

DAQ-MWトレーニングコース2019

DAQシステム概要

広島工業大学

長坂 康史

nagasaka@cc.it-hiroshima.ac.jp

目的と達成目標

- 目的

- データ収集システムとは何かを知り、データ収集システムを構築するための知識を修得する

- 達成目標

- データ収集システムの役割を知り、設計することができる

目次

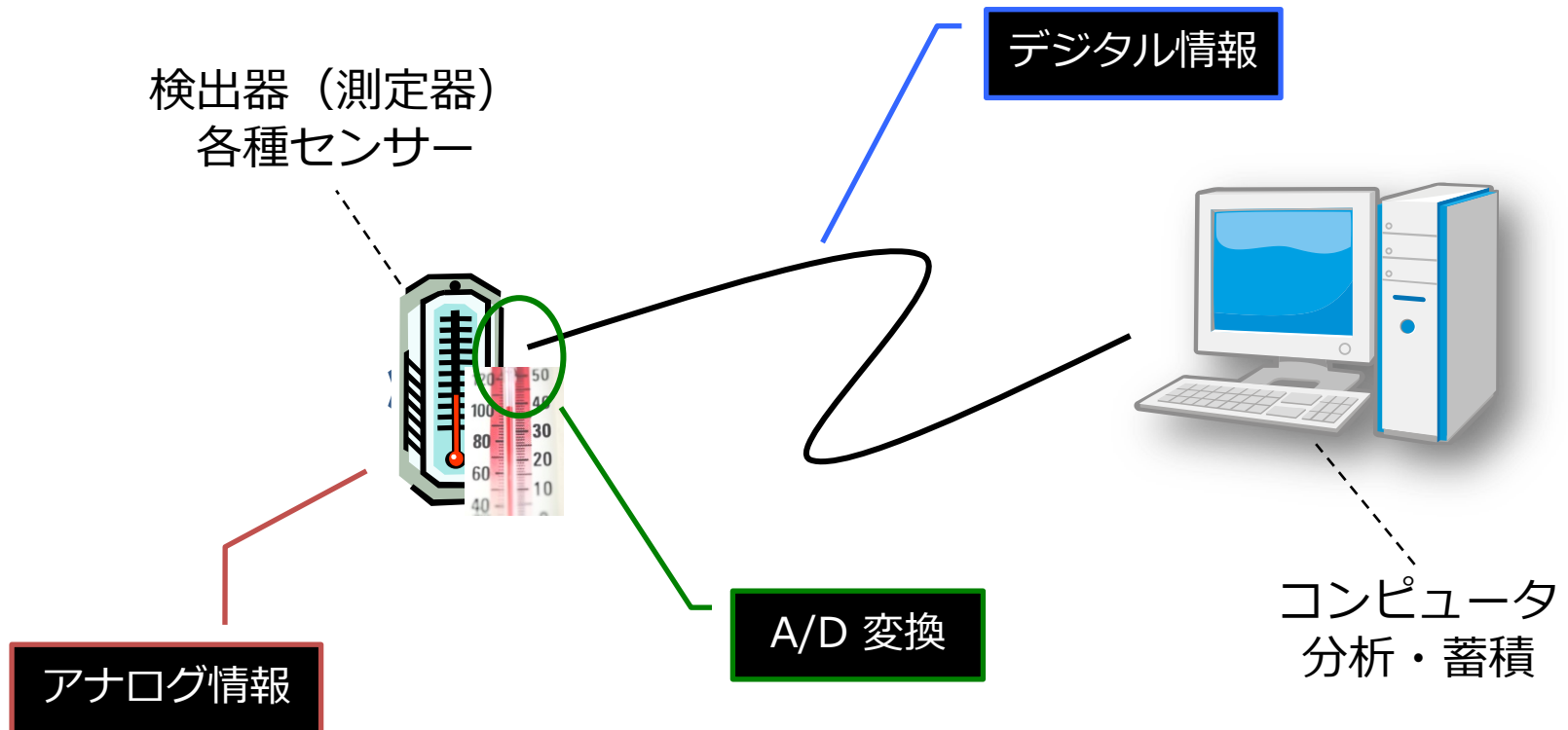
- データ収集システムとは
- データ収集システムの役割
- データ収集システムの概要
- データ収集システムの設計

- **データ収集システム (DAQシステム)**
 - DAQ : Data Acquisition System

データ収集システムの例

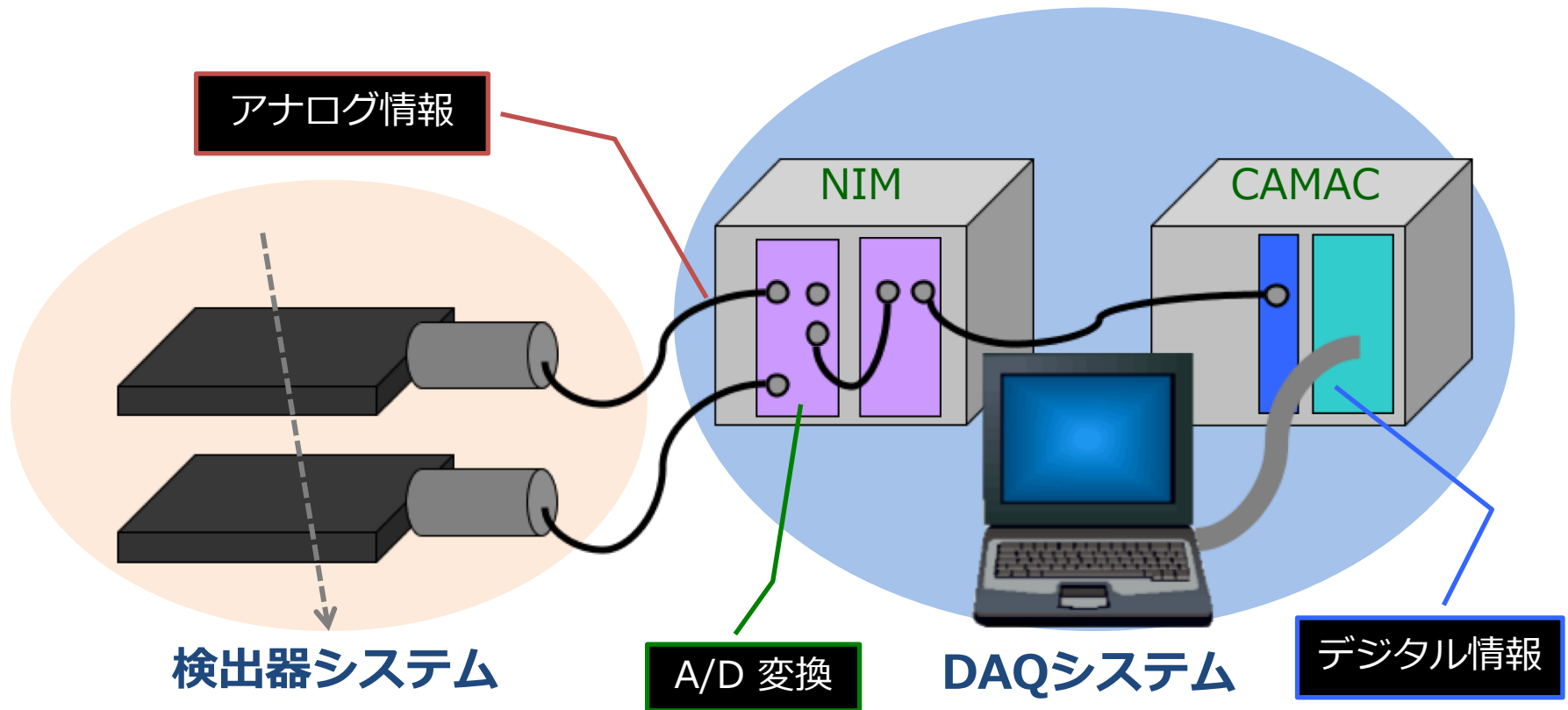
DAQシステムの例 1

- 小規模DAQシステムの例 1



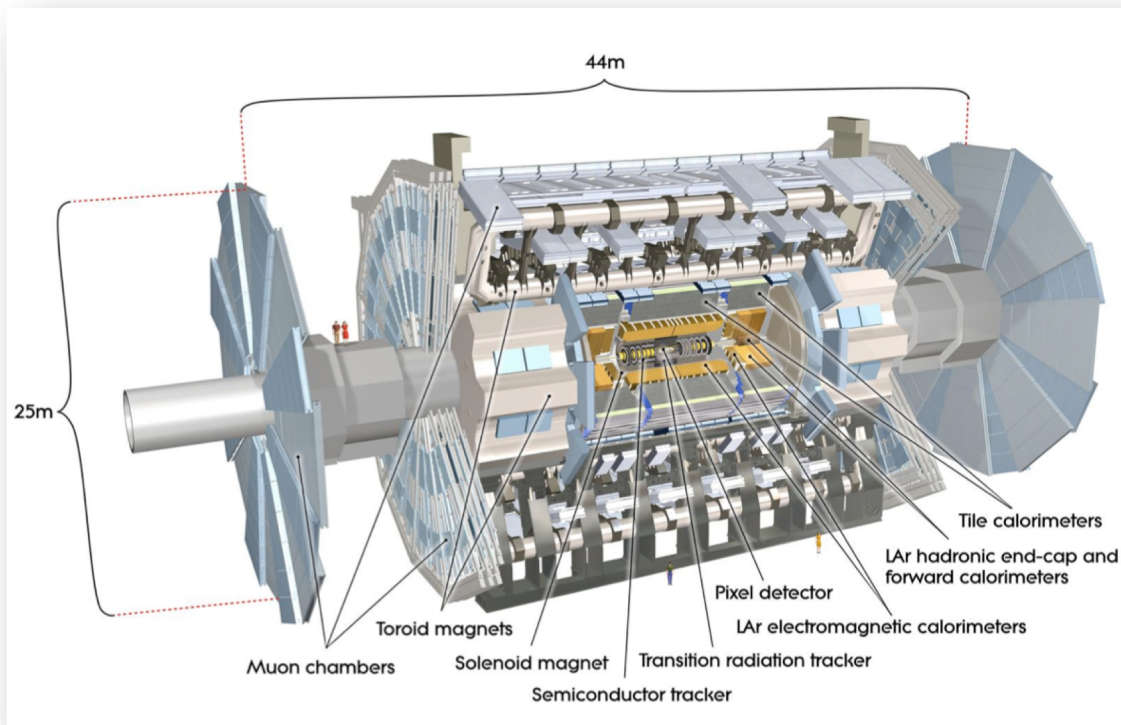
DAQシステムの例 2

- 小規模DAQシステムの例 2
 - 2枚のシンチレーション・カウンタを通過する宇宙線を検出してカウント



DAQシステムの例 3 (1)

- 大規模DAQシステムの例 (ATLAS実験)



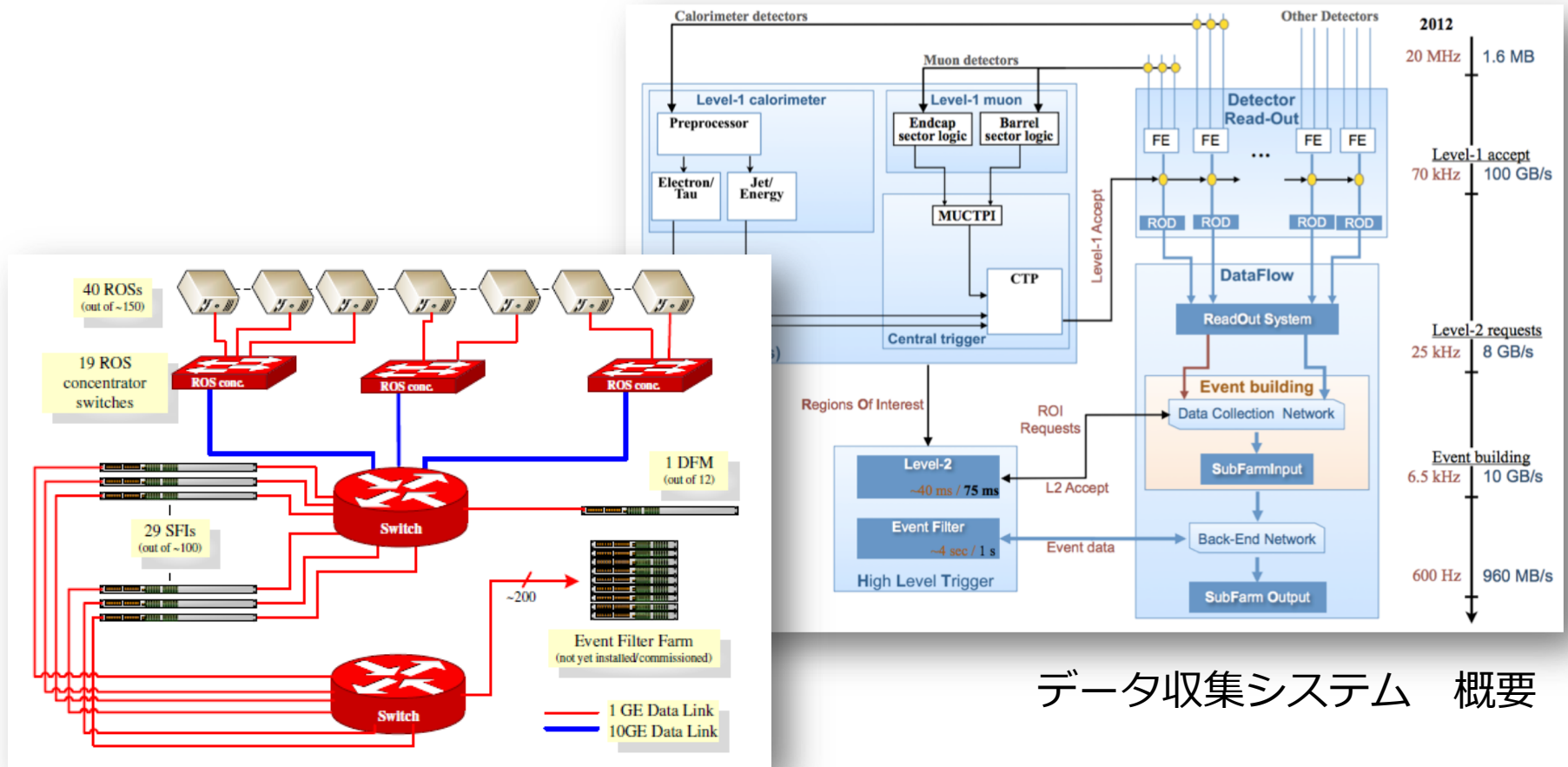
ATLAS実験測定器



ATLAS実験データ収集システム

DAQシステムの例 3 (2)

- 大規模システムの例 (ATLAS実験)



データ収集システム 構成

データ収集システム 概要

DAQシステムの例 3 (3)

The screenshot displays the ATLAS TDAQ SOFTWARE interface for Partition part_I2ef. The window title is "ATLAS TDAQ SOFTWARE - Partition part_I2ef" and the time is 1:01 AM. The interface is divided into several sections:

- File Commands:** Access Control, Settings, Logging Level, Help.
- Buttons:** Commit & Reload, Load Panels.
- Run Control State:** RUNNING (indicated by a green bar).
- Run Control Commands:** SHUTDOWN, BOOT, TERMINATE, INITIALIZE, UNCONFIG, CONFIG, STOP, START, HOLD TRG, RESUME TRG.
- Beam Stable:** A green indicator and buttons for Warm Start and Warm Stop.
- Run Information & Settings:** A table showing event processing statistics.
- Run Control Tree:** A hierarchical view of the system components, all in a RUNNING state. The selected component is L2SV-Segment-1:lxplus419.
- Component Details:** A diagram showing the internal structure of the selected component, including HW, PMG, and Infrastructure.
- Subscription criteria:** A section for filtering messages by severity (WARNING, ERROR, FATAL, INFORMATION) and expression.
- Message List:** A table for displaying system messages.

Level	Number	Rate
Lumi Block	2	
Level 1	2476639	0
Level 2	2476576	27.50 kHz
Event builder	24445	281.30 Hz
Event filter	2477	29.22 Hz
Recorded	0	0.00 mHz

TIME	SEVERITY	APPLICATION	NAME	MESSAGE
[Empty message list]				

データ収集システムの役割

科学の役割とDAQシステム

- **事実や経験から客観的な法則性を見出し、未来の生活に役立てる（備える）こと**
 - 客観的な法則
 - いつでも、どこでも、誰がやっても同じ結果
 - 法則性を見出すためには、過去の事実や経験の積み重ねが重要
- **データ収集（DAQ）システムは、事実を記録して法則性を見出すための一つの道具**

DAQシステムを必要とする状況

- コンピュータを利用したほうが良いシステム
 - 多くの時間、人手、お金を必要とする実験
 - 滅多に起こらない現象を辛抱強く待つ実験
 - 瞬間的に大量の情報が発生する実験
 - 可視化できても人間の目や耳ではとうてい認識不可



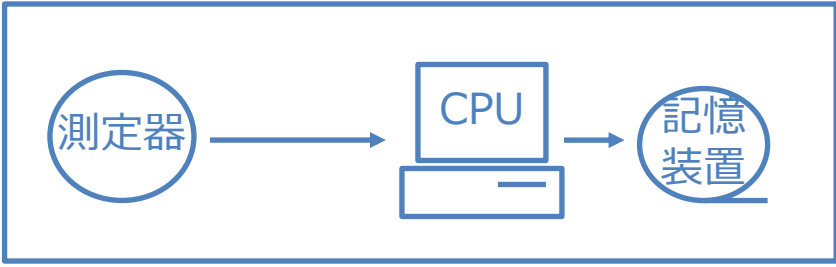
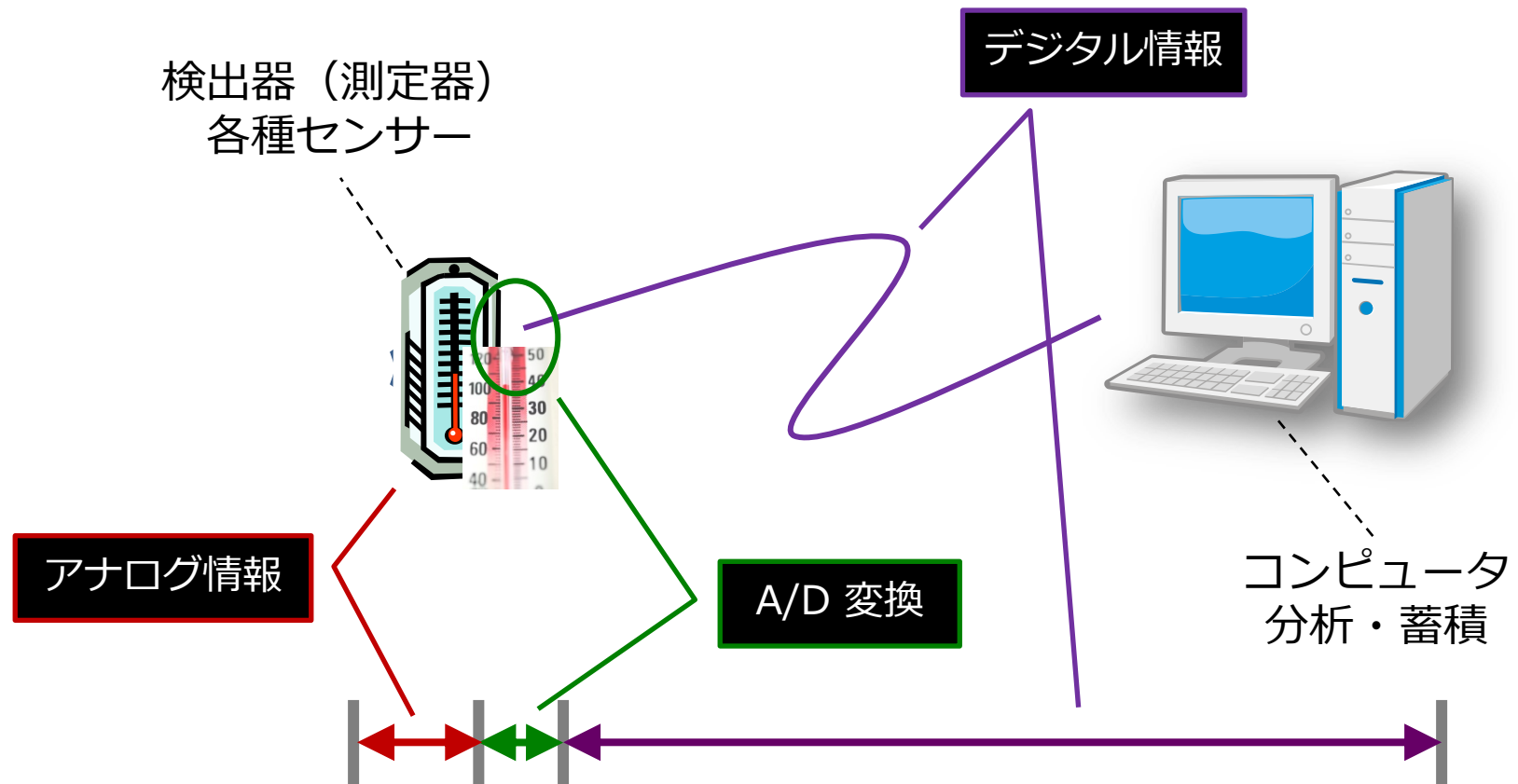
状態を計測し、
データを収集・記録し、
後から解析

DAQシステムの役割

- さまざまな場面で、ある状態を計測し、そのデータを**収集・記録**し、**伝えること**
 - 計測、収集したデータは、後で本人のみならず、他人も解析できることが必要
 - データではなく情報として伝えることが大切

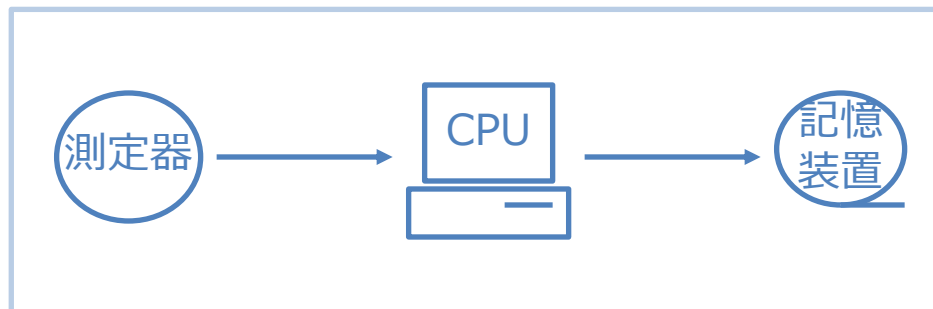
DAQシステムの概要

DAQシステムの基本



データ収集システム基本要素

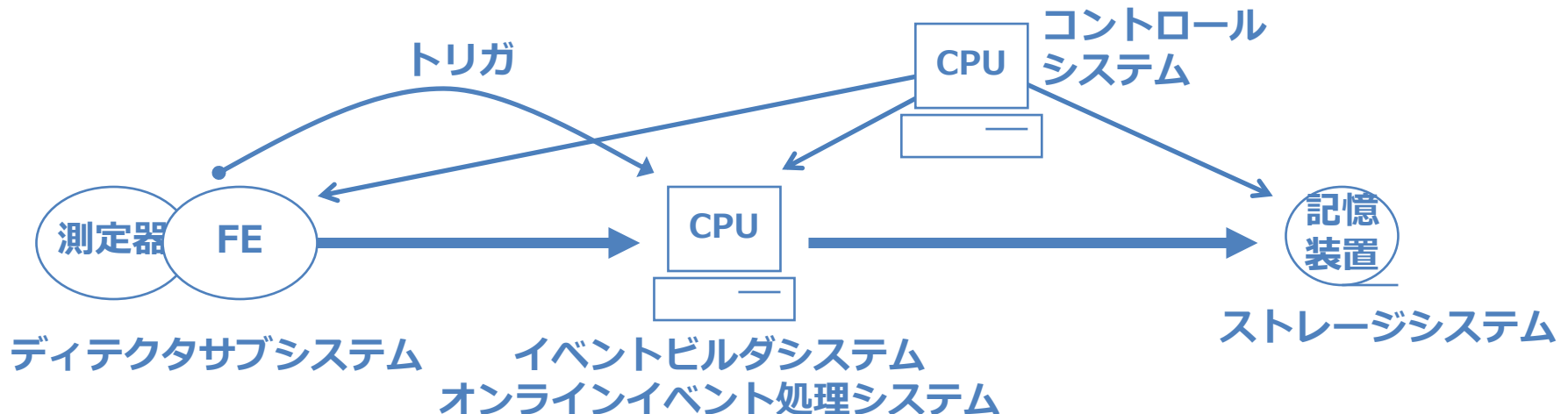
- **DAQ (Data Acquisition) System**
 1. **収集** (Acquisition)
 - A/D Conversion, Trigger, Event building
 2. **記録** (Storage)
 3. **制御** (Control)
 4. **モニタ** (On-line Monitor)
 5. **解析** (On-line Analysis)

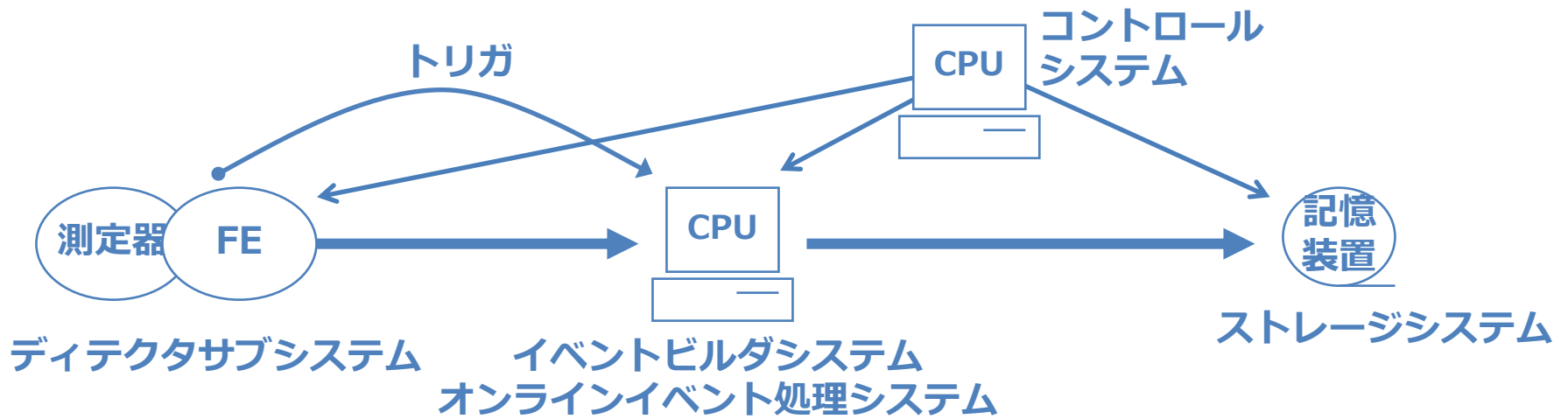


データ収集システム基本構成

- データ収集システムの構成要素
 - ディテクタサブシステム
 - フロントエンドシステム
 - イベントトリガシステム
 - イベントビルダシステム
 - オンラインイベント処理システム
 - ストレージシステム
 - コントロールシステム

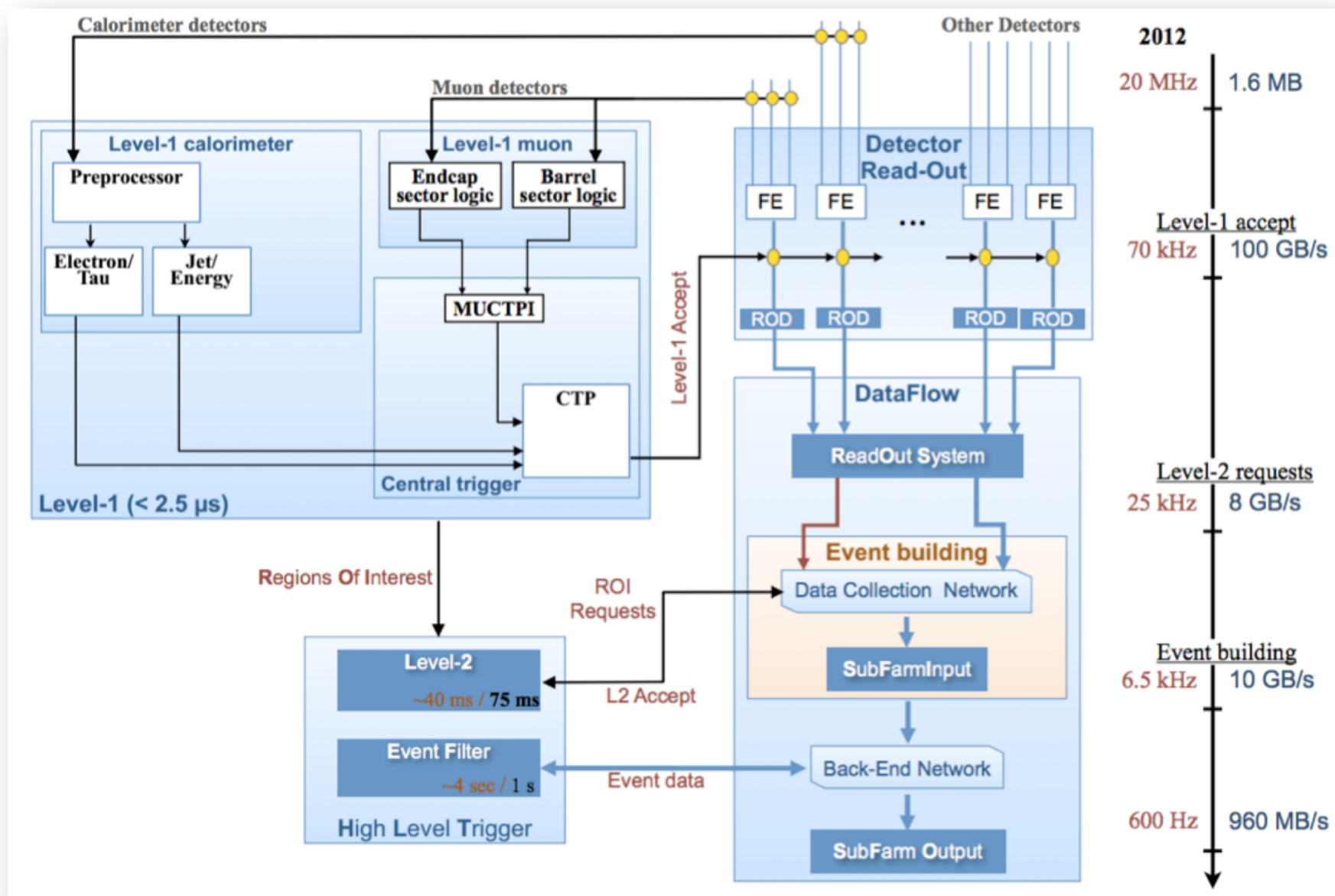
収集 ・ 記録 ・ 制御 ・ モニタ ・ 解析





データ収集システム イベントトリガシステム

ATLAS実験データ収集システムの例

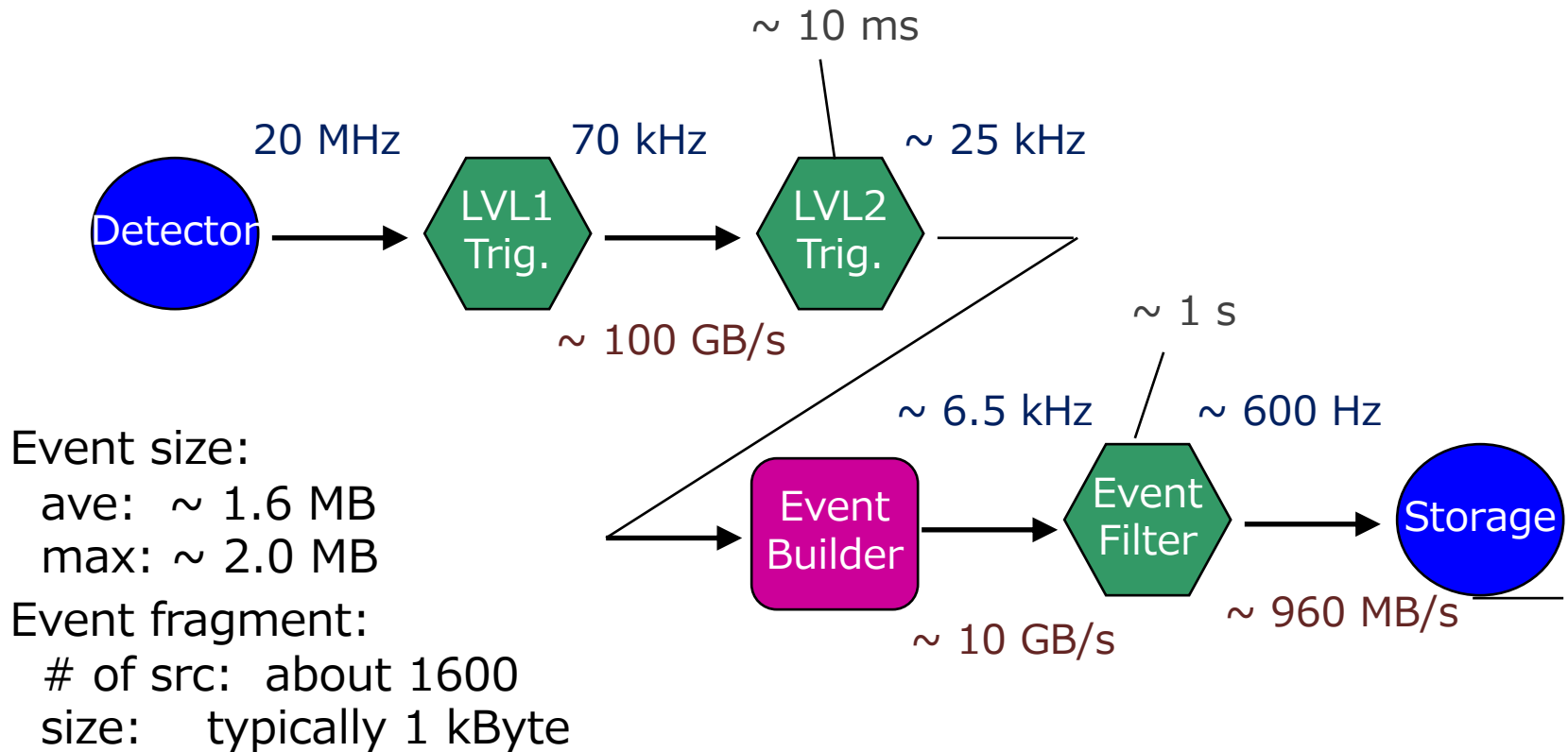


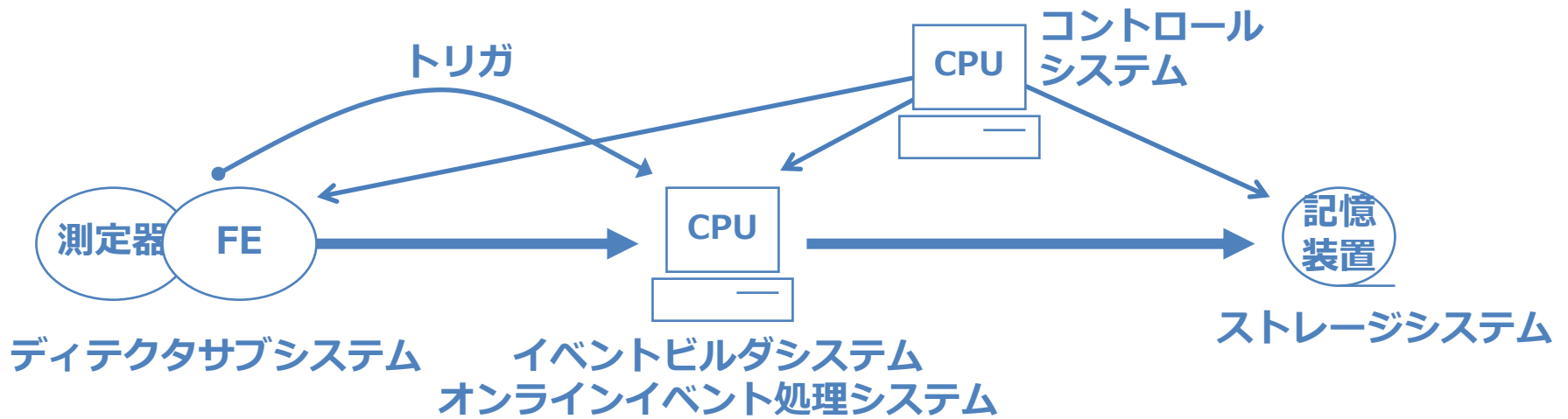
イベントトリガシステム

- イベントトリガシステム
 - 大量のデータを収集しながら処理し、不要なデータを記録しないようにするためのシステム
- トリガレベル
 - レベル1
 - いくつかのディテクタからの情報からトリガ生成
 - ハードウェア（デジタル回路）での処理
 - レベル2
 - いくつかのディテクタからの情報からトリガ生成
 - ソフトウェアでの処理
 - レベル3
 - イベントのすべての情報からトリガ生成
 - ソフトウェアでの処理

イベントトリガシステムの例

- ATLAS実験（初期）の例





データ収集システム イベントビルドシステム

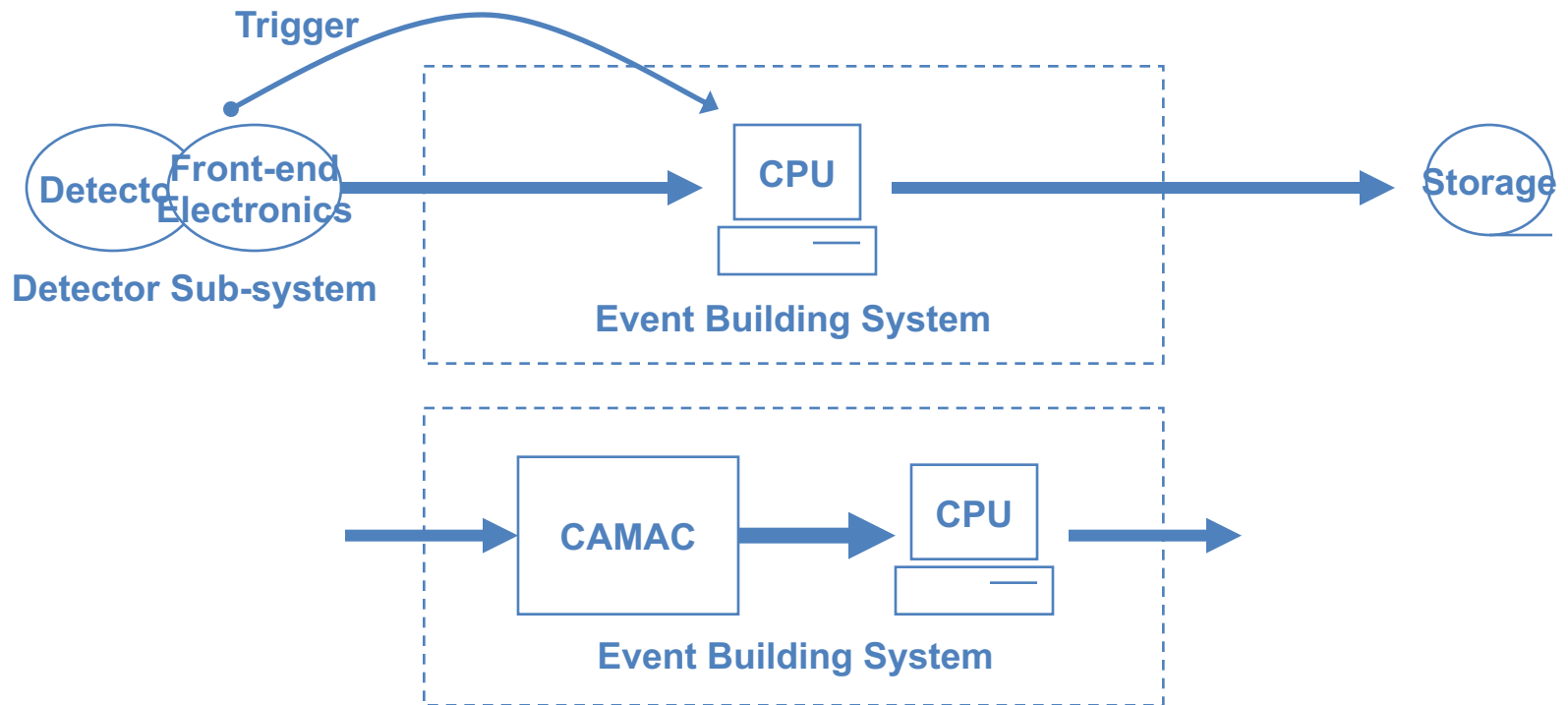
データ収集システムのタイプ

- 小規模システム (1970年代)
 - Single Detector - Single CPU システム
- 中規模システム (1980年代)
 - Multi Detector - Single CPU システム
- 大規模システム (1990年代以降)
 - Multi Detector - Multi CPU システム

小規模システム

- Single Detector - Single CPU システム

例 : Detector - CAMAC - CPU - Storage

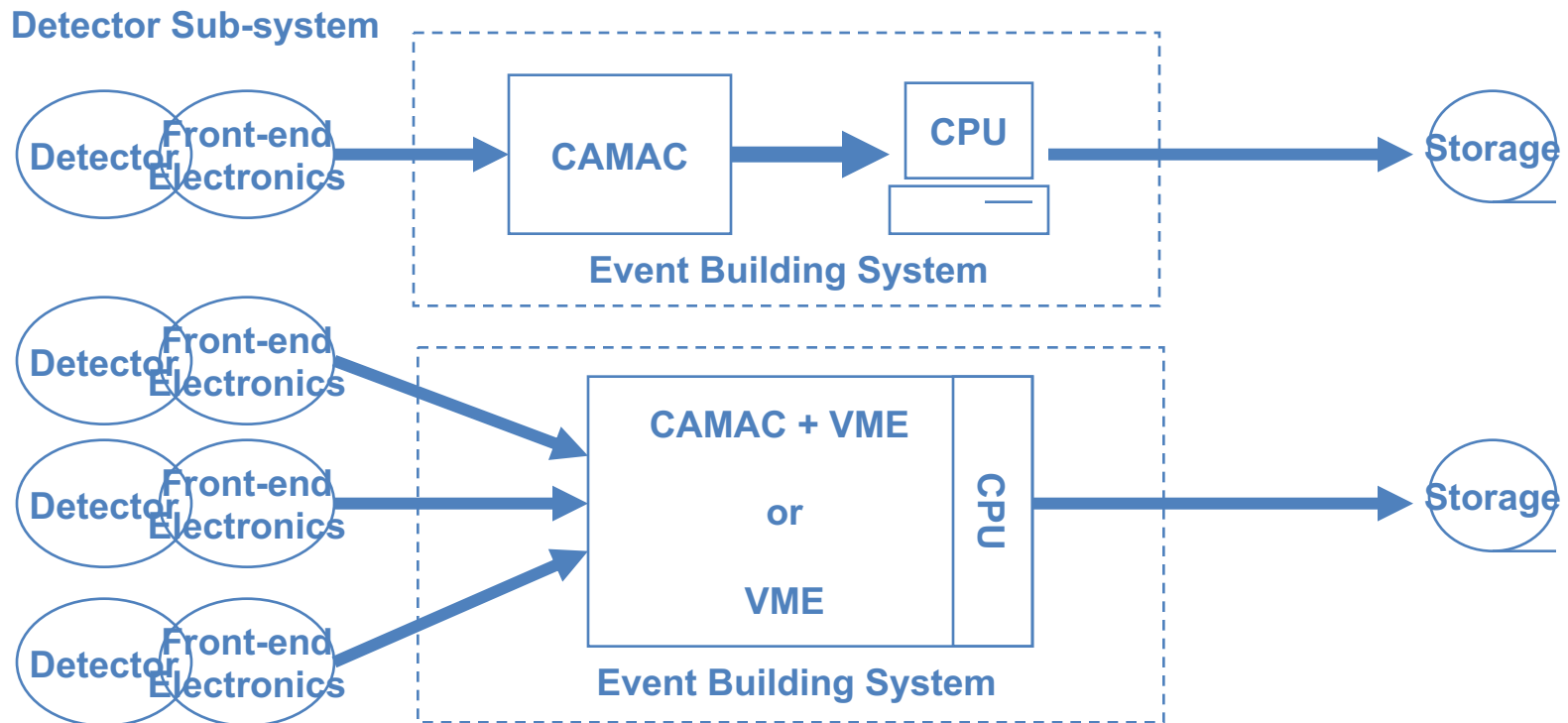


中規模システム 1

- Multi Detector - Single CPU システム

例： Detectors - CAMAC/Fastbus/VME

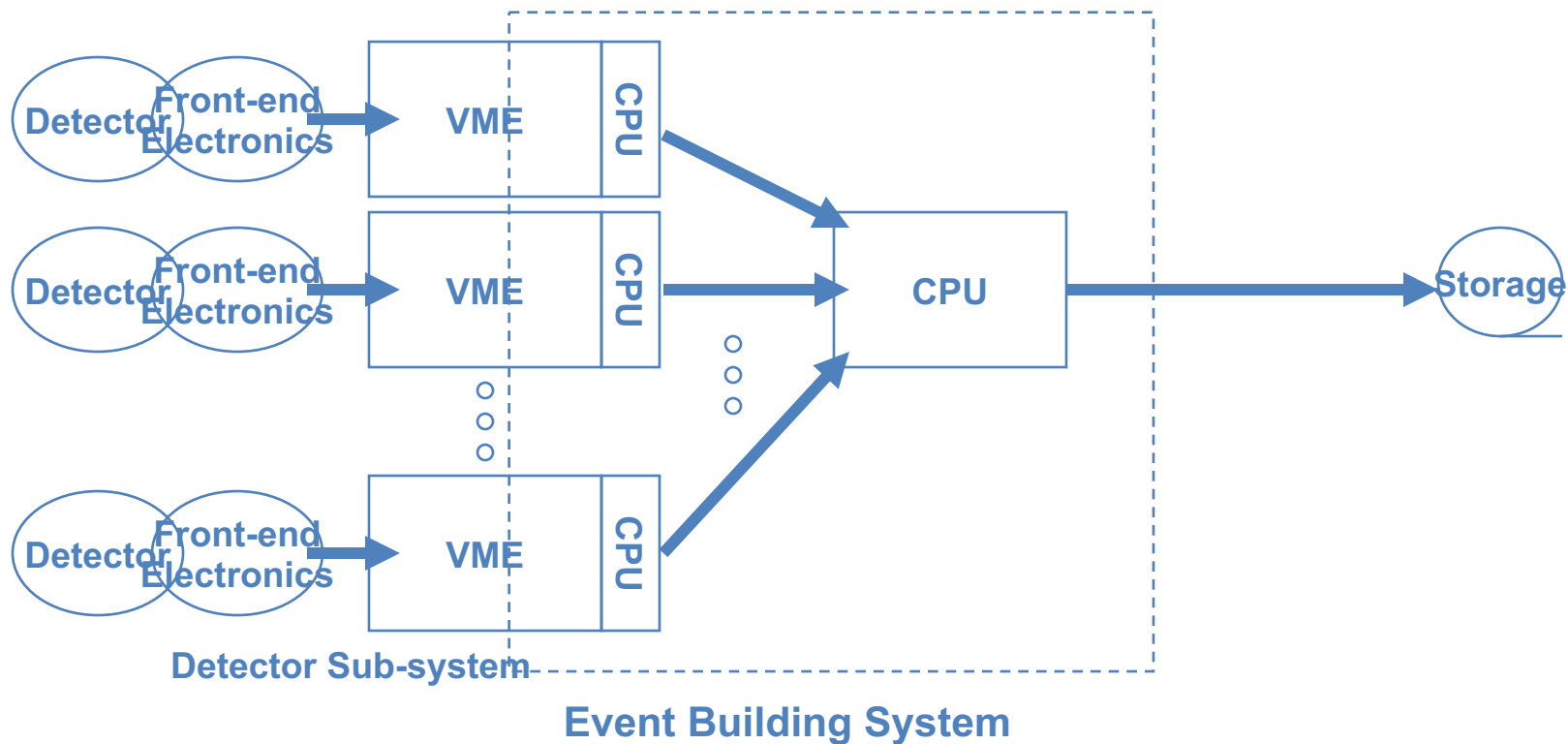
- CPU - Storage



中規模システム2

- Multi Detector - Multi CPU システム 1

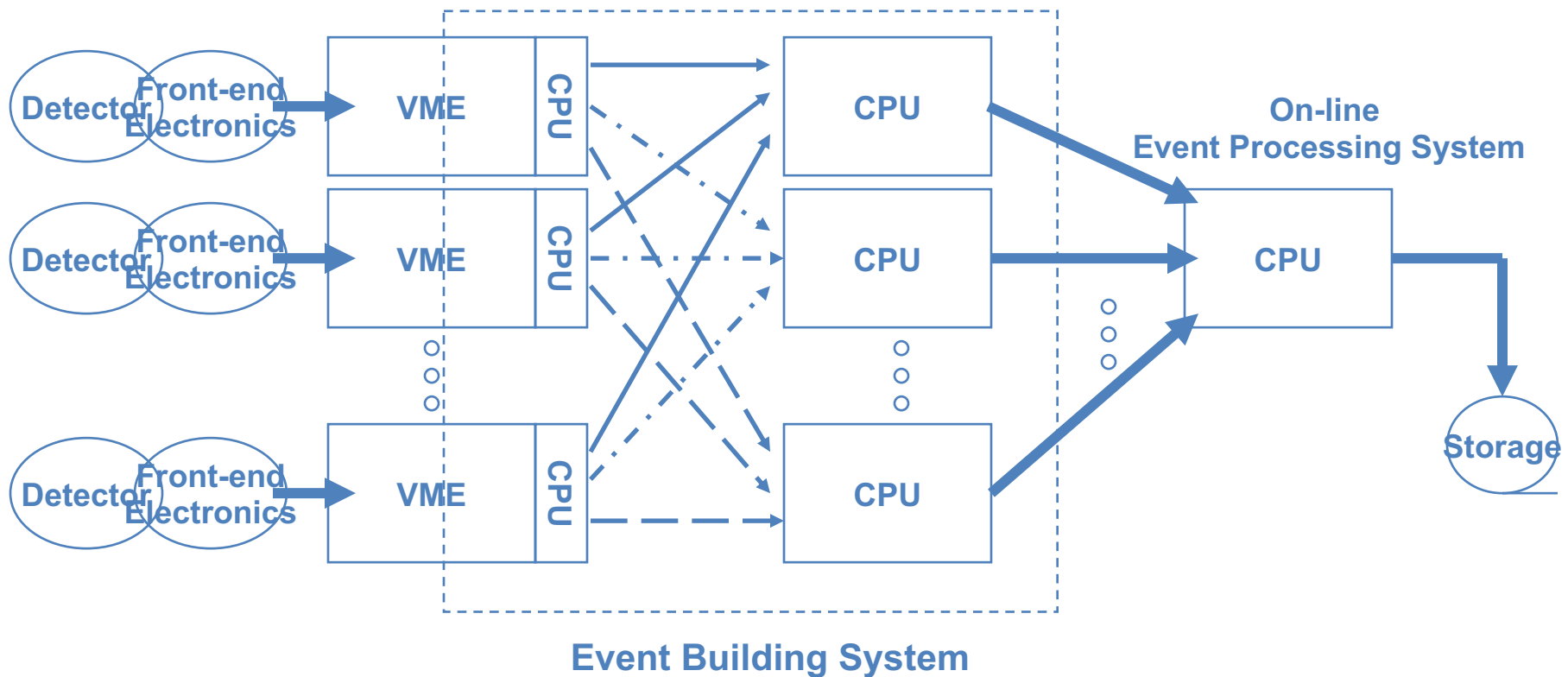
例： Detectors - CAMACs/Fastbuses/VMEs
- CPUs - Storage



大規模システム

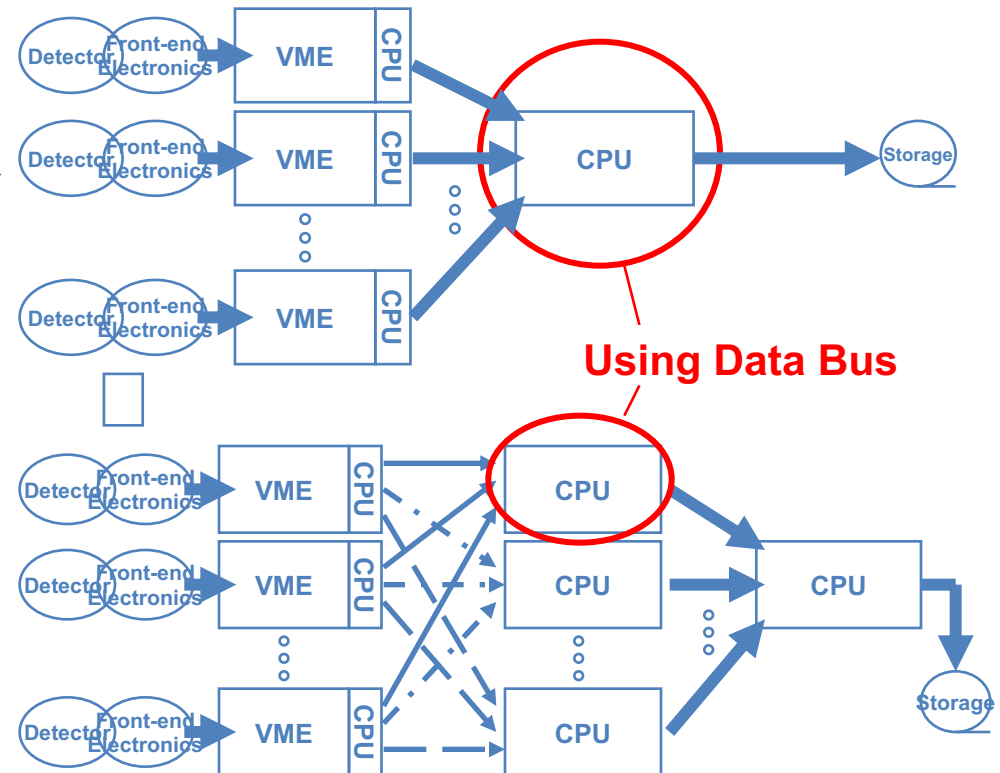
- Multi Detector - Multi CPU システム 2

例 : Detectors - CAMACs/Fastbuses/VMEs
- CPUs - Storage



中・大規模システムの問題点

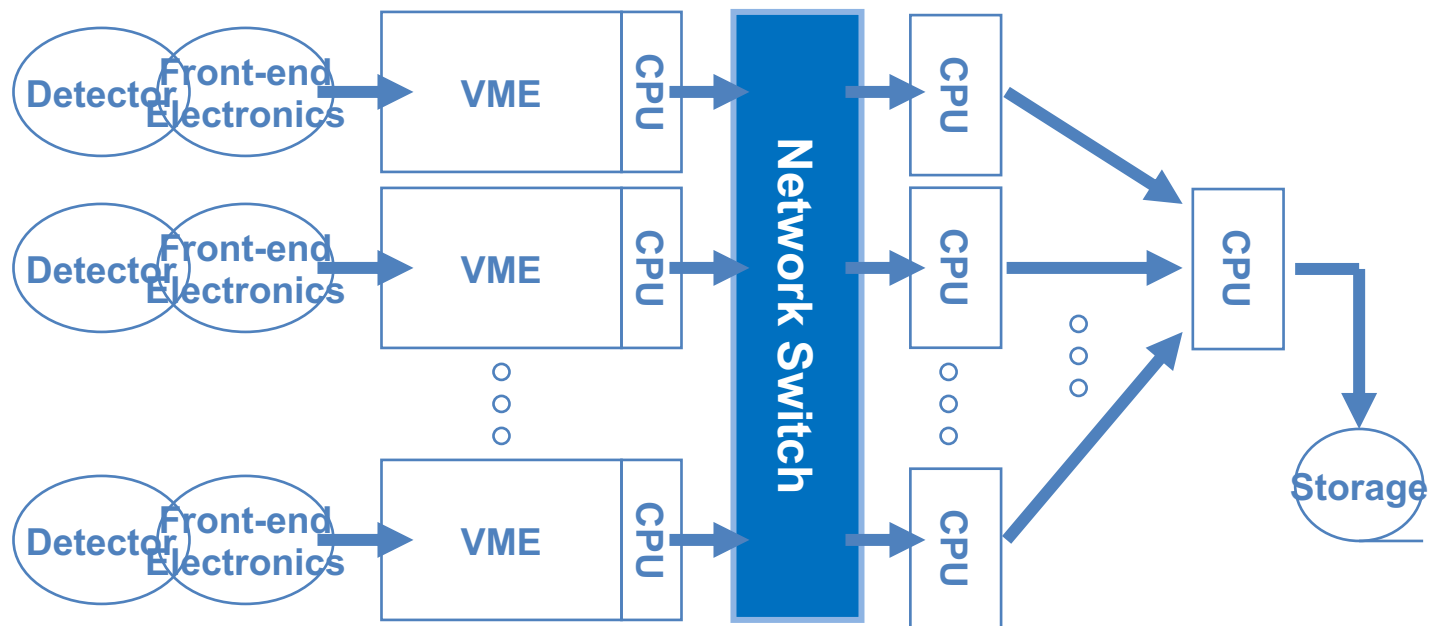
- Multi Detector (Crate) - Multi CPU
- バス型システム
 - バスがボトルネック
 - 多量なI/Fが必要
 - 煩雑なコネクション



大規模データ収集システムの現在

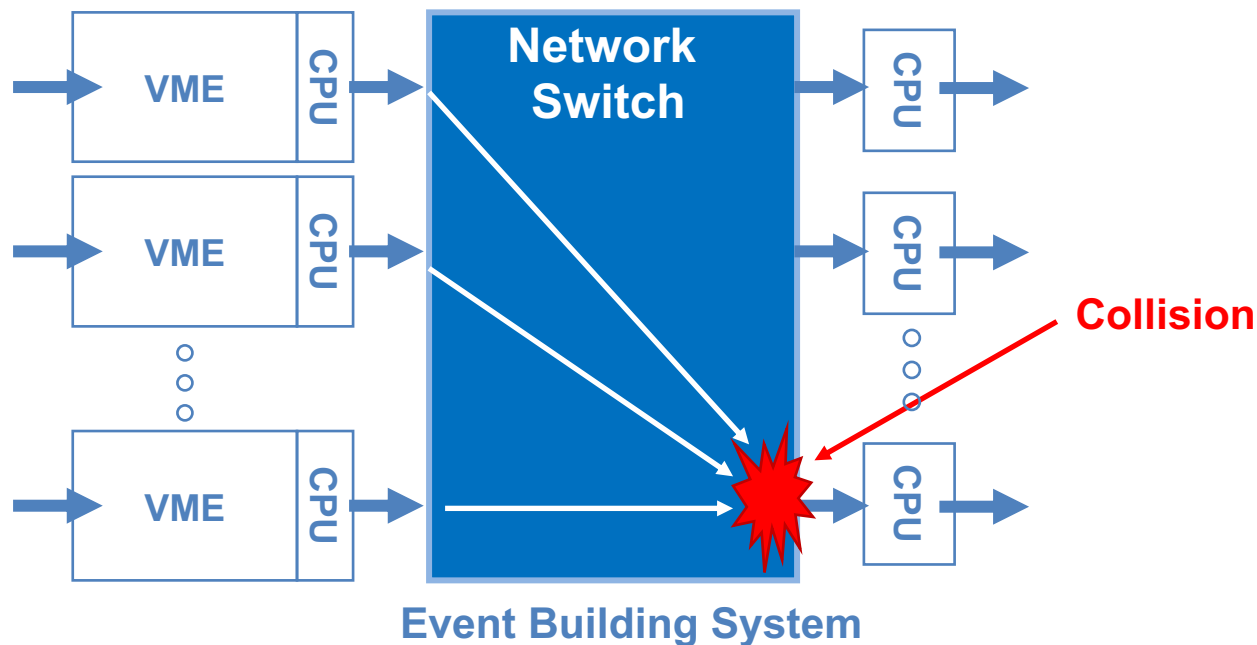
- スイッチ型システム

 - 煩雑性・I/Fの数 → スイッチ・ボトルネック



スイッチ型システムの問題点

- スイッチングネットワークの問題点
 - スイッチボトルネックの発生
 - データ（イベントフラグメント）の衝突の発生



スイッチ型システム問題の解決策

- 解決策

- 送信時にデータの流れをコントロール
 - データシャッフリング（データ送信時）手法の活用
 - 確率的にOK
 - QoS (Quality of Service) の活用
- グローバルトラフィックコントロールの利用
- 送信先からのデータ要求（プル型）
- スイッチ内で動的経路制御を行う方法

データ収集システムのデータフロー

- **プッシュ型データフロー**

- データの送信タイミングは送信側が決定

- 送信側はデータの準備が出来次第、データを送信
- 受信側の混雑具合は考慮せず

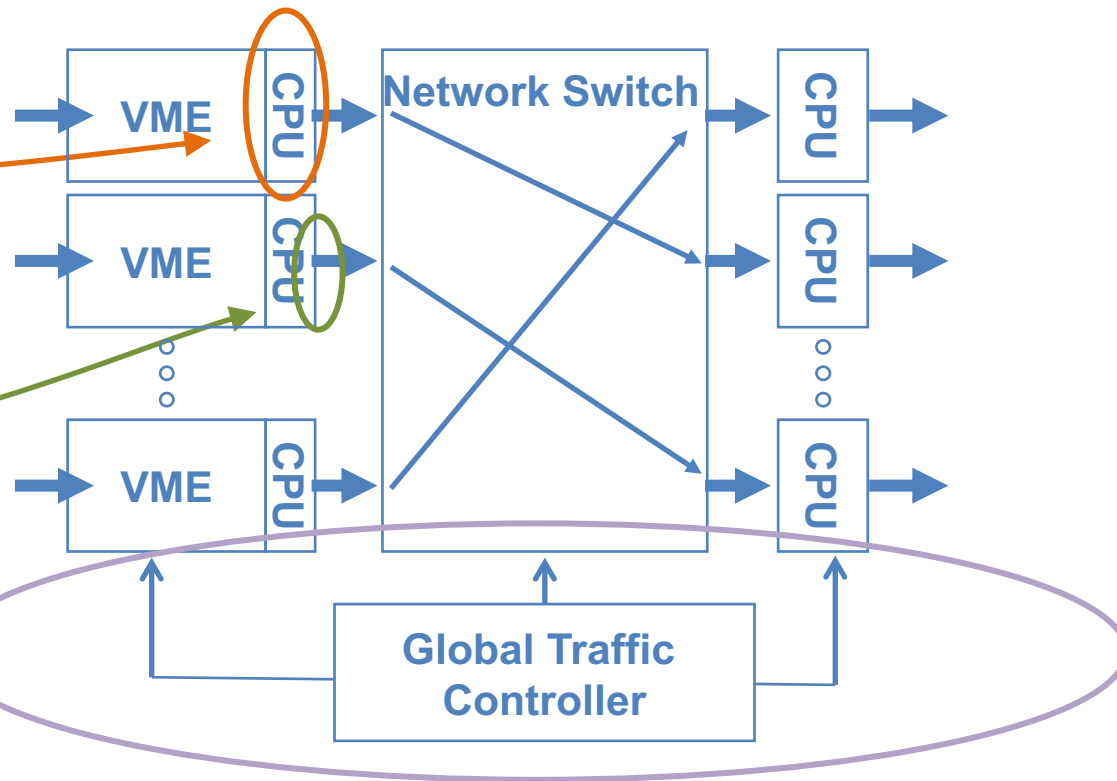
- **プル型データフロー**

- データの送信タイミングは受信側が決定

- 受信側が自分のCPUの空き具合を考え、欲しい時にデータを要求
 - 要求を受けた送信側での送信タイミングは送信側が決定

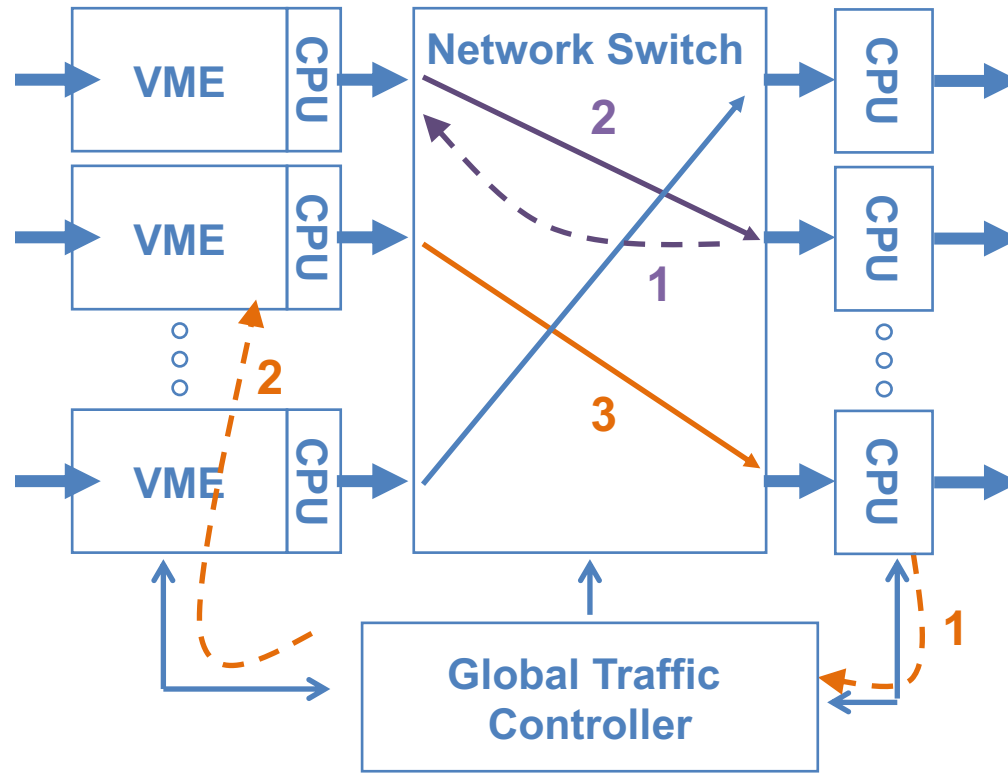
プッシュ型システム

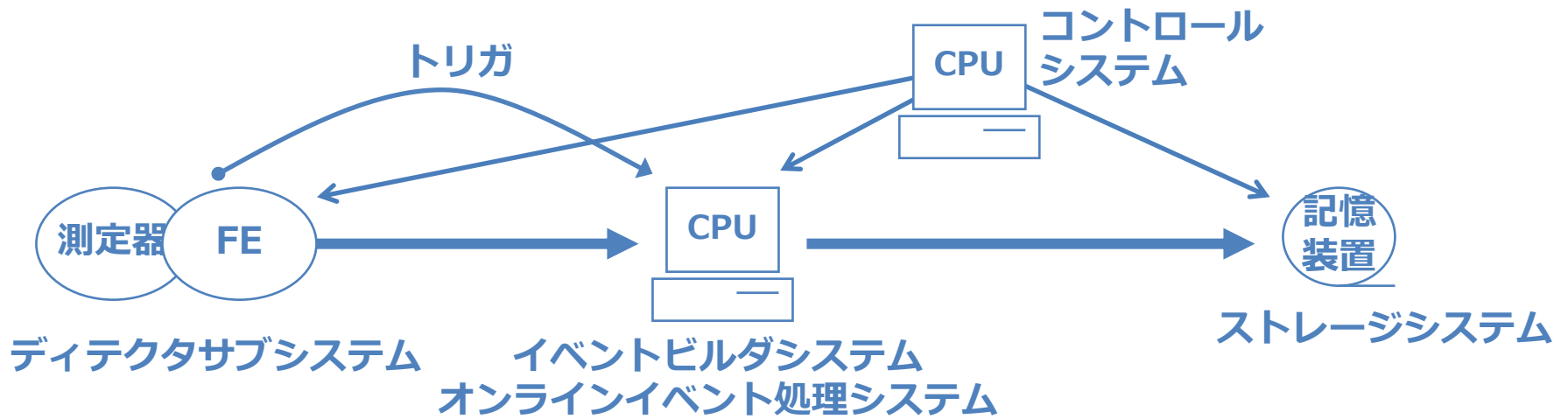
- Traffic Shaping
- Global Traffic Control
- QoS Control



プル型システム

- Direct on-demand Type
- Controller on-demand Type





データ収集システム コントロールシステム

データ収集システム全体の制御

- データ収集システムの状態管理

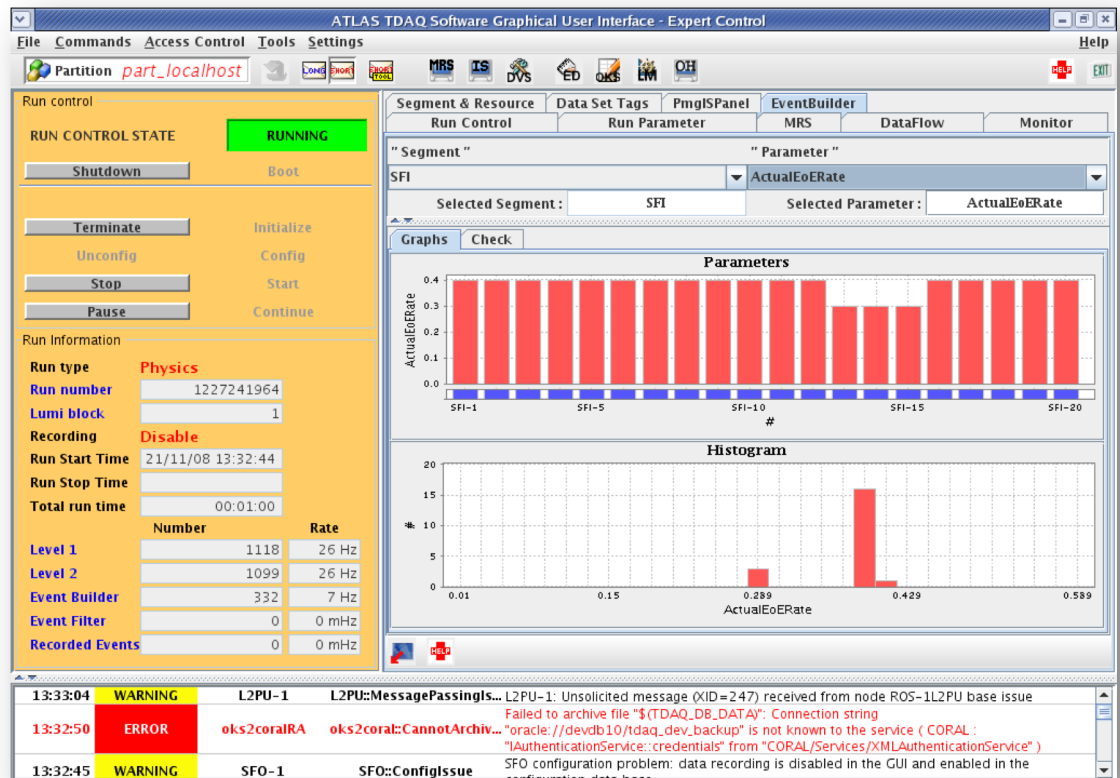
- 状態遷移の制御

- 停止状態 ⇔ 初期状態 ⇔ 収集状態
- エラー状態

- 状態のモニタ

- 記録と表示

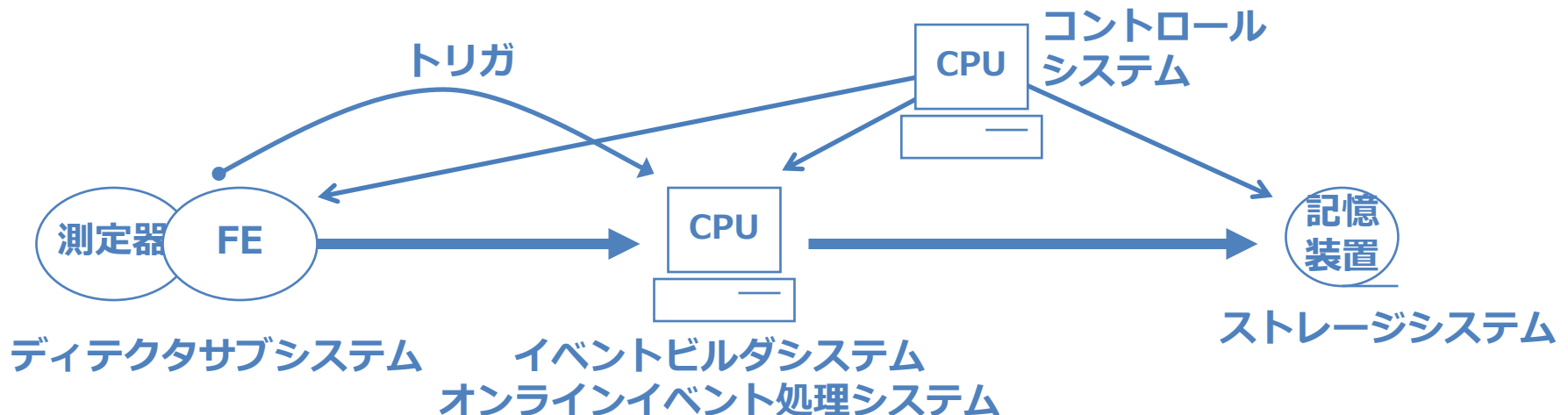
- オンラインでのデータモニタ



データ収集システム基本構成

- データ収集システムの構成要素
 - ディテクタサブシステム
 - フロントエンドシステム
 - イベントトリガシステム
 - イベントビルダシステム
 - オンラインイベント処理システム
 - ストレージシステム
 - コントロールシステム

収集 ・ 記録 ・ 制御 ・ モニタ ・ 解析



データ収集システムの設計

データ収集システム開発

- 実験の要求に見合うものを開発（選択）
 - 効率良くデータを収集するために

- 開発における設計の重要性
 - ソフトウェア工学
 - 開発プロセスモデル
 - 統一モデリング言語（UML）
 - オブジェクト指向開発
 - …

システムを考える上で

- 実験や観測の目的により、要求される機能、性能は異なる
 - 要求は必ずしも目的に合致していない
 - オーバースペックであることが多い
- 制約を考慮する
 - 期限、人的資源、予算、使用可能ツール、などの制約がある。特に期限は重要
- 開発だけでなく、検証、運用、保守、改良等も考慮する

性能を決定（1）

- 事象の属性を把握

- 事象を構成する機器とデータ伝送性能

- 事象の発生頻度（最大、最小、平均）

- 目的とする事象ではなく、読み込みを必要とする（バックグラウンド等を含む）事象であることに注意

- 各機器ごとの事象のデータサイズ（最大、最小、平均）

性能を決定（2）

- 記録性能を把握
 - 媒体は何にするか
 - 交換を必要とするか
 - 必要とするなら交換方式（方法）や交換時間
 - 事象にフィルターをかけるか
 - かけるならフィルター後の事象発生頻度

速度の目安

- **イーサネット**

- 10~1000 MByte/sec (100 Mbps~10 Gbps)
 - しかし目安としては最大でも半分

- **USB**

- 最大速度

- 1.1規格 1.5 MByte/sec (12 Mbps)

- 2.0規格 60 MByte/sec (480 Mbps)

- 3.0規格 600 MByte/sec (5 Gbps)

- でもバスがボトルネックになる可能性

- **DVD**

- 1 MByte/sec 程度

- **BD**

- 2~3 MByte/sec 程度

情報を伝える観点から

- 情報伝達の基本は5W1H
 - 必ずしも全部が必要というわけではないが、データだけでは情報が正しく伝達されるとは限らない
 - **いつ**（日時、時刻）
 - **どこで**（観測場所、実験施設など）
 - **誰が**（グループ名、シフト名など）
 - **何を**（実験対象、観測対象など）
 - **何のために**（本番、テスト、調整など）
 - **どのように**（実験条件・観測条件・データ形式など）
- システムとしては、データだけでなく、これらの基本情報も記録されていることが必要

データの読出と記録

- 計算機で後世に伝える手段は、半恒久的媒体に記録すること
- 基本はデータ源からデータを読み出し、記録媒体に書くこと
- 考えなければならないこと
 - データ源からのどのように読み出すか？
 - 記録媒体へどのように書き出すか？

システム設計におけるポイント

- ソフトウェア工学におけるシステム設計の課題
 - システムに求められる機能の高度化による複雑さ
- 複雑さ克服のための3つの指針
 - a. 抽象化とモデルの利用**
 - システムの性質のみをクローズアップさせる
 - 性質を正しく表現できる適切なモデルの適用
 - 本質ではない事項をそぎ落とす
 - b. 分割と階層化**
 - 構造化設計
 - システムの機能要素の階層化に主眼をおいた設計
 - c. 独立性**
 - 高い独立性
 - 要素内： 密結合
 - 要素外： 疎結合・単純

データ収集システム：収集データ

- 収集データ

- **データサイズ** (データフラグメントサイズ)

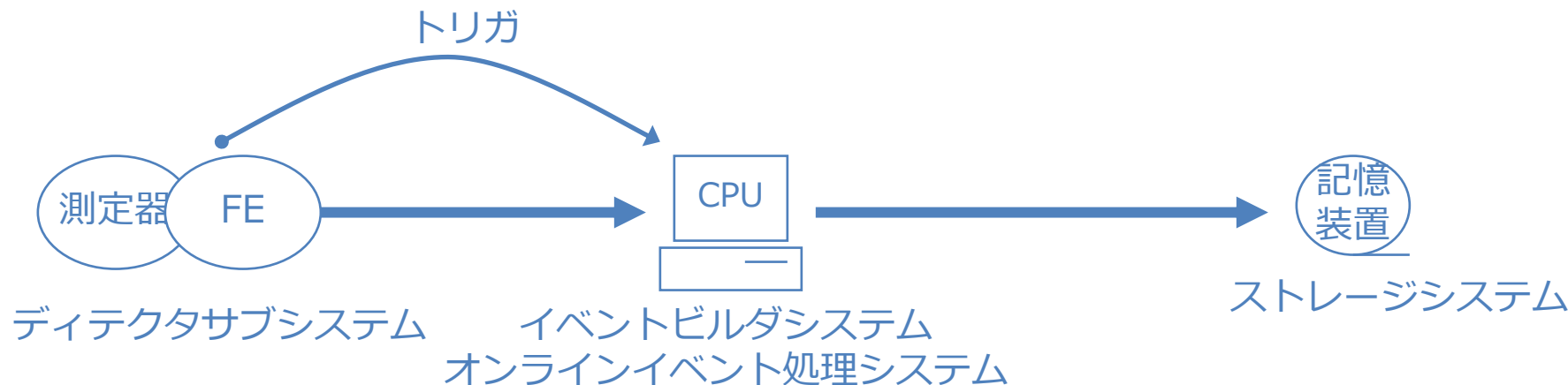
- 例) 1 kByte

- **データレート**

- 例) 1 kByte, 1 kHz \Rightarrow 1 MByte/sec

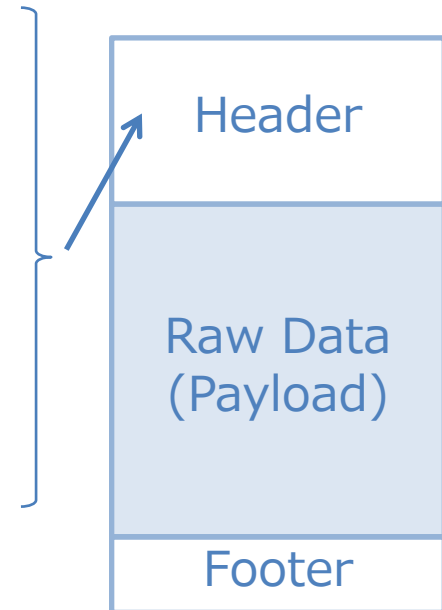
- **ディテクタサブシステム数**

- 例) 1 kByte, 1 kHz, 10 システム \Rightarrow 10 MByte/sec



データ収集システム：データ構造 1

- **ローデータ** (Raw Data)
 - 測定したバイナリデータそのもの
- **イベントデータ** (Event Data)
 - ローデータに処理するために必要な情報を付加したもの
 - 日付と時刻
 - シーケンス番号
 - 測定器番号 (サブシステム番号)
 - イベント番号 (イベントフラグメント番号)
 - イベントサイズ (イベントフラグメントサイズ)
 - イベントタイプ など



例) IPパケットフォーマット



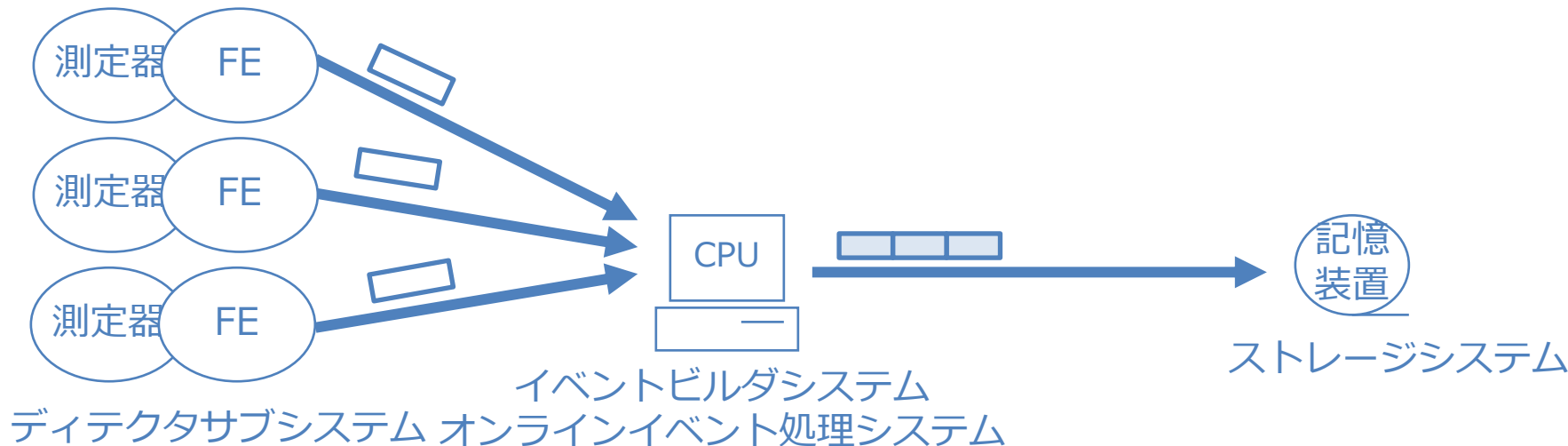
イベントとイベントフラグメント

- イベントフラグメント

- ディテクタサブシステムで生成されたローデータ
 - イベントフラグメントだけでは一つの事象を表すことはできない

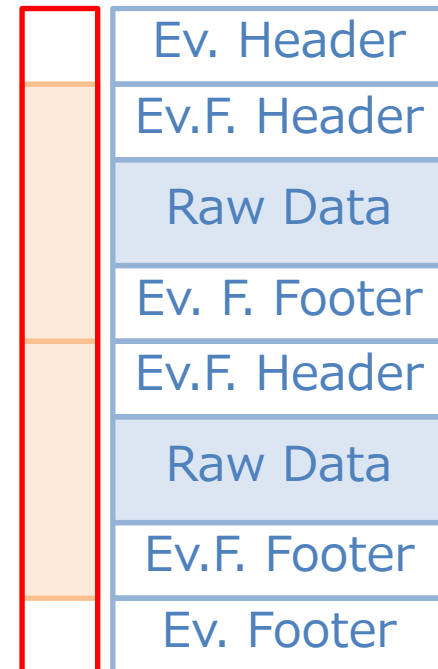
- イベント

- イベントフラグメントの集合
 - 一つの事象を表す



データ収集システム：データ構造 2

- **イベントデータ** (Event Data)
 - イベントフラグメントデータの集合
- **イベントフラグメントデータ** (Event Fragment Data)
 - ローデータに処理するために必要な情報を付加したもの
 - 日付と時刻
 - シーケンス番号
 - 測定器番号 (サブシステム番号)
 - イベント番号 (イベントフラグメント番号)
 - イベントサイズ (イベントフラグメントサイズ)
 - イベントタイプ など



データ収集システム：ステート

- データ収集システム
 - 状態が移り変わりシステムが動作するステートマシン
- 状態の決定
 - 必要な状態の定義と遷移条件
例)
 - STOPPED : ソフトウェア起動済み
 - CONFIGURED : システム動作のための設定済み
 - RUNNING : データ収集中
 - PAUSED : 一時停止中

状態遷移の例

- 状態（ステート）と遷移の例

- STOPPED

- ソフトウェア起動済み

- CONFIGURED

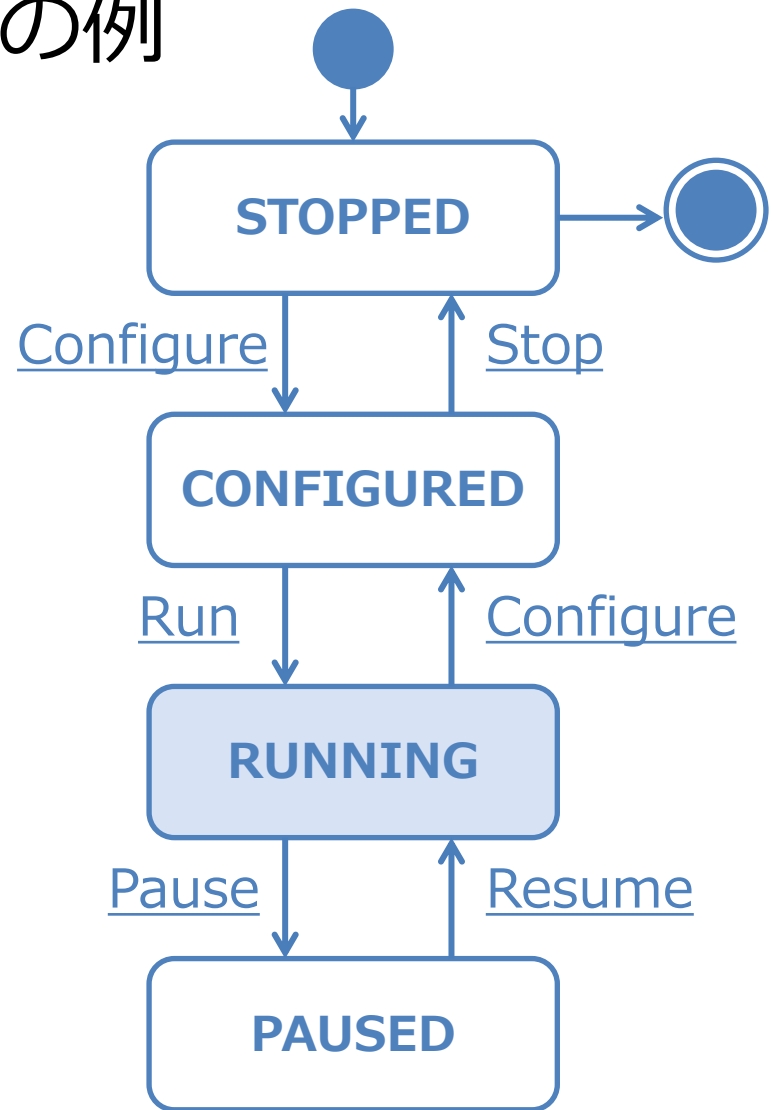
- システム動作のための設定済み

- RUNNING

- データ収集中

- PAUSED

- 一時停止



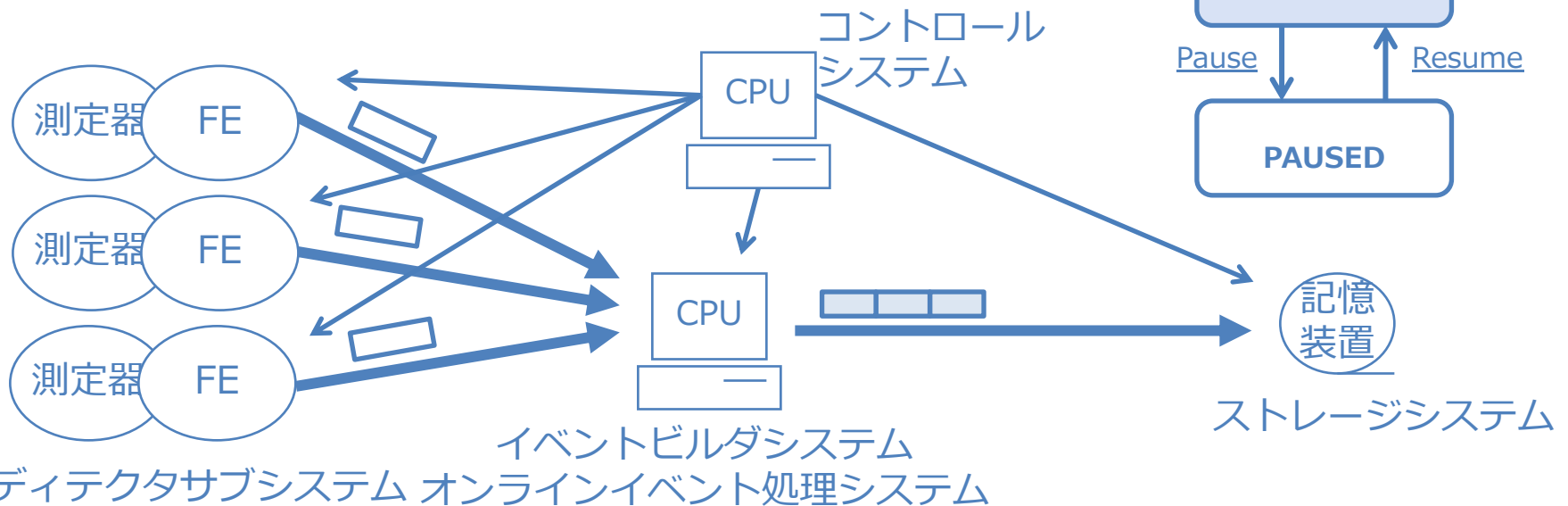
ステートとコマンド

- コマンド

- 状態を遷移させるためトリガとなるもの

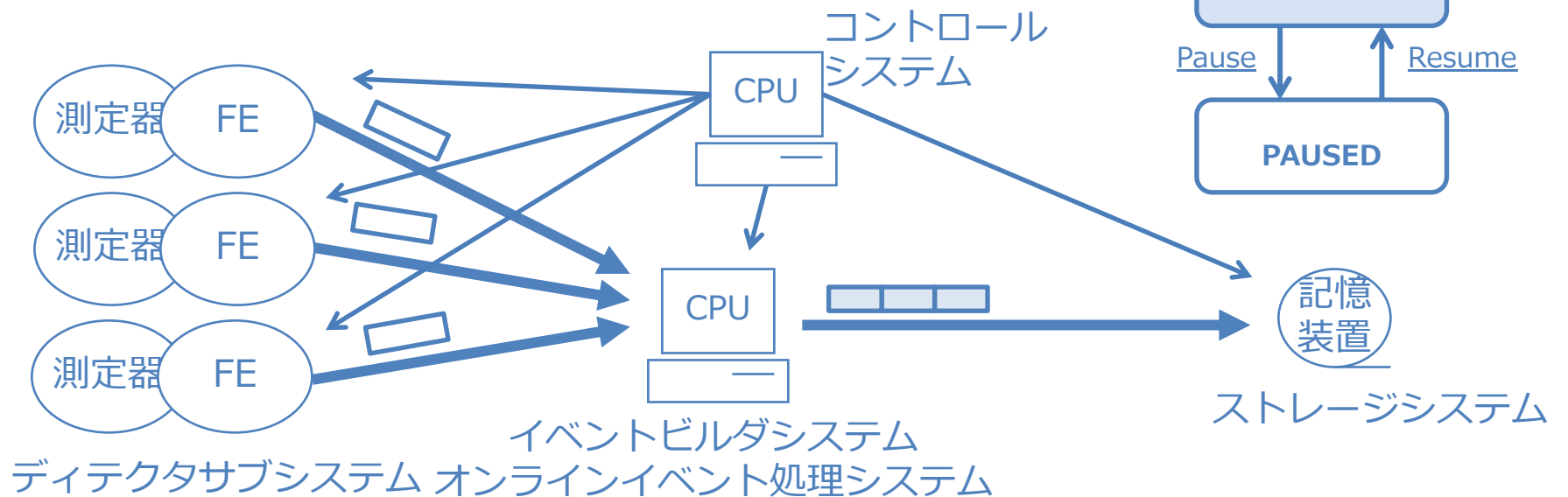
- 例)

- Configure • Run • Pause
 - Resume • Stop



エラー処理

- エラー
 - 種類
 - 発生場所
 - 通知方法



DAQシステムの構築のポイント 1

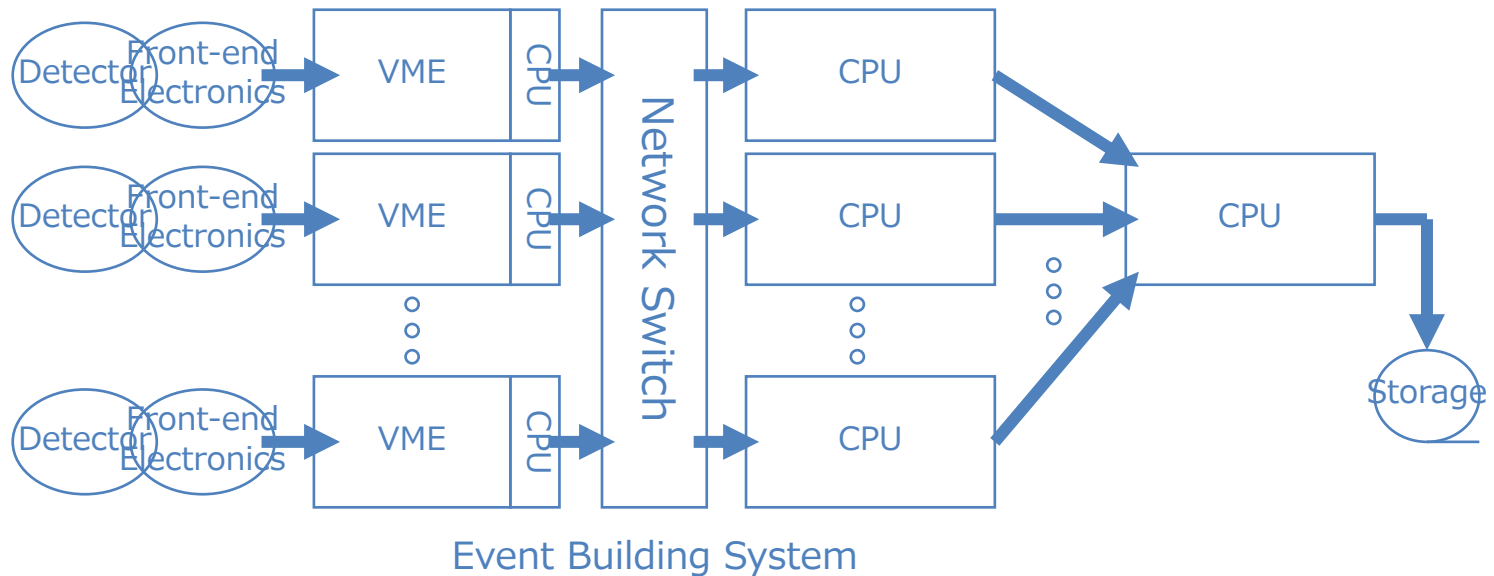
- ソフトウェアで実現する機能
 - 実験データ通信機能
（実験データ制御機能）
 - 制御データ通信機能
 - 状態管理機能
 - 実験データ読出し機能
 - 実験データ保存機能
 - データモニタ機能
 - システムモニタ機能
 - オンライン解析機能
 - システム制御機能
（ランコントロール機能）
 - メッセージ・ログ管理機能
 - システム設定機能
 - ユーザーインターフェイス機能

DAQシステムの構築のポイント 2

- 既存のものものの**再利用**
 - グループ（自分）が開発したもの
 - 共通フレームワークとして公開されているもの
 - DAQ-MW
- 機能の**重要度と優先度**
 - 重要な共通機能を持ったソフトウェアを設計
 - それをもとに各機能に対応するプログラムを設計
 - 重要度と優先度に従った開発

DAQシステムの構築のポイント3

- より効率良く収集可能なシステムの構築
 - どの位置にどのくらいのバッファを配置するか
 - どの位置でどのくらいのフラグメントしてまとめるか
 - プッシュ型かプル型か



まとめ

- **DAQシステム**とは、後から何度でも解析できるように、**データを情報として記録するシステム**
- ソフトウェアは後から変更可能と思わず、必ず**熟考してから開発を始める**
 - 後の解析のために、どのような情報をどのように記録しておくか十分に考察する
 - 正常に働いていることを常時示すために、正常であることを示すパラメータは何かを考察する
- 可能な限り標準化された規格などを使う
 - 標準規格を使えばマニュアルやテストツールなども揃っていることが多い
- ソフトウェアにすべての解を見つけようとせず、広く柔軟に解をさがす