトウェア・フレームワークである。DAQ-MiddlewareにはDAQコンポーネントと呼ばれるソフトウェア・コンポーネントがあり、データを読み取るReader、データを処理し結果を表示するMonitorなどのコンポーネントをユーザは任意に組み合わせてDAQシステムを構築する。DAQ-Middlewareの利点は、開発は各コンポーネントに対して集中すればよく、それらのコンポーネントによりシステムを構成する際に必要なことはすべてミドルウェアで用意されている。また、ネットワーク分散環境でDAQシステムを構築することができるので、高負荷の実験にも耐えることができる。

現在、このDAQ-MiddlewareでCAMACのマルチチャンネル読み出しを開発している。今回は、DAQ-Middlewareのバージョンアップにともない、シングルチャンネル読み出しから新たに開発し直し、これを基にマルチチャンネル読み出しを開発した。これにより、ADCだけでなくTDCも同時に読み出し可能となり、検出器からのデータ収集で必要最小限のシステム構成となる。さらにMonitorの機能拡張も同時に行った。このシステムを用いて、試行実験とデータ解析を行った。講演では、DAQ-Middlewareの概要、開発手順、今後の展望について報告する。

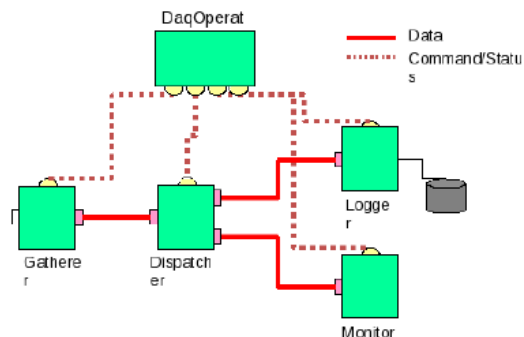
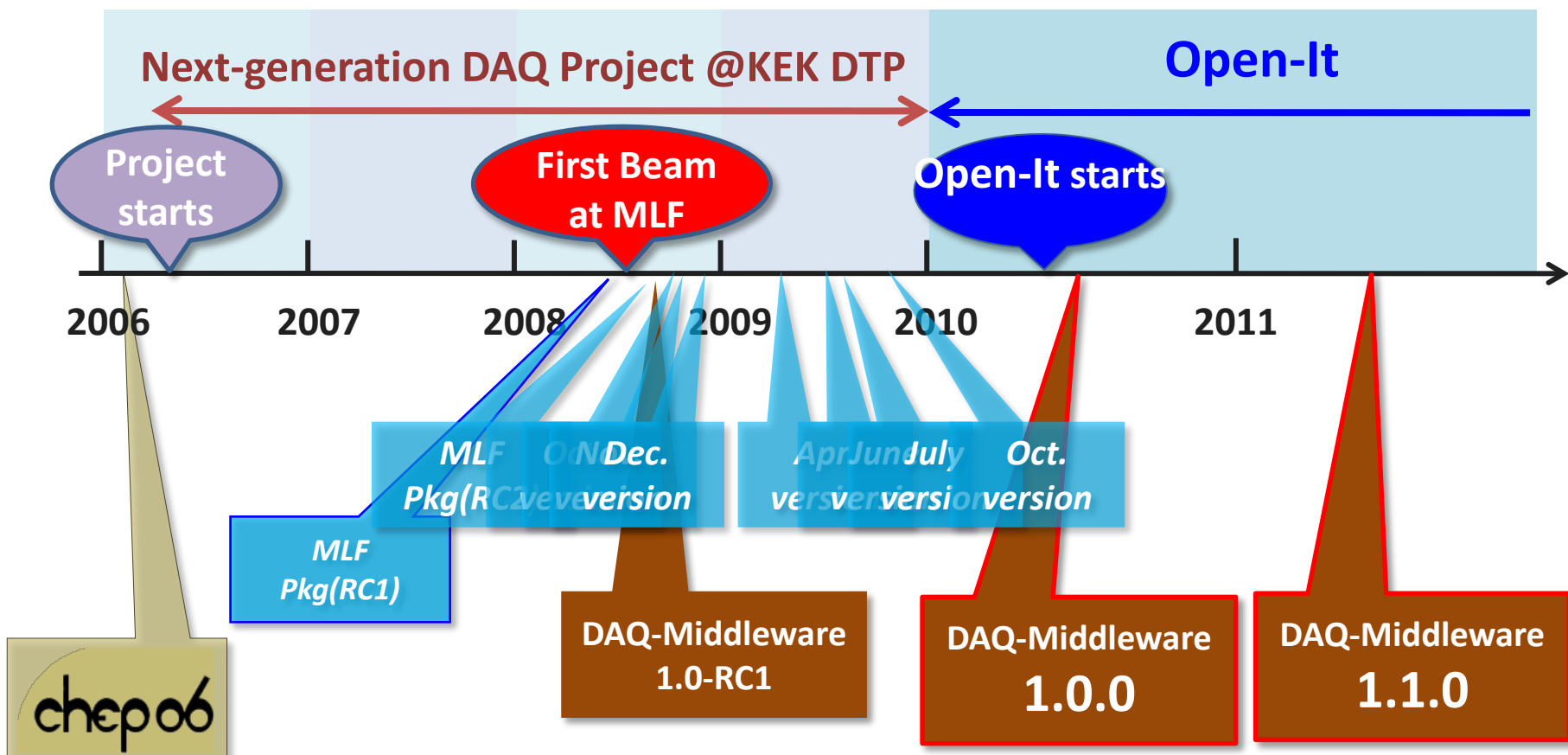


図1. DAQコンポーネント構成例

開発体制など

DAQ-Middlewareの歴史



開発体制

- 2010年4月 DAQ-Middleware Core グループ
結成
- メンバー
 - 仲吉、千代、安、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)

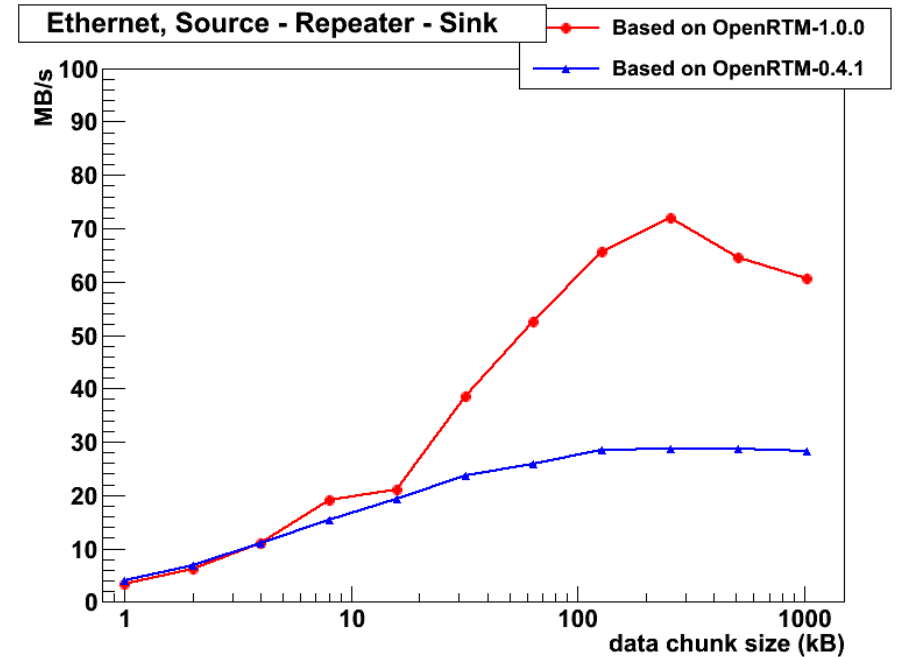
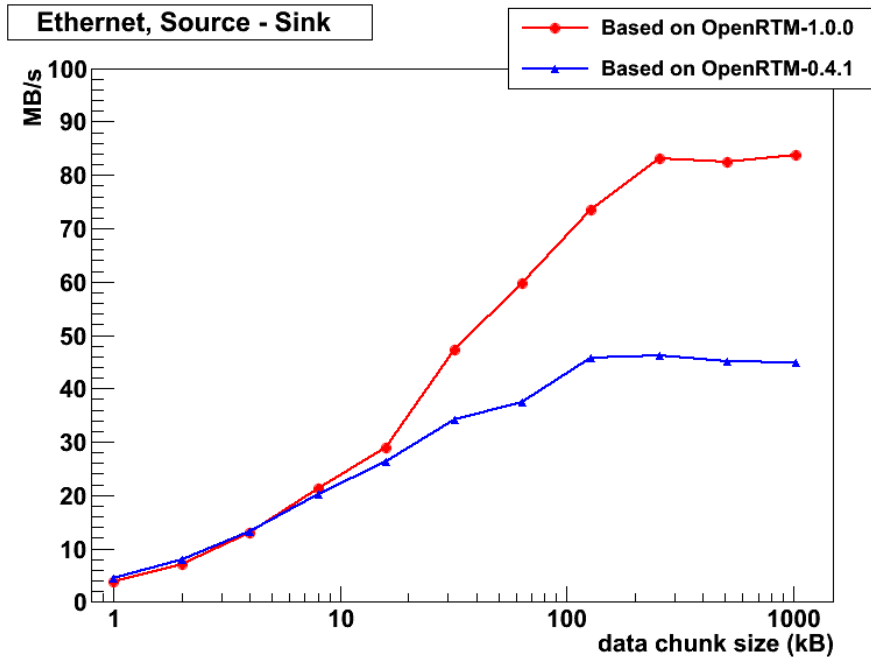
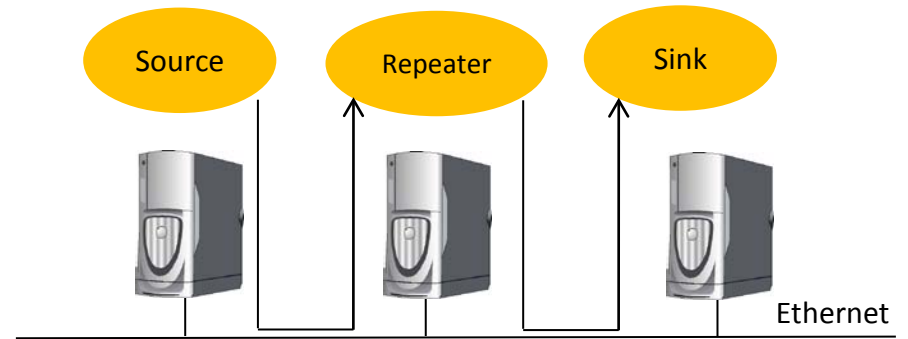
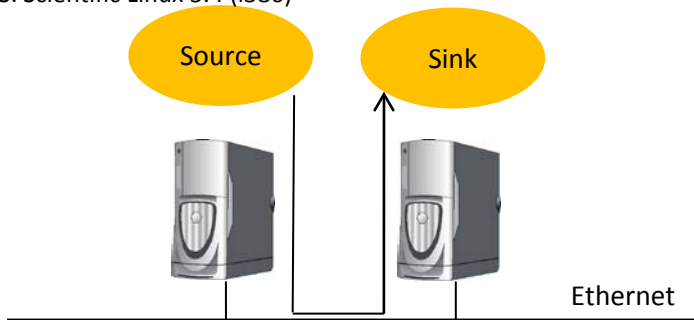
学会発表、展示会等

- 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)

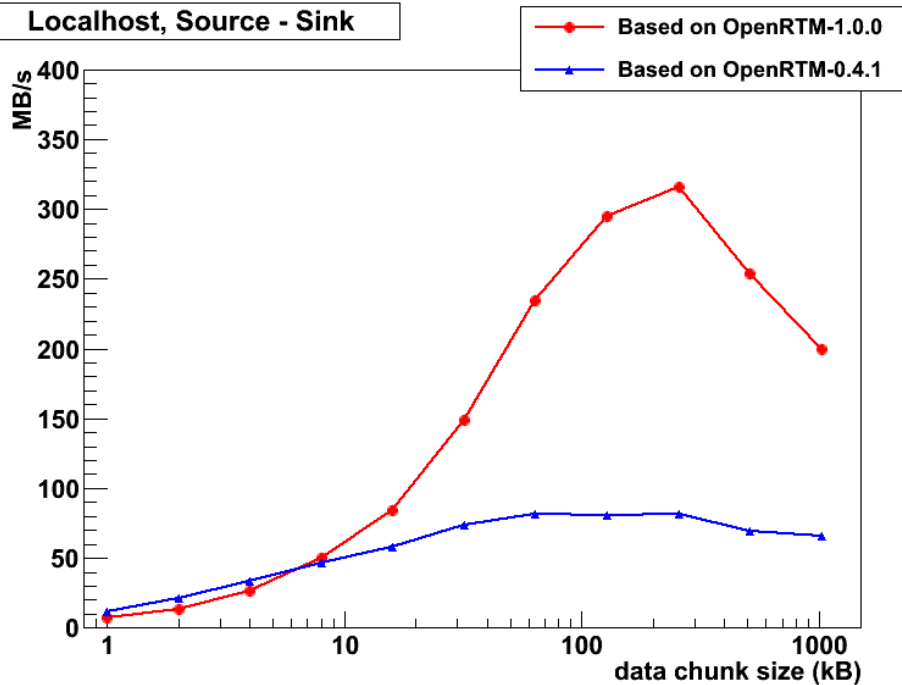
Run on one multi core CPU PC.
Each components communicate via
loopback device (localhost).



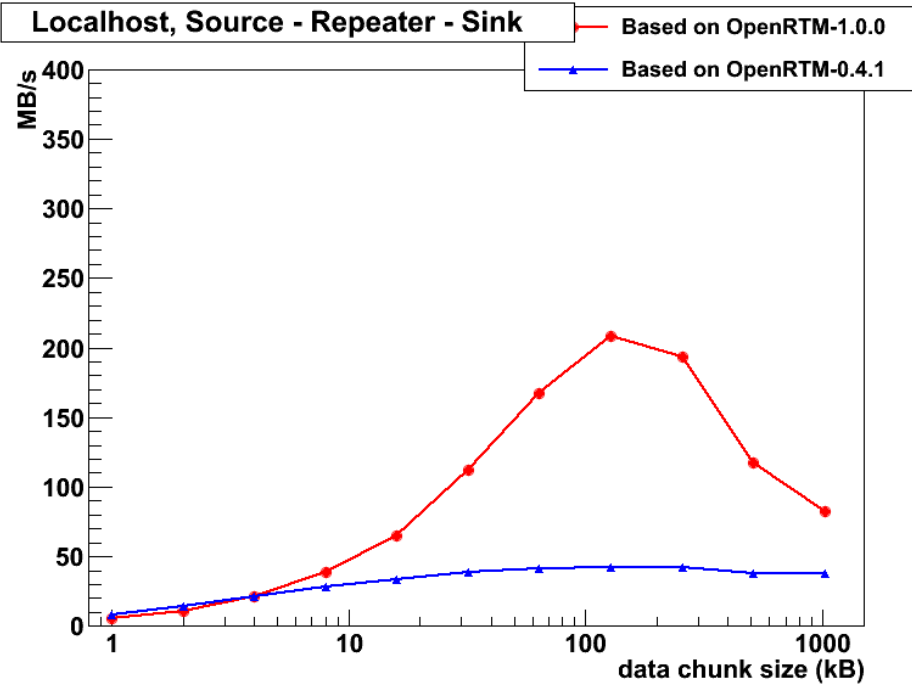
Same as left but put a Repeater component
between two components.

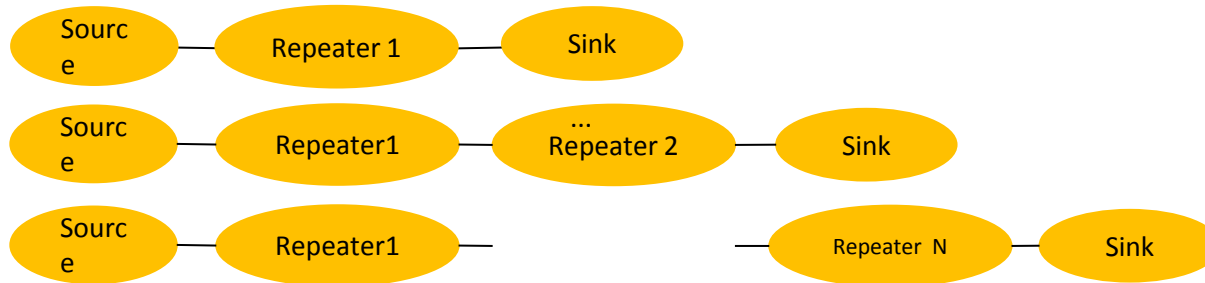


Localhost, Source - Sink

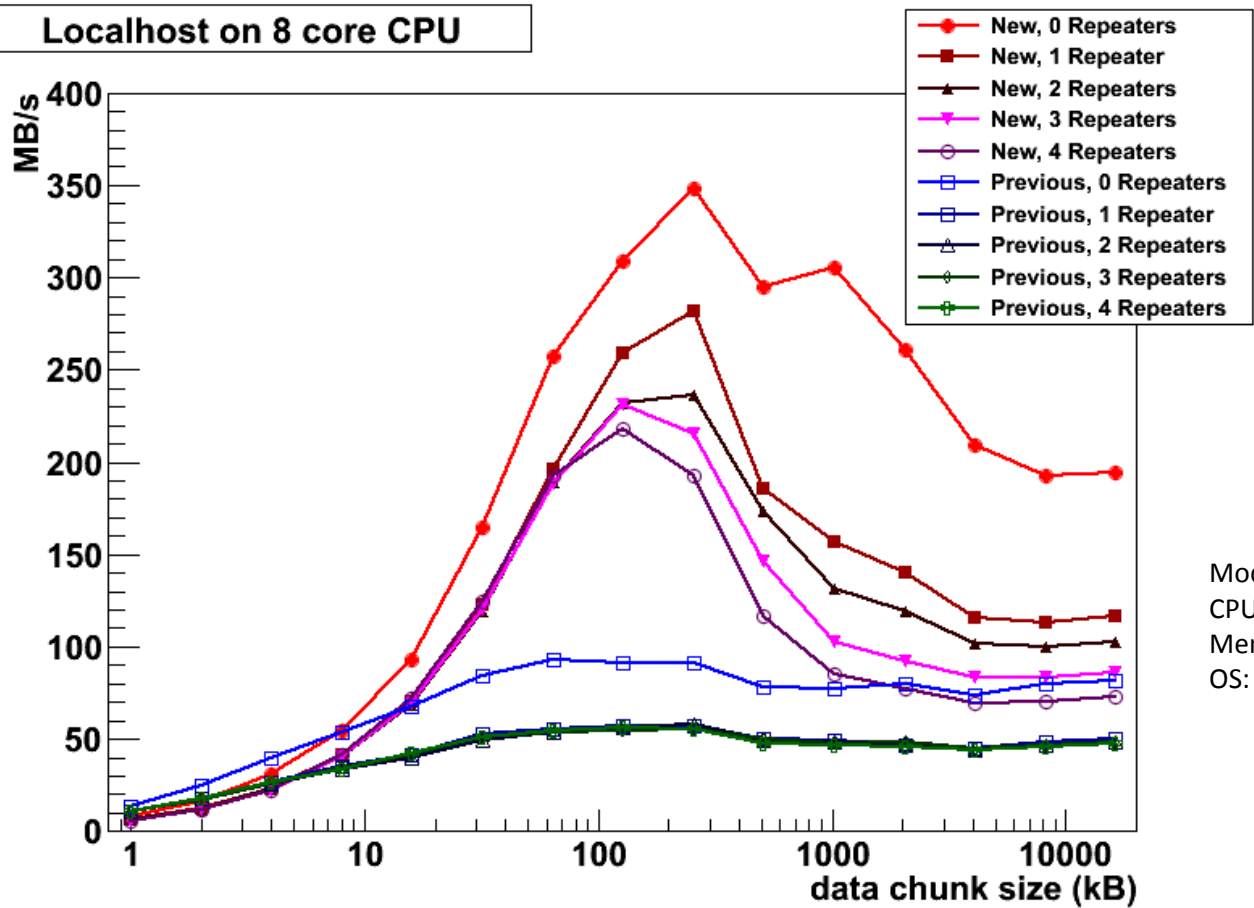


Localhost, Source - Repeater - Sink





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 1.1.0開発マニュアル

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

関連文献

- <http://daqmw.kek.jp/papers.html>

実習環境準備状況確認

- まだ準備していないかたがたの確認

和田さんのトークの部分

