今この瞬間を掴まえる！

~リアルタイム処理がもたらすアプリケーションの変化~

岩瀬 高博

Kobe Digital Labo, Inc.

Twitter:@okuyamaoo

Mail:iwase@kdl.co.jp

URL:https://github.com/okuyamaoo/

自己紹介

- 岩瀬 高博
  - > 株式会社　神戸デジタル・ラボ所属
  - > Twitter:　@okuyamaoo
- 活動
  - >分散キーバリューストア 

の開発
  - >分散処理系の勉強会の参加、主催
  - >分散処理の研究及び、業務適応

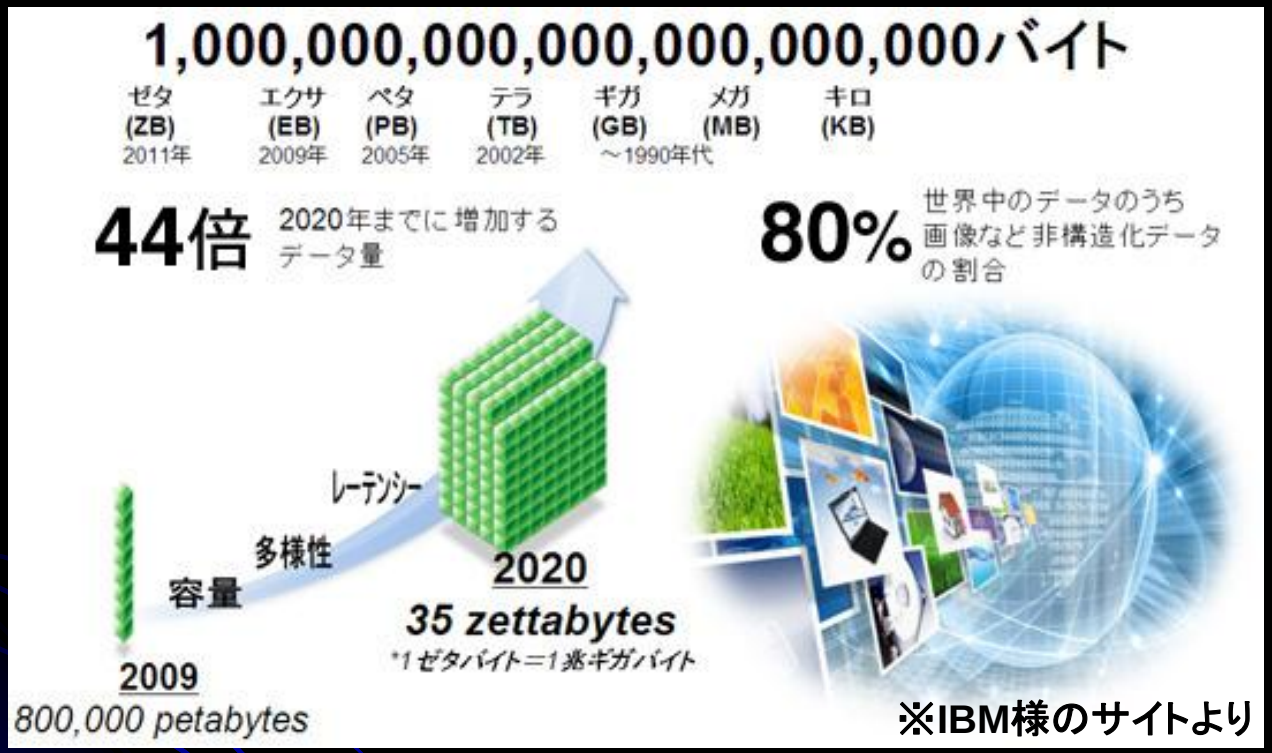
アジェンダ

1. 昨今のデータをとりまく環境
2. CEPとは？
   - Complex Event Processing？？
   - どのようなことが出来るのか？
3. Setsunaの紹介
   - アーキテクチャ
   - 現在の機能

昨今のデータをとりまく環境？

昨今のデータをとりまく環境

・大量のデータが生み出され続ける状態



facebook Google+ twitter amazon.com

一部のサービスが作り出したデータ以外にも
大量のデータが企業内にも存在する。

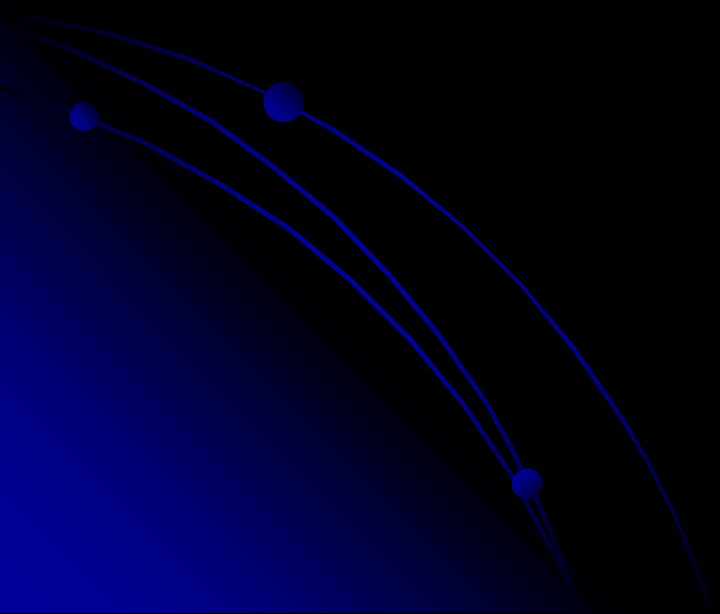
昨今のデータをとりまく環境

- どのように処理するか？

一度全てストレージに格納してから定期的に処理を行うバッチが主流。
Hadoopなど非常に有効に活用されている。

昨今のデータをとりまく環境

- 本日紹介する技術は？？
  - 複数データのリアルタイム処理エンジンの一種



昨今のデータをとりまく環境

・本日紹介する技術は？？

複数データのリアルタイム処理エンジンの一種

CEP

CEPってなに？

- Complex Event Processing??
- どんなことが出来るの？
- どんなものがあるの？

概要

・CEPは

Complex Event Processingの略

意味は？

概要

・CEPは

Complex Event Processingの略

意味は？

Complex = 複合

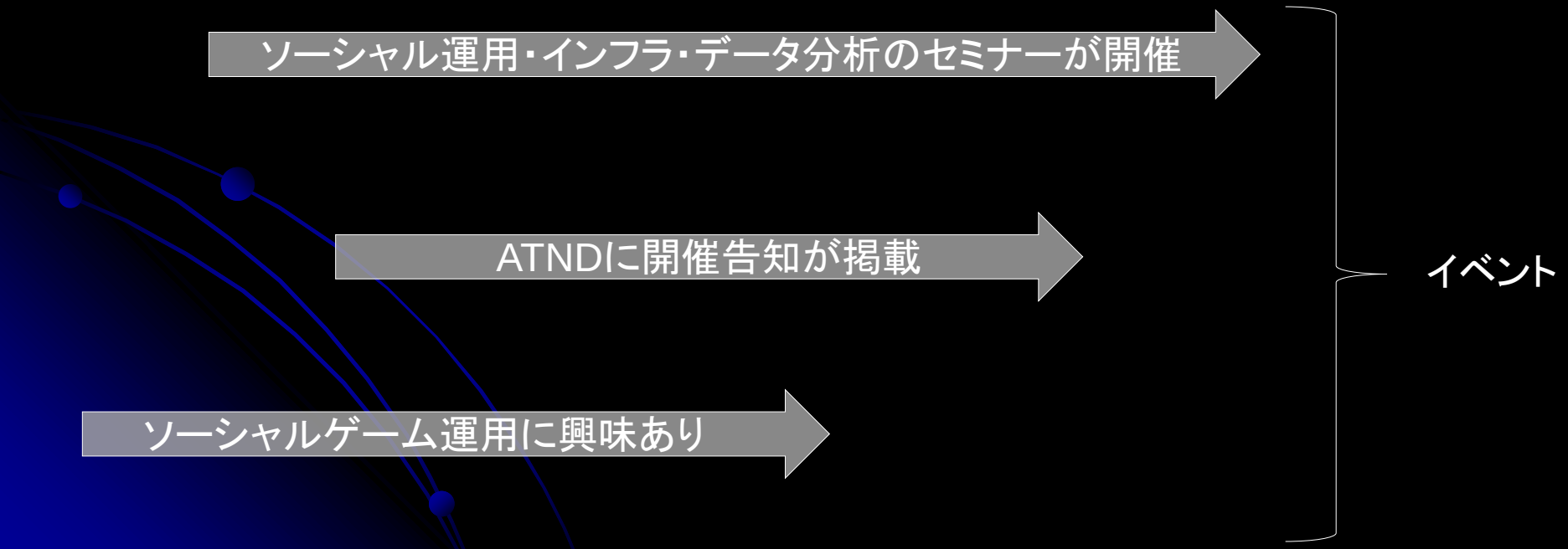
Event = 出来事

Processing = 処理

概要

・複合　イベント　処理

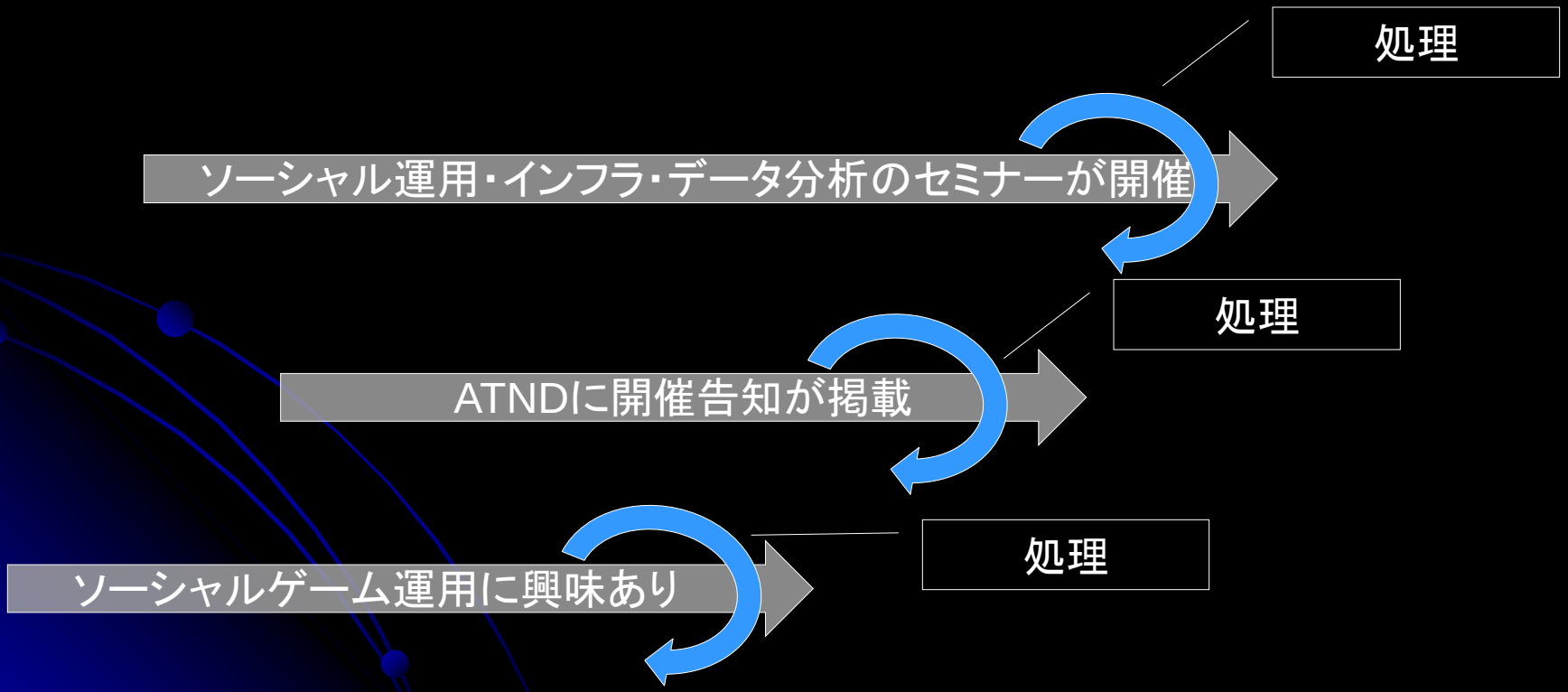
複数のイベント(出来事)を処理すること



概要

・イベントを処理するのはいいが

今までとの違いは？？



概要

・イベントを処理するのはいいが

今までとの違いは？？

CEPの”**C**”の意味　＝　**複合**がポイント！！

どのようなことが出来るのか？

・今までのイベントを基に考えてみましょう

これらのデータがそれぞれ個別に発生

① ソーシャルの勉強会が開催される

② ATNDに告知が掲載されている

③ ソーシャルゲーム運用に興味あり

どのようなことが出来るのか？

・今までのイベントを基に考えてみましょう

これらのデータがそれぞれ個別に発生

それぞれのデータでは意味が見いだせない

① ソーシャルの勉強会が開催される

② ATNDに告知が掲載されている

③ ソーシャルゲーム運用に興味あり

どのようなことが出来るのか？

・今までのイベントを基に考えてみましょう

これらのデータがそれぞれ個別に発生

それぞれのデータでは意味が見いだせない

全部横断的に見た場合は？？

① ソーシャルの勉強会が開催される

② ATNDに告知が掲載されている

③ ソーシャルゲーム運用に興味あり

どんなことが出来る？

・全部ひっくるめて見てみる。

ソーシャルの勉強会が開催される → 勉強会が開催

ATNDに告知が掲載されている →

ソーシャルゲーム運用に興味あり →

どんなことが出来る？

・全部ひっくるめて見てみる。

ソーシャルの勉強会が開催される → 勉強会が開催

ATNDに告知が掲載されている → 勉強会告知

ソーシャルゲーム運用に興味あり →

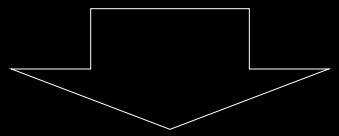
どんなことが出来る？

・全部ひっくるめて見てみる。

ソーシャルの勉強会が開催される → 勉強会が開催

ATNDに告知が掲載されている → 勉強会告知

ソーシャルゲーム運用に興味あり → 運用に興味有り



どんなことが出来る？

・全部ひっくるめて見てみる。

ソーシャルの勉強会が開催される → 勉強会が開催

ATNDに告知が掲載されている → 勉強会告知

ソーシャルゲーム運用に興味あり → 運用に興味有り

↓

メールによる訴求を行う

どのようなことが出来るのか？

・と、このようにデータの流れを横断的に見ていくと、
いままで見えなかったことが見えてきたりします

これを高速に処理して
「メール訴求」まで
面倒を見てくれるのが

CEP

活用されているところ

①クレジットカードの不正利用監視

1つのクレジットカードが2つの異なる場所で、
ほぼ同時刻に利用された場合にアラートを出す。

②株の自動売買

ある銘柄の株価が閾値を超えたら売りの処理を
おこなう。もしくは買い付ける。

③サイトへの攻撃検知

特定のIPアドレスから一定時間に閾値以上の
アクセスがあった場合に遮断する。

Setsunaの紹介

- アーキテクチャ
- 現在の機能

その前に

- Setsunaの名前の由来

漢字では刹那と書きます

小数の単位を表す用語日本には以下のようなものがあります。

分、厘、毛、糸、忽、微、繊、沙、塵、埃、渺、漠、模糊、逡巡、須臾、瞬息、弾指、刹那、六徳、虚空、清浄、阿頼耶、阿摩羅、涅槃寂静(計24単位)
(wikipediaより)

速そう $>10^{-24}$を表す

Setsunaの紹介

- アーキテクチャ
- 現在の機能

アーキテクチャ

- 言語

  Java

- 構成要素

  Setsunは1つのコア

  1. SetsunaCore

  3つのカスタマイズ要素で構成されています

  1. Adapter
  2. Query
  3. UserEvent

アーキテクチャ

・構成要素

1. SetsunaCore

   Setsuna本体

   カスタマイズ要素を制御し、全体をコントロールしています

   データベースを内蔵し投入されるデータの管理を行う

アーキテクチャ

・構成要素

1.Adapter

AdapterはSetsunaにイベントを入力する部分になります

入力後のデータを好きな単位で分解してカラム情報としてSetsuna本体のDBに格納します

アーキテクチャ

・構成要素

2.Query

Adapterからの入力に検索をおこない
データの変化を調べる箇所

トリガーというごく単純な変化を掴む箇所

クエリーというSQLで複雑に検証する箇所

アーキテクチャ

・構成要素

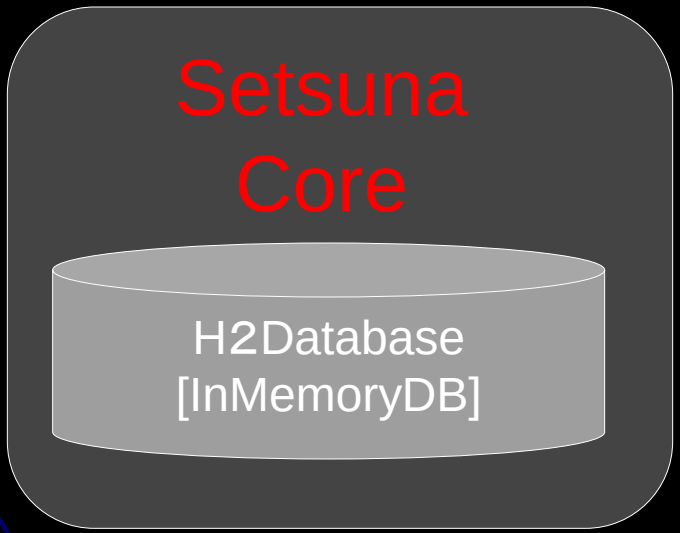
3.UserEvent

Queryで目当てのデータを見つけた場合に実行したいタスク

先ほどの例の「メール訴求」を行う部分

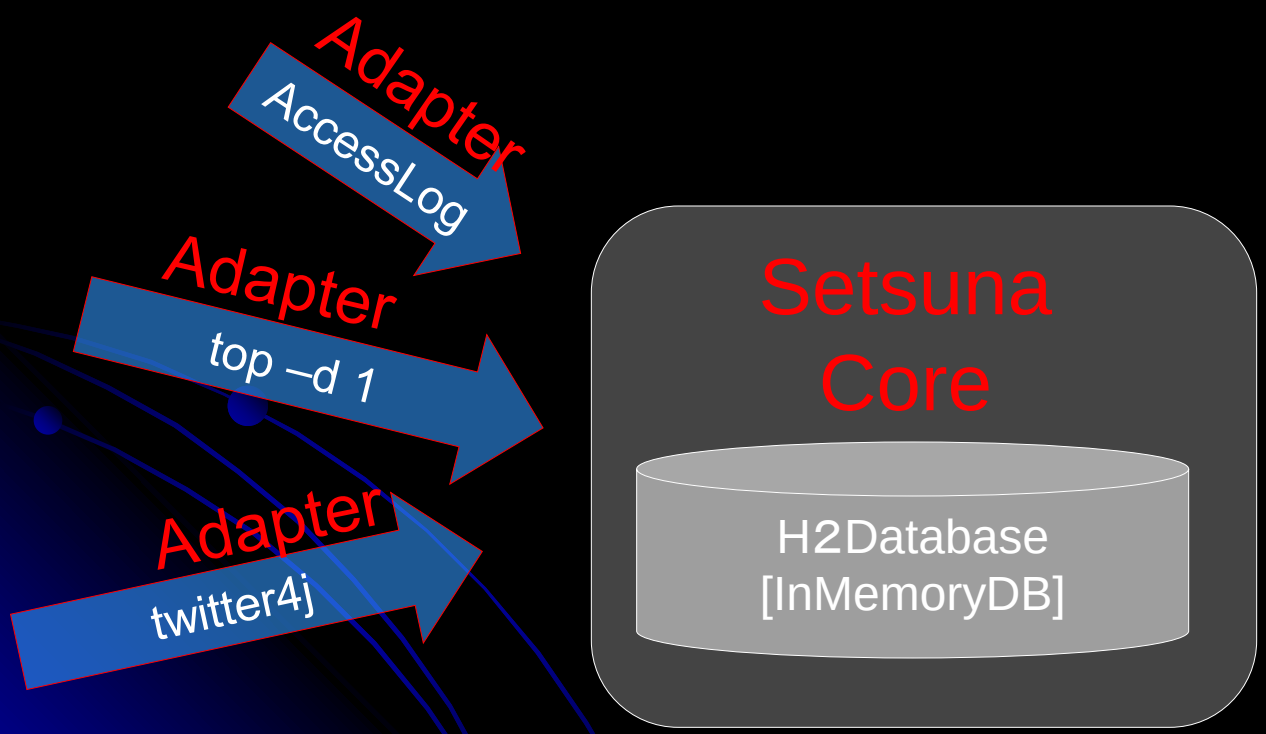
アーキテクチャ

- Over view



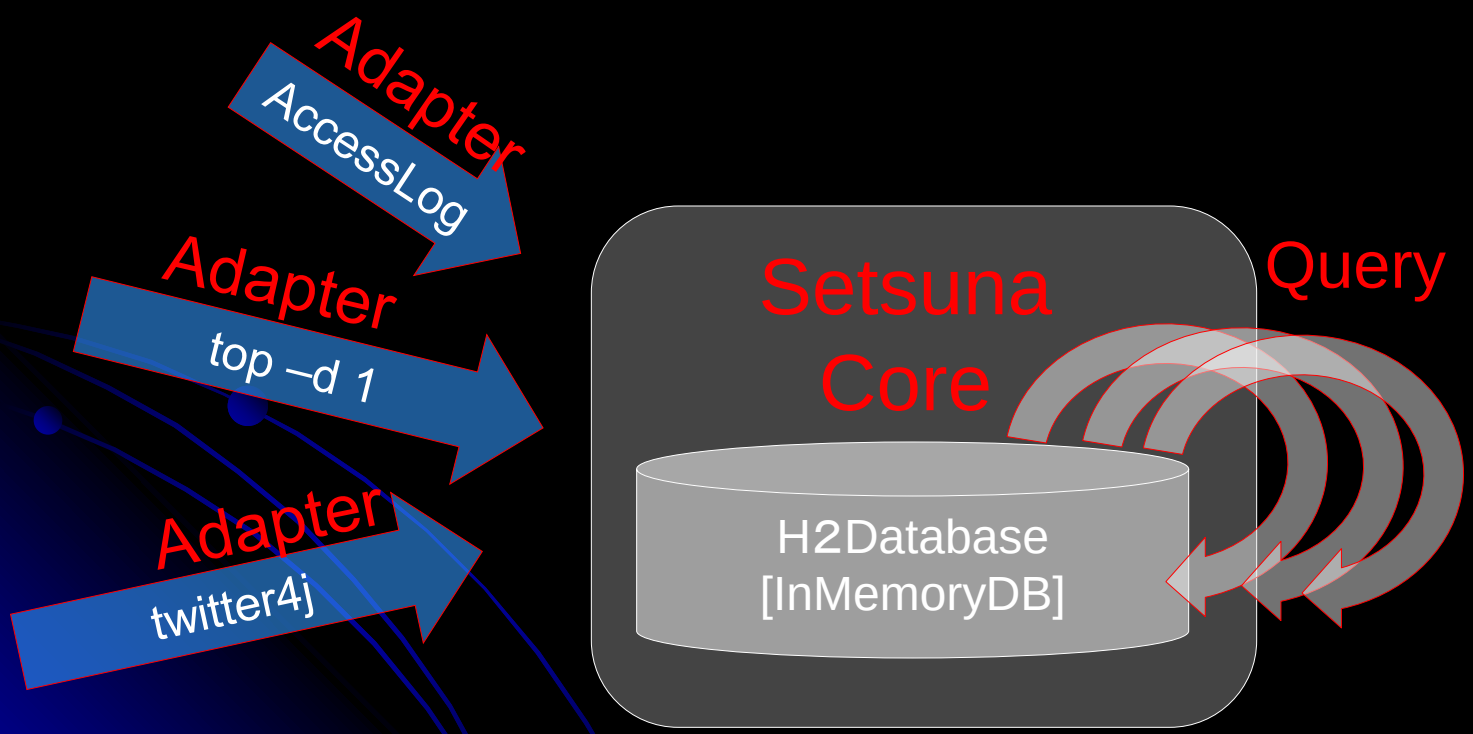
アーキテクチャ

・Over view



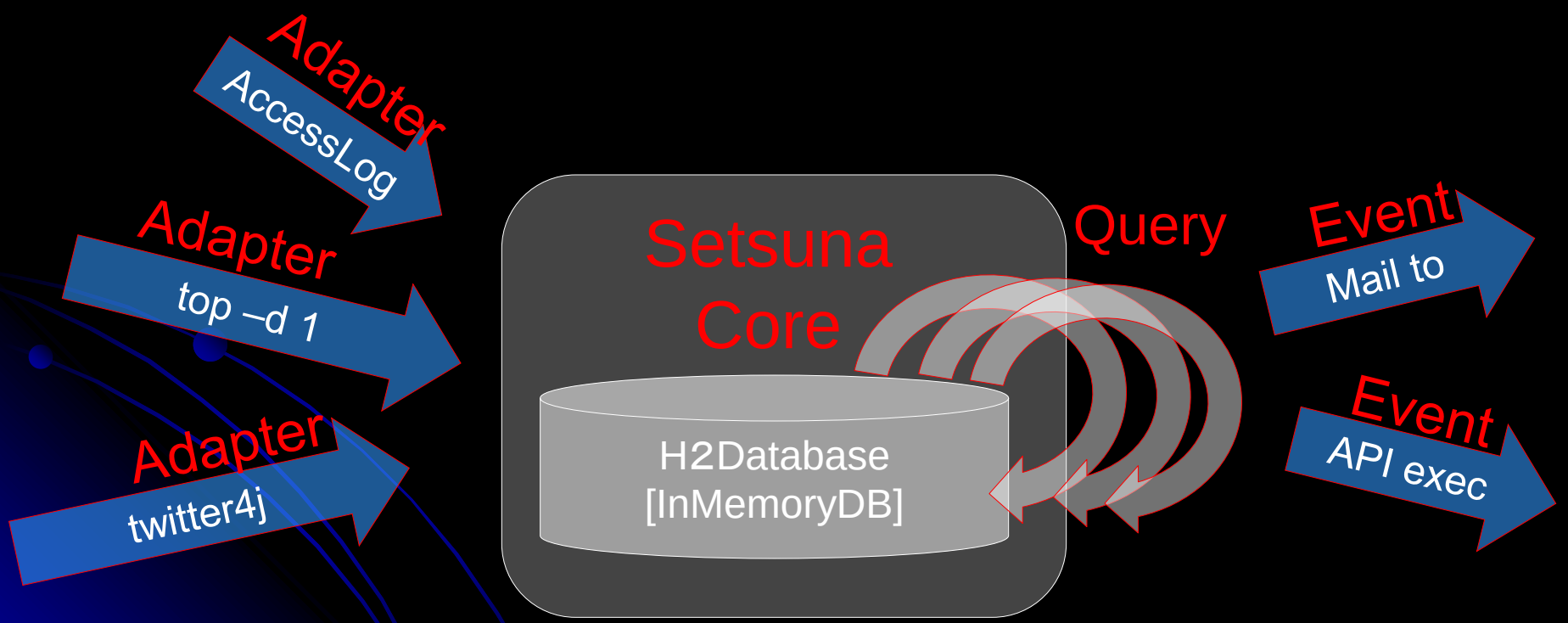
アーキテクチャ

・Over view



アーキテクチャ

・Over view



Setsunaの紹介

- アーキテクチャ
- 現在の機能

現在の機能

- 先日リリースしたSetsuna-0.0.2では
  あらかじめ、3つの構成要素を実装済みで
  リリースしています

現在の機能

- Adapter

  あらかじめ2種類のAdapterを実装済み

  ①パイプライン入力を受付るAdapter
  　OS標準のコマンドと簡単に連携できます

  ②MessagePack-RPCでのサーバモード
  　MessagePack-RPCが使える言語で
  　あれば簡単にデータが渡せる

現在の機能

①パイプライン入力を受付るAdapter指定例

```
$top –b –d 1 | grep –line-buffered top “top -” | ¥
  java -jar setsuna.jar ¥
  –stream top
```

topコマンドの先頭行だけを渡している

現在の機能

②MessagePack-RPCでのサーバモード

```
java -classpath ./:./lib/msgpack/*:setsuna.jar ¥
 setsuna.core.SetsunaMain -server true
```

RPCクライアントのメソッド定義

```
1   public static interface RPCInterface {
2       /**
3        *
4        * @param data カラムデータの配列。必ずカラム数分必要
5        * @return int 結果:正常終了=0, カラム定義と合わない=-9, 内部エラー=-1
6        */
7       int next(String[] data);
8   }
```

現在の機能

- Query

  起動引数でトリガーとクエリーを渡せます

現在の機能

- Query

-trigger 指定

```
$top –b –d 1 | grep –line-buffered top “top -” | ¥
  java -jar setsuna.jar
  –stream top
  –trigger “COLUMN10 > 1”
```

現在の機能

- Query

-query 指定

```
$top –b –d 1 | grep –line-buffered “top -” | ¥
 java -jar setsuna.jar –stream top
  –trigger “COLUMN10 > 1” ¥
  -query “select
              *
            from (
                    select
                      avg(to_number(COLUMN10)) as ldavg
                    from
                      top
                    where
                      C_TIME > (current_timestamp - 60000)
                   ) ldavgtable
             where
               ldavg > 2”
```

現在の機能

- UserEvent

  起動オプション実行したいイベントを指定できます。今のところサポートしているのは

  - コマンドラインでの命令
  - SQL実行

現在の機能

- UserEvent

  -eventでコマンドや、独自で作成したシェルを呼び出す

```
$top –b –d 1 | grep –line-buffered “top -” | ¥
 java -jar setsuna.jar
   –stream top
   –trigger “COLUMN10 > 1” ¥
   -query “select * from ・・・省略 ・・・”
   -event  “logger ロードアベレージの平均が閾値です！！！！！”
```

現在の機能

- UserEvent

  -eventqueryでSetsunaの内部DBからデータを取得する

```
$top –b –d 1 | grep –line-buffered “top -” | ¥
 java -jar setsuna.jar
   –stream top
   –trigger “COLUMN10 > 1” ¥
   -query “select * from ・・・省略 ・・・”
   -eventquery  “select
                    avg(COLUMN10) as ldavg
                  from
                   top
                  where
                    C_TIME > (current_timestamp – 60000)”
```

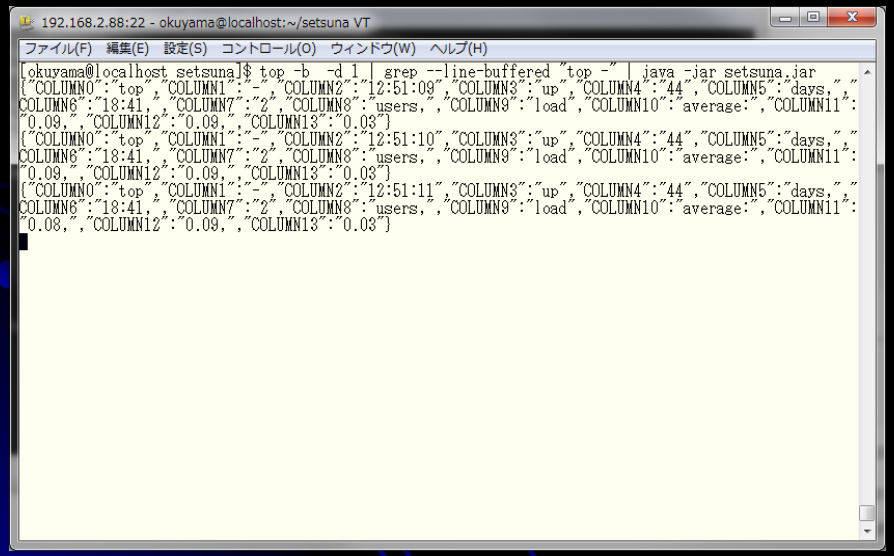
現在の機能

今までの例だと1つのデータのストリームに処理している

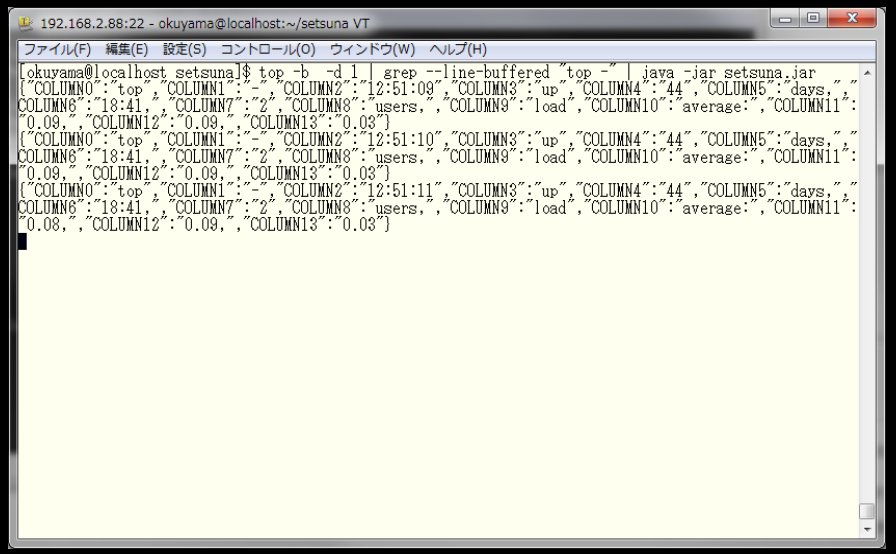
複数の異なるストリームのデータを処理するには？？

>複数のプロセスを立ち上げるだけでOK

ターミナル1



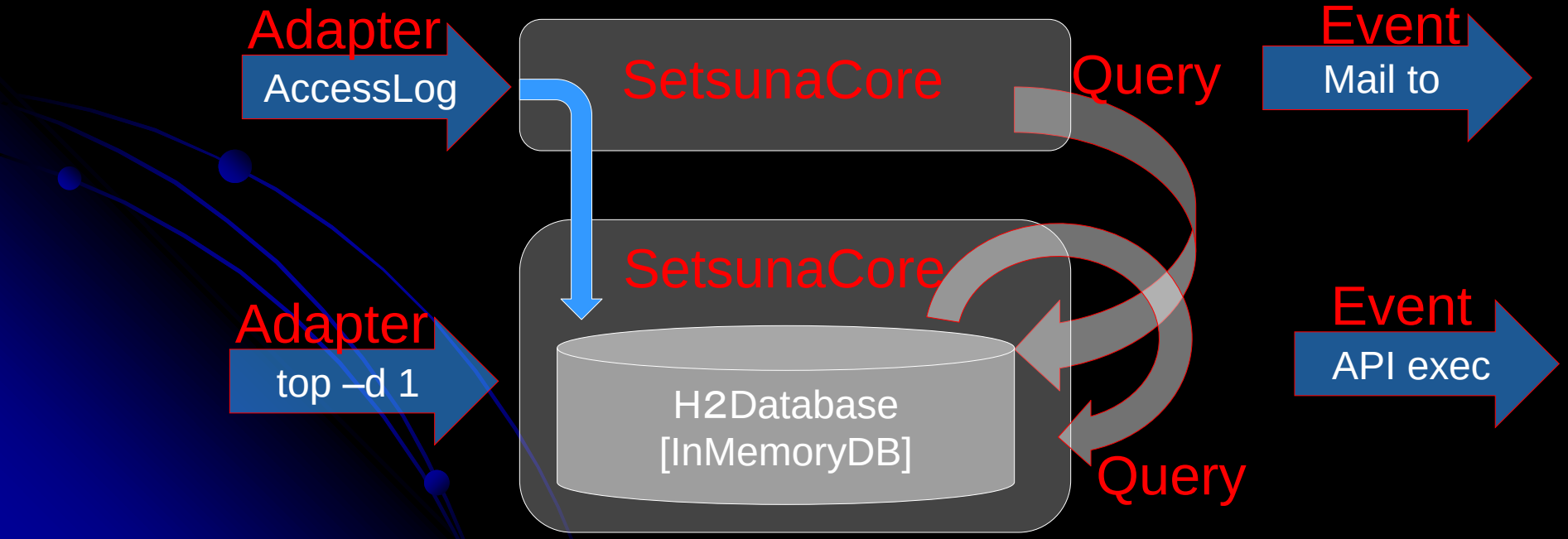
ターミナル2



互いのデータを参照できる

現在の機能

- 異なるストリームのデータを処理する
  これは複数のプロセスを立ち上げた場合にどれか1つの
  Setsunaが自動的にDBサーバとなり動くため
  そのDBを全Setsunaプロセスで共有するため全ての
  箇所から全てのデータが参照可能



最後に

- Information

Development

https://github.com/okuyamaoo/setsuna/

http://sourceforge.jp/projects/setsuna/ (※リリースのみ)

Facebook

http://www.facebook.com/takahiro.iwase

twitter

@okuyamaoo

Thank you!

Setsuna