



# E16実験へのDAQ-Middlewareの応用

濱田英太郎, 田中真伸, 内田智久, 池野正弘, 千代浩司, 四日市悟<sup>A</sup>, 小沢恭一郎, 森野雄平<sup>A</sup>, 高橋智則<sup>A</sup>, 中井恒<sup>BA</sup>, 川間大介<sup>A</sup>, 小原裕貴<sup>B</sup>

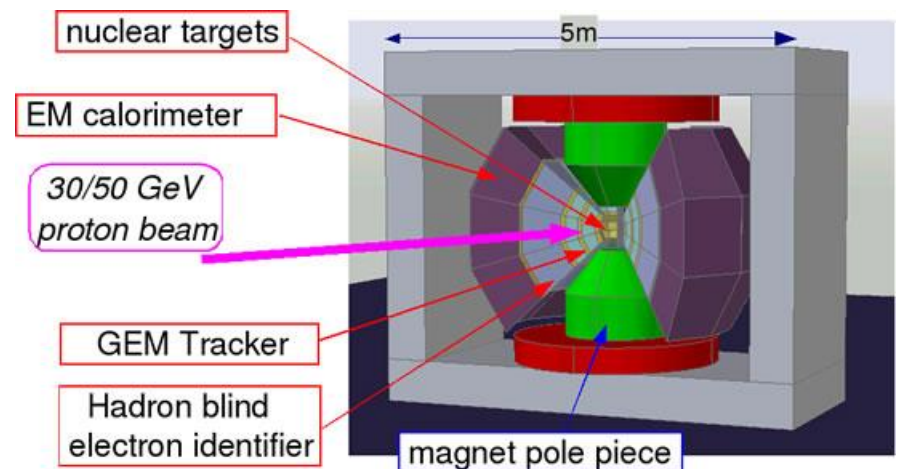
KEK素核研, 理研<sup>A</sup>, 東大理<sup>B</sup>

# 目次

- ・E16実験について
- ・DAQ-Middlewareについて
- ・作成したプロトタイプの紹介
  - 概要
  - 2つの特徴
  - DAQ性能
- ・まとめ

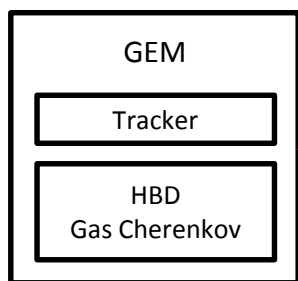
# E16実験

- 茨城県東海村 J-PARCハドロン実験施設で開始予定
- ベクトル中間子の質量変化現象を高統計・高分解能で測定



# 自分の担当

~96000 ch (56000+40000)



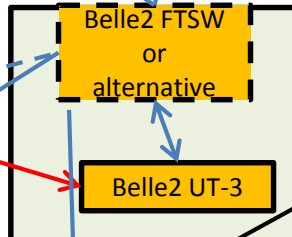
APV25-hybrd

Junction box

TRG-ADTX  
(ASD)

DAQ-Middlewareを用いて、  
データ収集、保存、解析を行なう  
ソフトウェアを作成

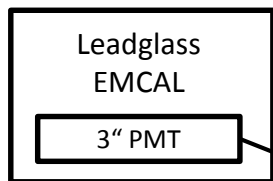
TRG-MRG  
(MUX, GTX)



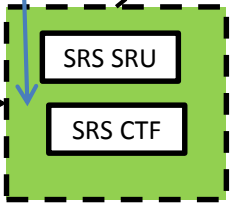
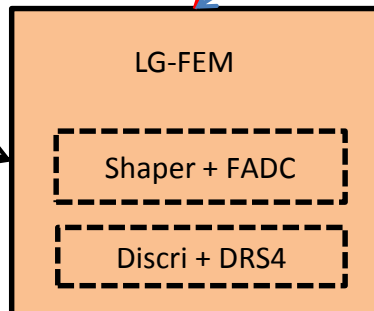
Network  
switch

DAQ  
PC

450MB/s



~1000 ch



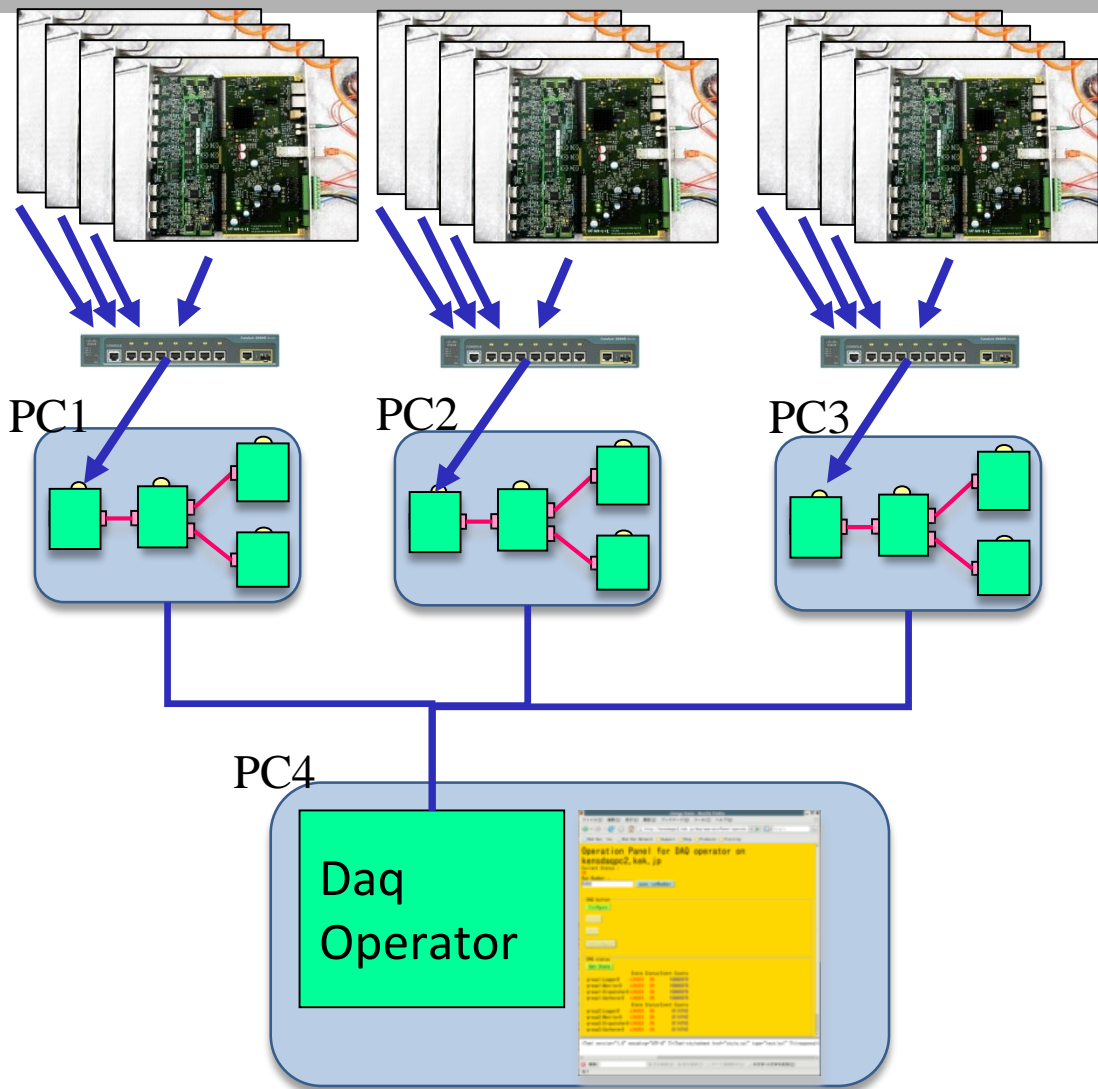
DTC-LINK

- Data flow
- Trigger primitive
- Global clock, global trigger

# DAQ-Middleware ネットワーク分散

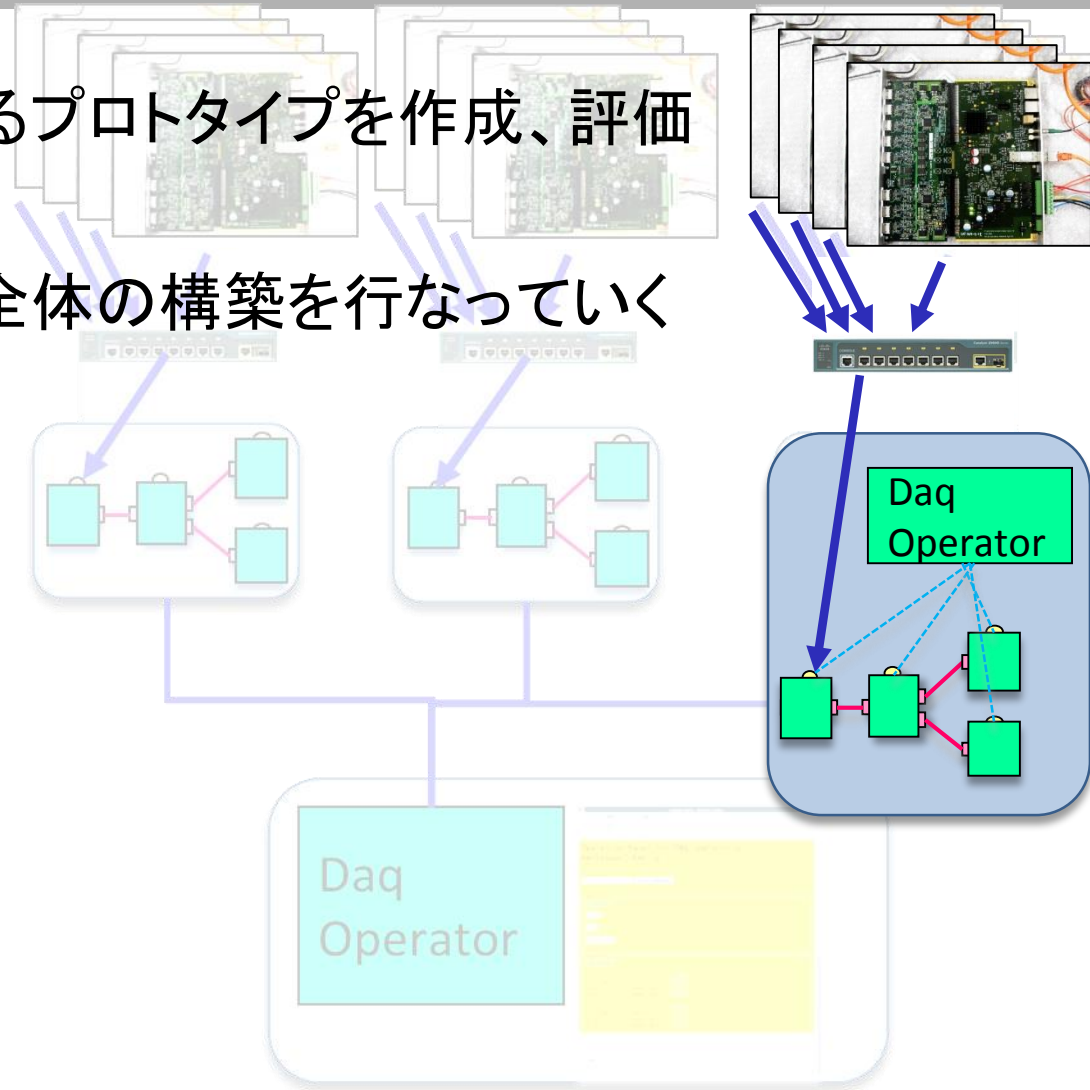
## ネットワーク分散

分散して配置されたノード同士が連携して通信するネットワークを形成

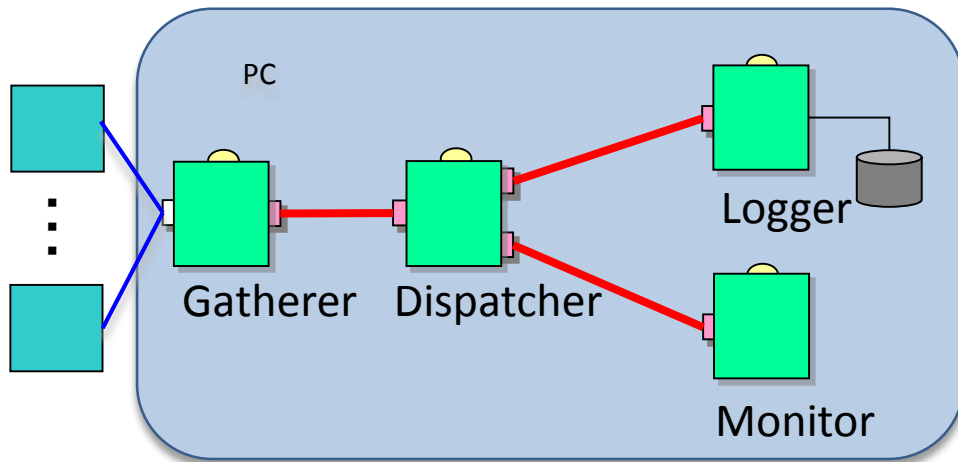


# E16実験 DAQソフトウェア開発状況

- PC1台を用いて構成されるプロトタイプを作成、評価
- プロトタイプを利用して、全体の構築を行なっていく



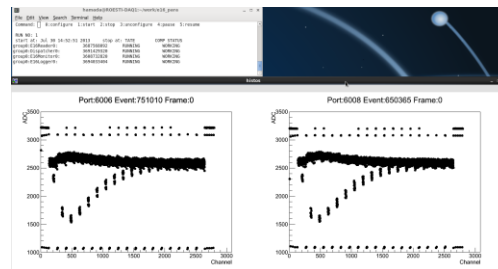
# E16実験用プロトタイプ 概要



1リードアウトモジュール  
から送られるデータ

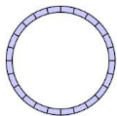
	平均	最大値
1イベントあたりの データサイズ	14kB	45kB
トリガーレート	1kHz	2kHz

- Gatherer :複数のリードアウトモジュールからデータを読み込み
- Dispatcher :Gathererから受け取ったデータを、LoggerとMonitorに送信
- Logger :全データをPCのハードディスクに保存
- Monitor :一定期間ごとにデータをモニタリング

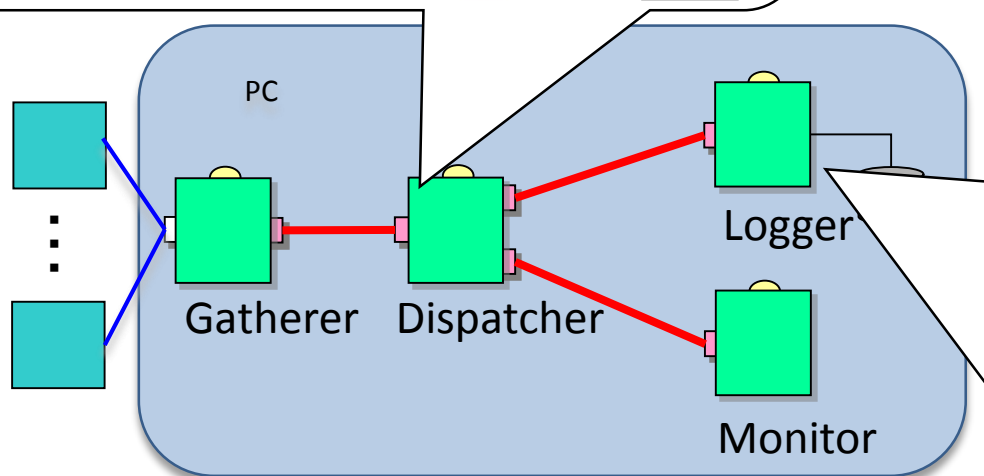
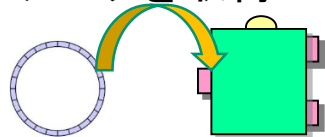


# 特徴1 リングバッファの拡大

①受信したデータは、リングバッファに格納



②プログラムがバッファからデータを取得



**Loggerコンポーネントでリングバッファが小さい場合**

①ハードディスクの書き込みにレイテンシが発生するときがある  
(しばらく書き込めない)

②書きこめない間、処理が進まない

③リングバッファにデータが溜まっていき、あふれてしまうことがある

④ソフトウェア全体が適切に動作しない

リングバッファを大きくする(最新の追加機能)

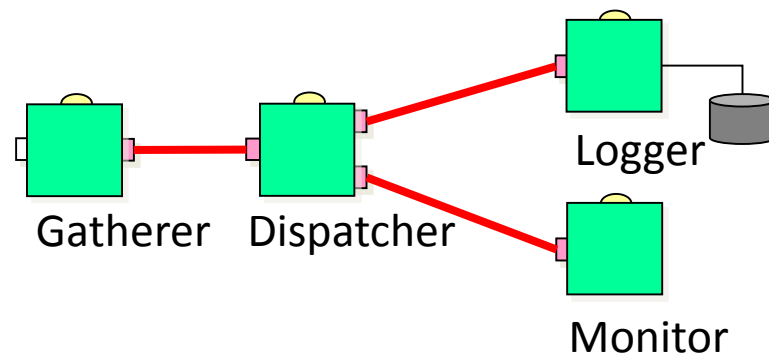
→HD書き込みのレイテンシの問題を解決



# 特徴2 コンポーネント間通信処理オーバーヘッドの軽減

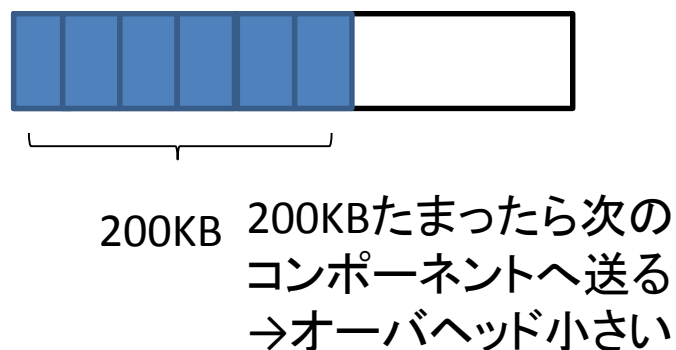
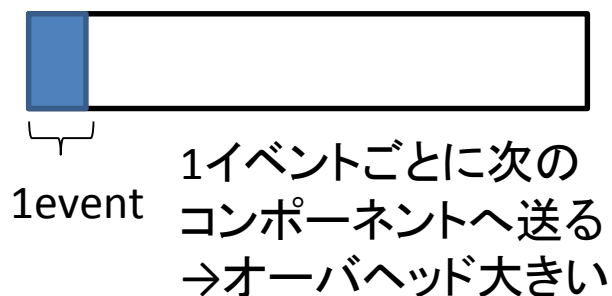
## オーバーヘッド

コンポーネント間でのデータのやり取りを行なう際の  
転送以外の処理  
多いとDAQ性能の低下につながる



影響を小さくする方法

- コンポーネント間のデータのやり取りを少なくする
- データをまとめる



# DAQ性能

## テスト環境

PC  
(エミュレータ)

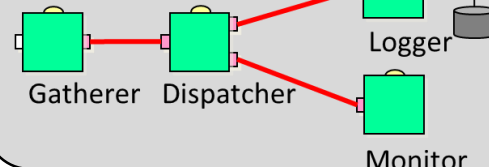
PC  
(DAQ PC)

CPU Intel(R) Xeon(R)  
X5650 @ 2.67GHz  
6コア

Memory 24GB

OS Scientific Linux 6.4

## DAQ-MW



## 1リードアウトモジュール から送られるデータ

	平均	最大値
1イベントあたりの データサイズ	14kB	45kB
トリガーレート	1kHz	2kHz

## テスト方法

エミュレータPCからDAQ PCにテストデータを送信

送ったデータの転送速度とプロトタイプの処理速度を比較

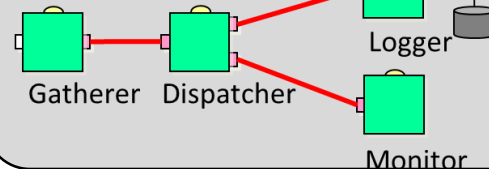
# DAQ性能

## テスト環境

PC  
(エミュレータ)

PC  
(DAQ PC)  
CPU Intel(R) Xeon(R)  
X5650 @ 2.67GHz  
6コア  
Memory 24GB  
OS Scientific Linux 6.4

## DAQ-MW

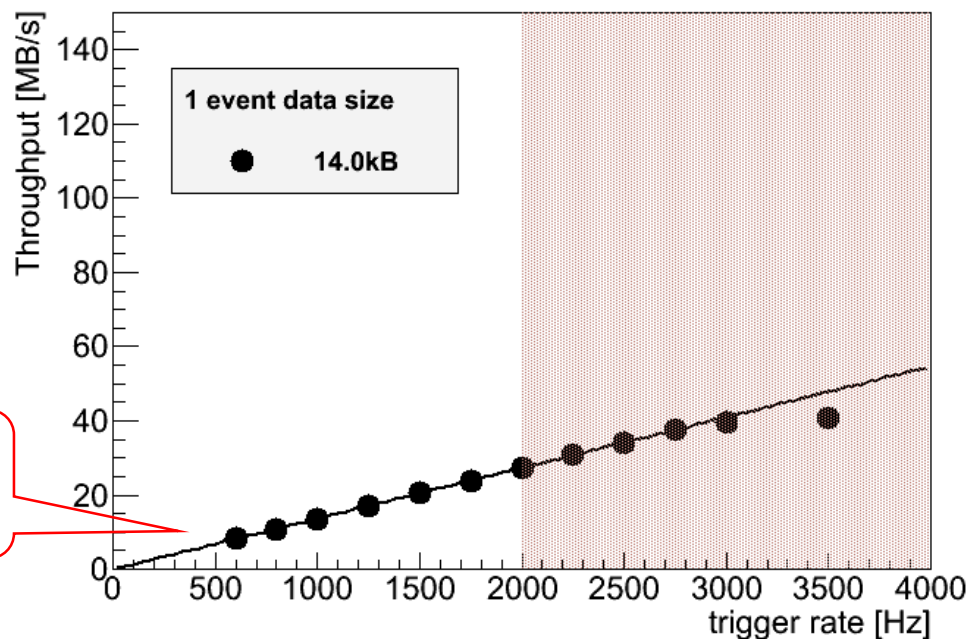


## 1リードアウトモジュール から送られるデータ

	平均	最大値
1イベントあたりの データサイズ	14kB	45kB
トリガーレート	1kHz	2kHz

2kHzまで、プロトタイプは送られた  
データ全てを処理している

## テスト結果



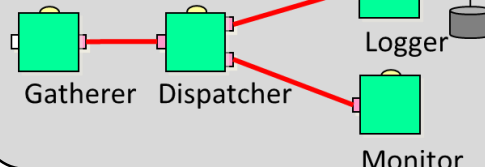
# DAQ性能

## テスト環境

PC  
(エミュレータ)

PC  
(DAQ PC)  
CPU Intel(R) Xeon(R)  
X5650 @ 2.67GHz  
6コア  
Memory 24GB  
OS Scientific Linux 6.4

## DAQ-MW

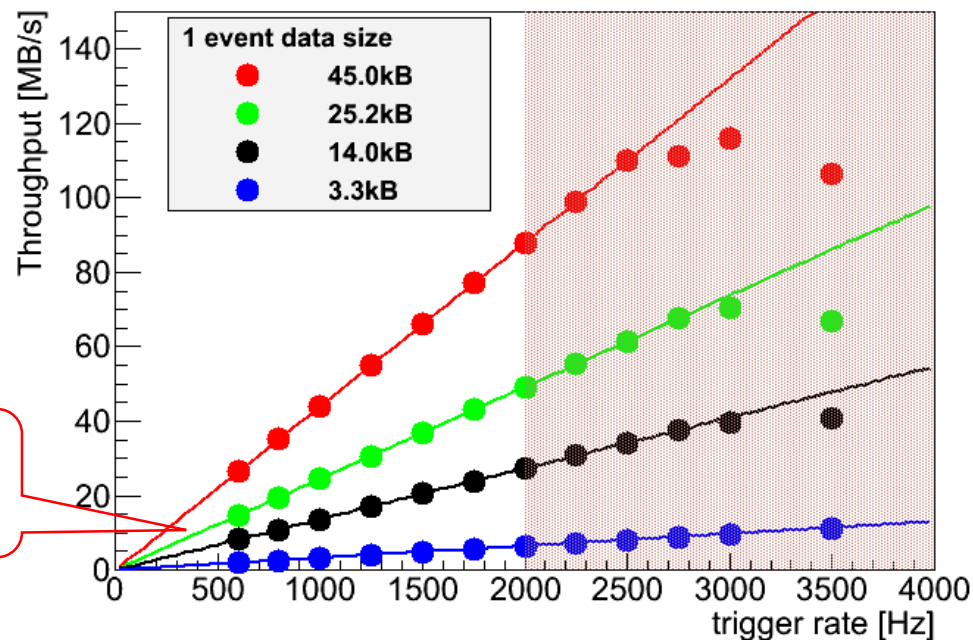


## 1リードアウトモジュール から送られるデータ

	平均	最大値
1イベントあたりの データサイズ	14kB	45kB
トリガーレート	1kHz	2kHz

2kHzまで、プロトタイプは送られた  
データ全てを処理している

## テスト結果



- E16実験DAQのソフトウェアを開発

現在は、1PCで動くプロトタイプを作成

- E16実験用プロトタイプ

特徴

- リングバッファの拡大
- コンポーネント間通信処理オーバーヘッドの軽減

性能

- リードアウトモジュール1台の場合、実験の想定された性能を満たしている

- 今後の予定

- 多くのエミュレータを用意した場合のテスト
- プロトタイプを用いて、システム全体の設計を行なっていく

# Backup

# DAQ性能

## テスト環境

PC  
(エミュレータ)

PC  
(エミュレータ)

CPU

PC  
(DAQ PC)

Intel(R) Xeon(R)  
X5650 @ 2.67GHz  
**6コア**

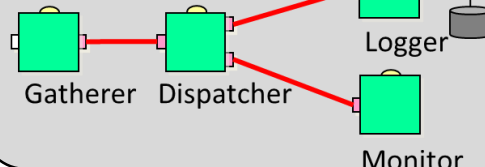
Memory

24GB

OS

Scientific Linux 6.4

## DAQ-MW

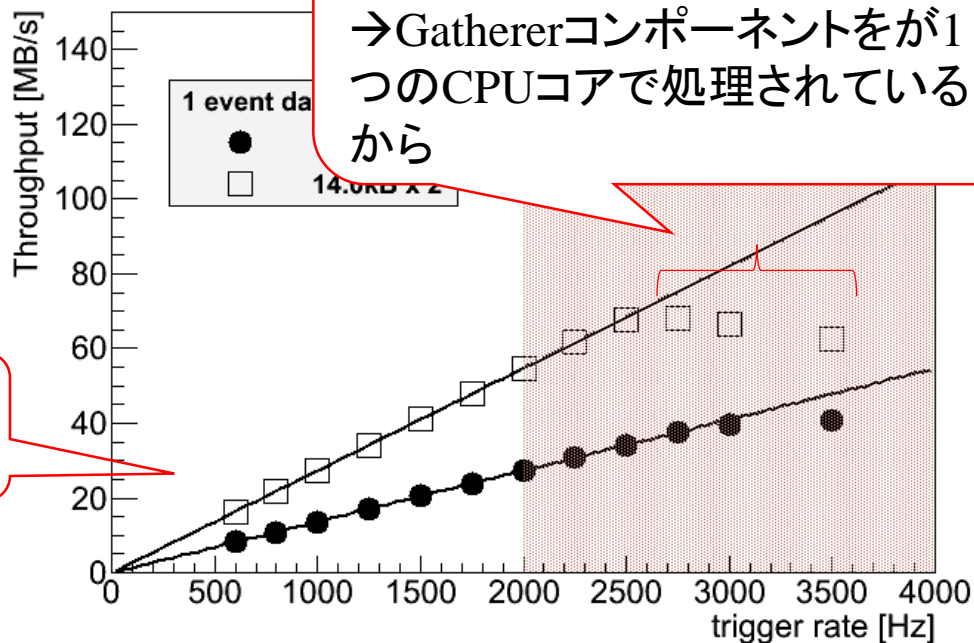


## DAQ PCに求められる性能

	平均	最大値
1イベントあたりのデータサイズ	14kB	45kB
トリガーレート	1kHz	2kHz

2kHzまで、プロトタイプは送られたデータ全てを処理している

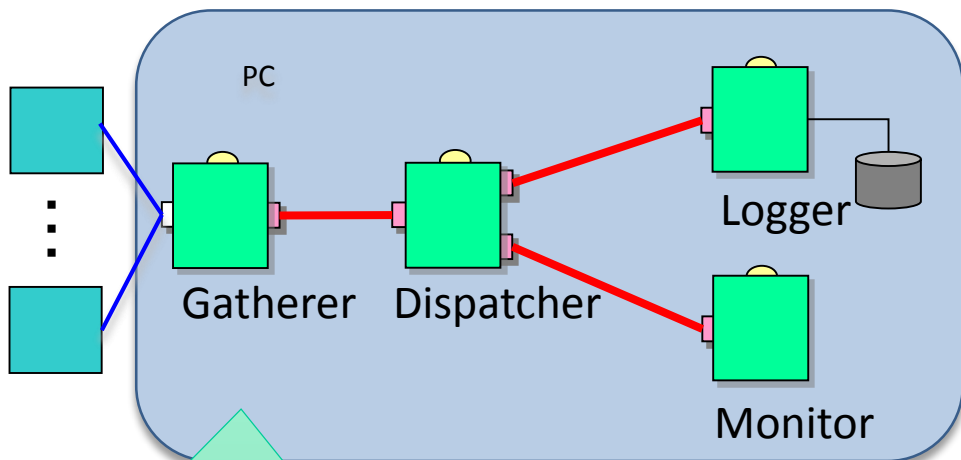
## テスト結果



スケーラブルの変化ではない

→Gathererコンポーネントをが1つのCPUコアで処理されているから

# コンポーネントの考えている構成

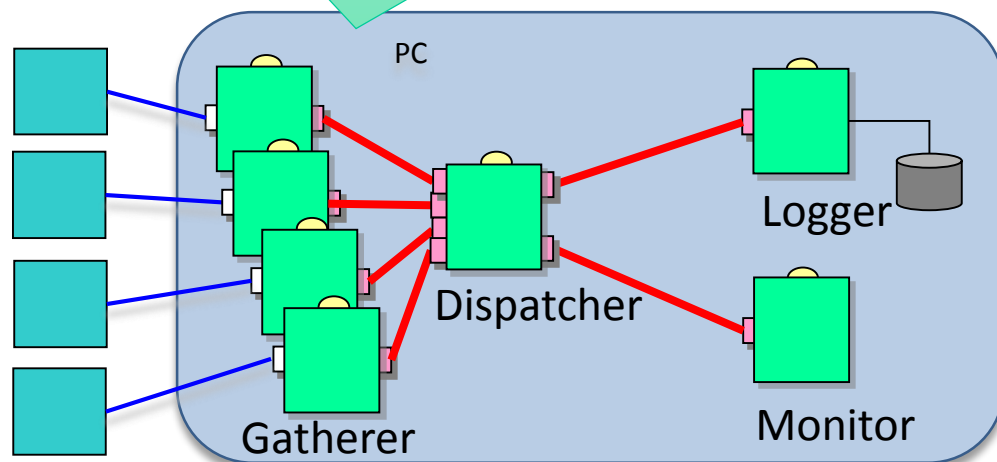


## 考えている構成

Gatherコンポーネントを1つのCPUコアで処理を行なっている  
(2つ以上のモジュールを読む場合でも)

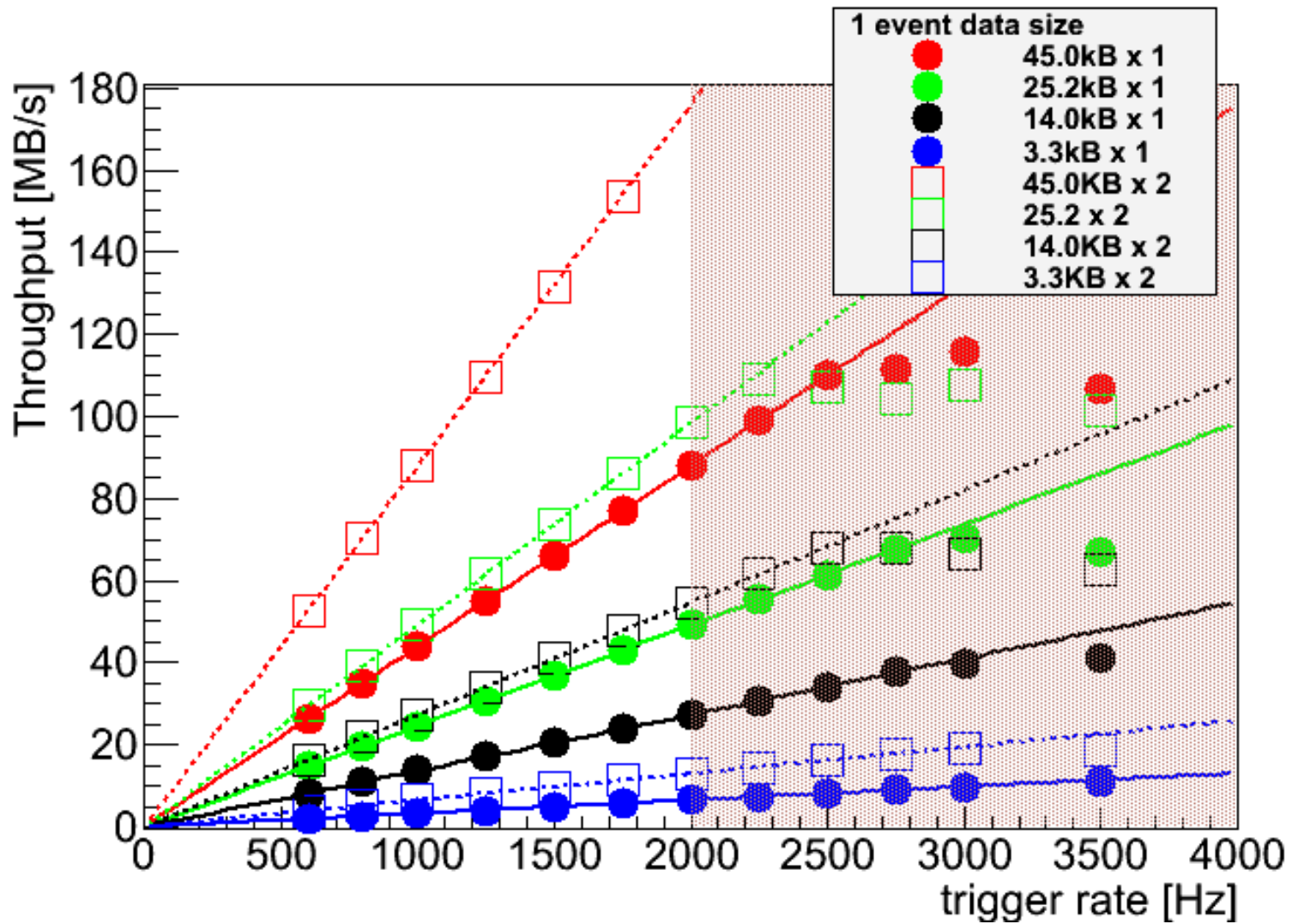
今

Gatherコンポーネントを1つのCPUコアで処理を行なっている  
(2つ以上のモジュールを読む場合でも)





# 全テスト結果

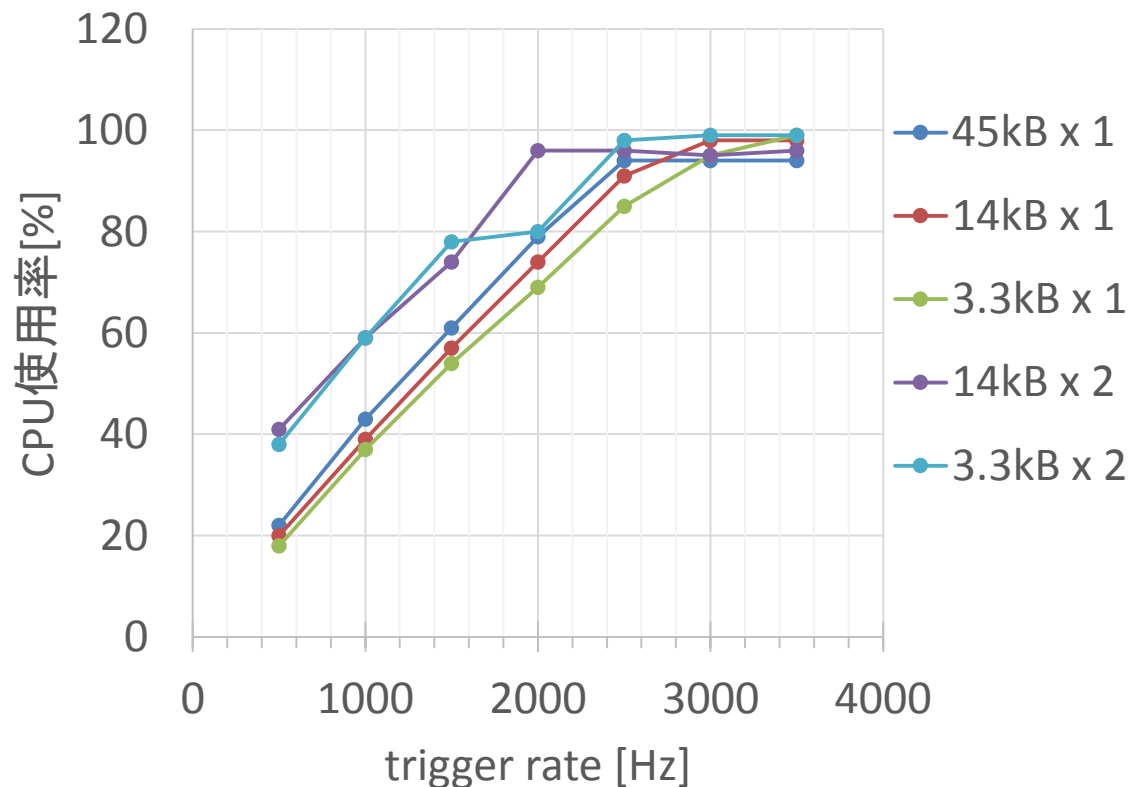


# PC使用率

TOPコマンドを用いて、GathererコンポーネントのCPU使用率を取得

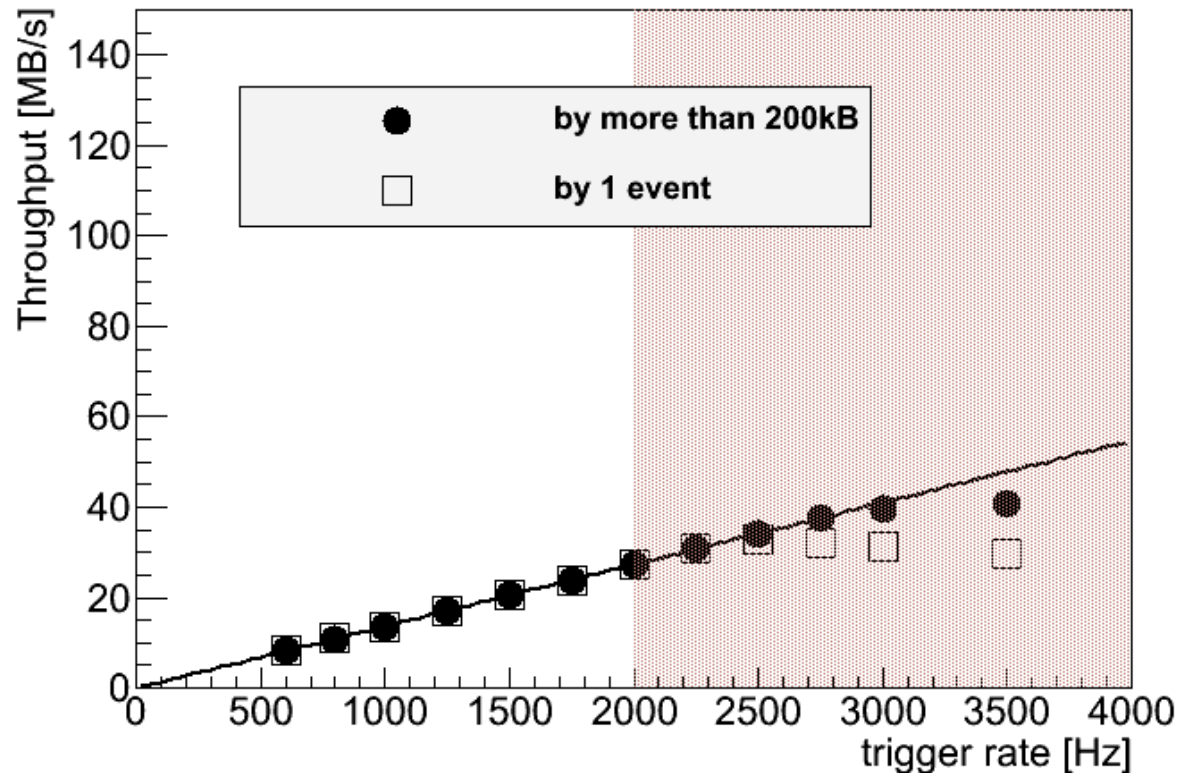
trigger rateをあげるにつれてCPU使用率も上昇

1イベントあたりのデータ量が上がるにつれてCPU使用率も上昇



# overhead 性能テスト

1イベントあたりの  
データ量  
14kBの場合



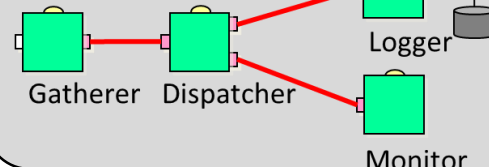
# DAQ性能

## テスト環境

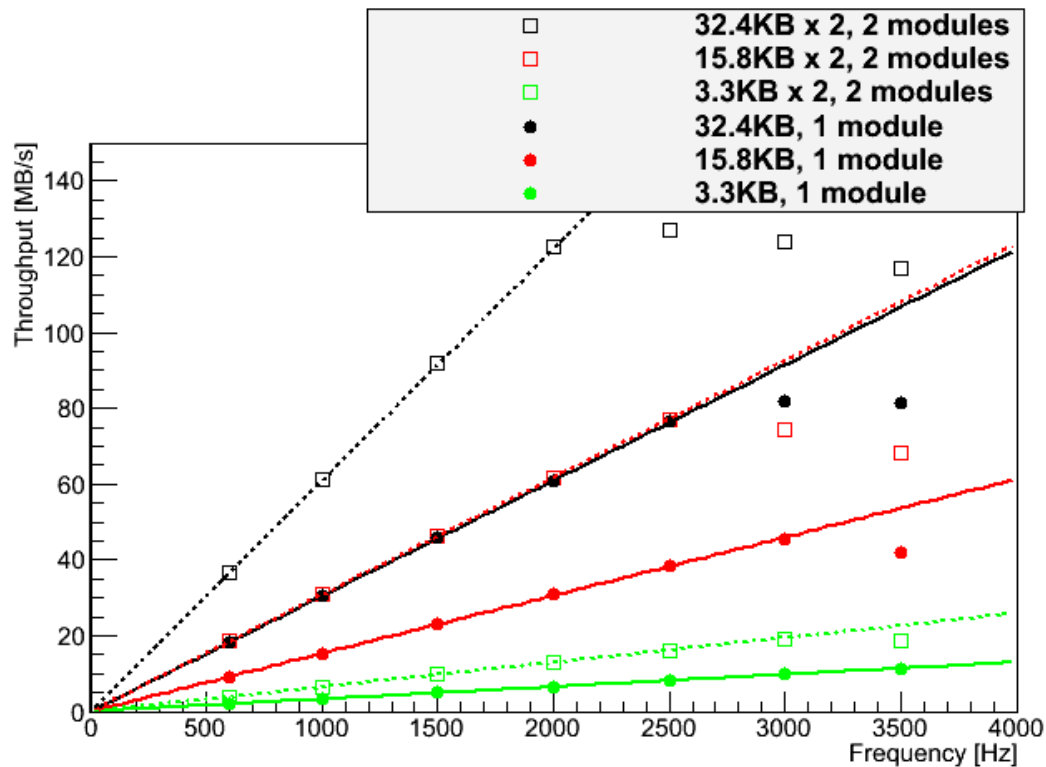
PC  
(エミュレータ)

PC  
CPU Intel(R) Xeon(R)  
X5650 @ 2.67GHz  
**6コア**  
Memory 24GB  
OS Scientific Linux 6.4

## DAQ-MW



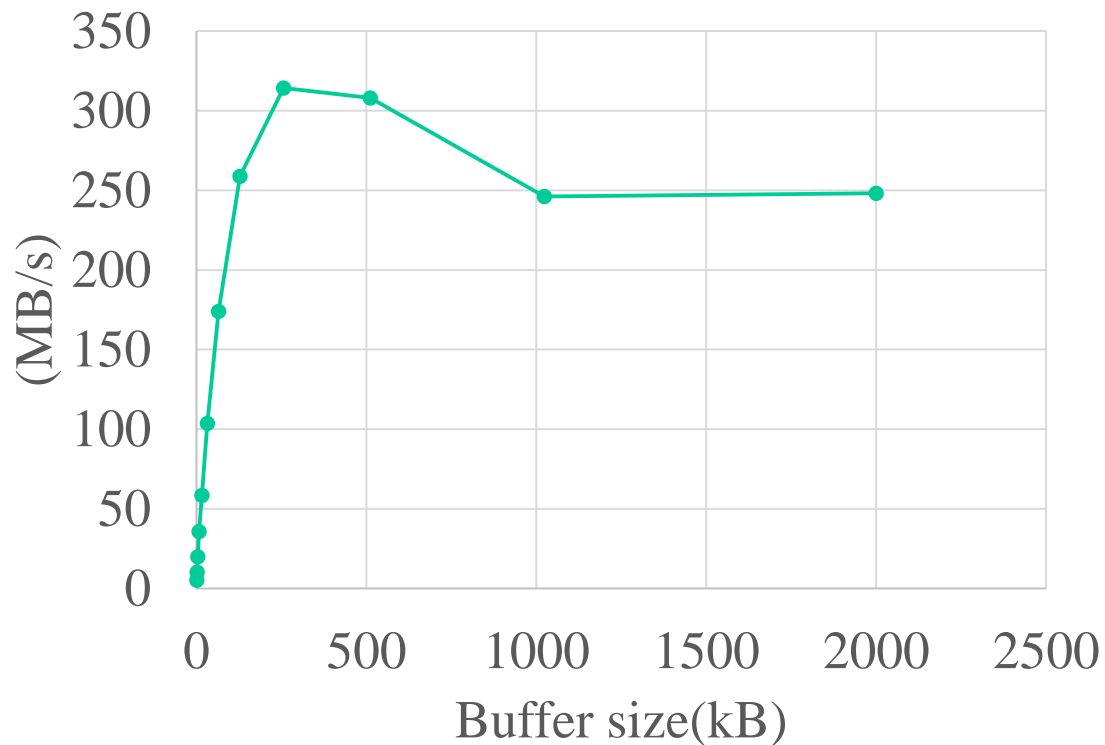
- 1イベントのデータを16回かけて送信
- 実験では、1kHz~2kHz  
この間では、問題ない
- 2modulesにすると、性能が落ちる  
→Gatherが1コアで処理を行なっているから



Source  
Dispatcher  
Sink  
Sink  
の4つのコンポーネント

SourceはDispatcherに、データを送ります。  
Dispatcherは、exampleと変わっていません。2つのSinkにデータを送ります。  
Source,Sinkは千代さんが以前、性能テストで使ったコンポーネントと同じことを行なっています。

各BufferSizeを5分間測定したときの結果です。



Source  
Dispatcher  
Sink  
Sink  
の4つのコンポーネント

SourceはDispatcherに、データを送ります。  
Dispatcherは、exampleと変わっていません。2つのSinkにデータを送ります。  
Source,Sinkは千代さんが以前、性能テストで使ったコンポーネントと同じことを行なっています。

各BufferSizeを30分間測定したときの結果です。  
これを5回行ないました。  
(色の違いは、何回目に測定したか、です)  
30分測定こともあり、ほとんど色にばらつきはありません  
この結果より、このコンポーネントの組み合わせの場合、最大約340MB/sのスループットで処理できることが分かりました。

