TIS
IT Holdings Group
Go Beyond

PostgreSQLの監視運用を楽にするツールのご紹介

～pg_monzとfluentd～

TIS株式会社

中西 剛紀

自己紹介

TIS IT Holdings Group | Go Beyond

- 氏名：中西 剛紀 (なかにし よしのり)
- 所属：TIS株式会社　OSS推進室
- 主な活動領域：PostgreSQL全般
  - 日本PostgreSQLユーザ会(JPUG)
    勉強会でたまに講演しています
    http://www.slideshare.net/naka24nori/jpug25
  - PostgreSQLエンタープライズコンソーシアム (PGECons)
    WGの主査としてセミナー講演してみたり
    http://itpro.nikkeibp.co.jp/atcl/column/15/052800134/052900004/?ST=oss&a
- お仕事：OSSのサポート，技術支援

[PR]TIS OSSサポートサービス

TIS IT Holdings Group | Go Beyond

- OSSのプロダクトサポートやってます。
  - 対象OSS

[PR]TIS OSSサポートサービス

TIS IT Holdings Group | Go Beyond

- お問合せ先

TIS株式会社 OSS推進室
OSSサポートグループ

oss-sales@ml.tis.co.jp
http://www.tis.jp/service_solution/oss/

このセッションのゴール

TIS IT Holdings Group | Go Beyond

- pg_monz, Fluentdの存在を知る。
- pg_monz, Fluentdでできることを知る。
- pg_monzを試したくなる。
- PostgreSQLを使いたくなる。

AGENDA

TIS IT Holdings Group | Go Beyond

- PostgreSQL標準機能を使った運用
- pg_monzを使ったPostgreSQLの監視
- Fluentdを使ったPostgreSQLのログ管理

はじめに

- 本日紹介するノウハウは機能的に実現可能なことを確認済ですが、本番運用に適用する際は、サーバにかかる負荷等を各自検証されることをお勧めします。

はじめに

TIS IT Holdings Group | Go Beyond

- 動作確認したプロダクト
  - CentOS 6.6(x86_64)にRPMで導入

プロダクト名	バージョン
PostgreSQL	9.4.3
Zabbix	2.4.5
Fluentd(td-agent)	0.12.7(2.2.0)

  - 以下のFluentdプラグインを導入

プラグイン名	バージョン
fluent-plugin-parser	0.5.0
fluent-plugin-record-reformer	0.6.3
fluent-plugin-pgjson	0.0.7

データベースの運用管理

TIS IT Holdings Group | Go Beyond

- データベース運用管理の目的
  - DBの状態を把握して健全な状態に保つ
- データベース運用管理の種類
  - 死活監視
  - リソース監視
  - 性能監視/分析/チューニング
  - バックアップ/リストア
- 監視が運用管理の基本のき
  - 正しく現状を把握しなければ何もできない
  - 今回は監視に着目してノウハウをご紹介

監視に使うPostgreSQLの機能

TIS IT Holdings Group | Go Beyond

- 稼働統計情報
- ログ

監視に使うPostgreSQLの機能

TIS IT Holdings Group | Go Beyond

- 稼働統計情報
- ログ

稼働統計情報とは

TIS IT Holdings Group | Go Beyond

- 稼働統計情報って何？
  - PostgreSQLのアクティビティ情報を蓄積
  - stats_collectorというプロセスが記録
  - ユーザはビューを通して参照可能
- 稼働統計情報の取得方法
  - SELECT文でビューにアクセスして取得

稼働統計情報とは

TIS IT Holdings Group | Go Beyond

- 稼働統計情報で何がわかる？
  - 代表的な稼働統計情報用のビュー

ビュー名	取得できる情報
pg_stat_activity	接続中のクライアントの処理状況 ・ 実行しているSQL ・ 接続開始、トランザクション開始、SQL開始時間 ・ プロセス状態（SQL実行中/問合せ待ち/ロック待ち）
pg_stat_database	DB単位のアクティビティ状況 ・ コミット/ロールバック数 ・ 現在の接続数 ・ 更新&参照件数 ・ deadlock数
pg_stat_user_tables	テーブル単位のアクティビティ状況 ・ シーケンシャルスキャン回数、件数 ・ 更新件数 ・ 最後の autovacuum、analyze 時間

稼働統計情報の難点

TIS IT Holdings Group | Go Beyond

- SQLを書くのがダルい
  - キャッシュヒット率を知りたいとき

```
SELECT  datname, round(blks_hit*100/(blks_hit+blks_read), 2)  AS cache_hit_ratio
FROM pg_stat_database WHERE blks_read > 0

datname | cache_hit_ratio
------------+-----------------
postgres | 99.00
```

- 主に起動してからの累積値を保持
  - 最新の値だけでは判断できないことが多い
  - 過去の値との差分(変化)から判断可能

面倒なことは Zabbix と pg_monz に任せませんか？

Zabbix

TIS IT Holdings Group | Go Beyond

- OSSの統合監視ツール

- 非常に多くの機器の監視が可能
  - NW機器,サーバ,ミドルウェア,アプリケーション
  - 対応するプラットフォームも多い
- 監視ツールに必要な機能を網羅
  - 収集データの保管、傾向分析
  - メール等での障害通知
  - Webインタフェースによるグラフィカル表示
- 国内でもZabbixの導入事例が増えている。

pg_monz（ぴーじーもんず）

- 正式名称：
  PostgreSQL monitoring template for Zabbix
- ZabbixにPostgreSQLの監視機能を追加するテンプレート＆スクリプト
  - TISとSRA OSS, Inc.日本支社で共同開発
  - Apache License Version 2.0で公開
  - 2013年12月 Version 1.0 リリース
  - 2015年　4月 Version 2.0 リリース
- 私も開発メンバーの1人です。

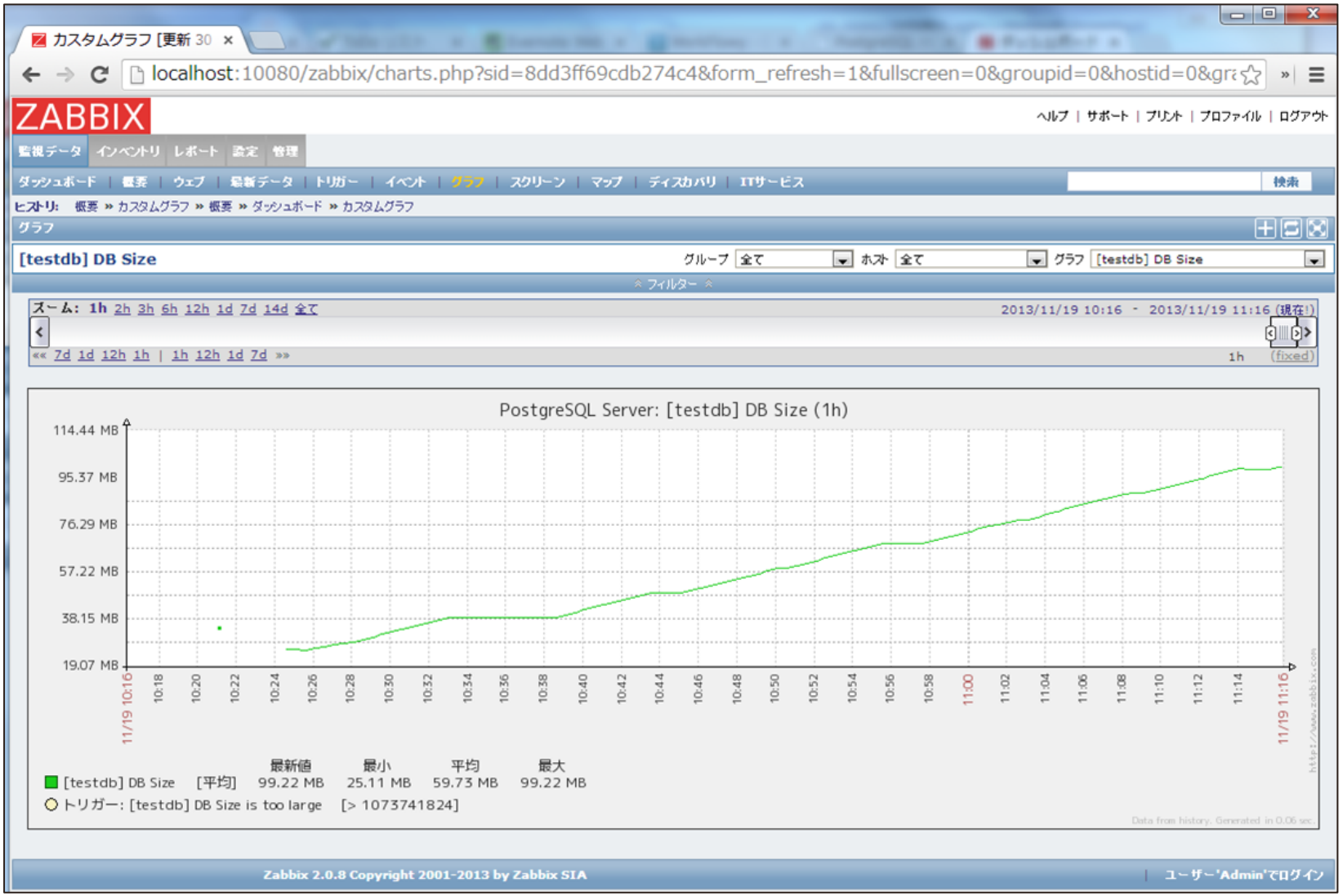
pg_monzで監視できること

TIS IT Holdings Group | Go Beyond

- PostgreSQLサーバの状態
  - サーバ自体の死活状況
  - 接続数, 接続状態,トランザクション量
  - ログのエラーメッセージ
  - チェックポイントの実行状況
- データベースの状態
  - 容量と不要領域の回収率
  - データベース, テーブル単位の稼働状況
  - キャッシュヒット率

基本的なPostgreSQLサーバの監視

監視イメージ(データベース容量)



ここまでは Version 1.0 でもできました。

ここからは Version 2.0 で可能となった「PostgreSQLクラスタ監視」について紹介します。

pg_monzで監視できること

- ストリーミングレプリケーションの状態
  - どのサーバがPrimary/Standbyか一目瞭然

アイテム	pgsql01	pgsql02	pgsql03
PostgreSQL service is running	Up (1)	Up (1)	Up (1)
Primary Server	Up (1)	Down (0)	Down (0)
Standby Server	Down (0)	Up (1)	Up (1)

  - Primary障害時のフェイルオーバ発生の様子をイベントとして通知
  - Standbyへのデータ伝搬の遅延状況
  - 同期レプリケーションのプライオリティ

pg_monzで監視できること

- pgpool-IIの状態
  - フロント/バックエンドの接続数
  - バックエンドの稼働状況

名前	最新のチェック時刻	最新の値
pgpool.nodes (9アイテム)		
[ID_0_192.168.1.13_5432] Backend role	2015/06/10 18:00:29	primary
[ID_0_192.168.1.13_5432] Backend status	2015/06/10 18:00:29	2
[ID_0_192.168.1.13_5432] Backend weight	2015/06/10 18:00:29	0.33
[ID_1_192.168.1.14_5432] Backend role	2015/06/10 18:00:29	standby
[ID_1_192.168.1.14_5432] Backend status	2015/06/10 18:00:29	2
[ID_1_192.168.1.14_5432] Backend weight	2015/06/10 18:00:29	0.33
[ID_2_192.168.1.15_5432] Backend role	2015/06/10 18:00:29	standby
[ID_2_192.168.1.15_5432] Backend status	2015/06/10 18:00:29	2
[ID_2_192.168.1.15_5432] Backend weight	2015/06/10 18:00:29	0.33

  - クエリキャッシュ使用時のキャッシュ状況
  - watchdog構成での仮想IP保持状況

pg_monzで監視できること

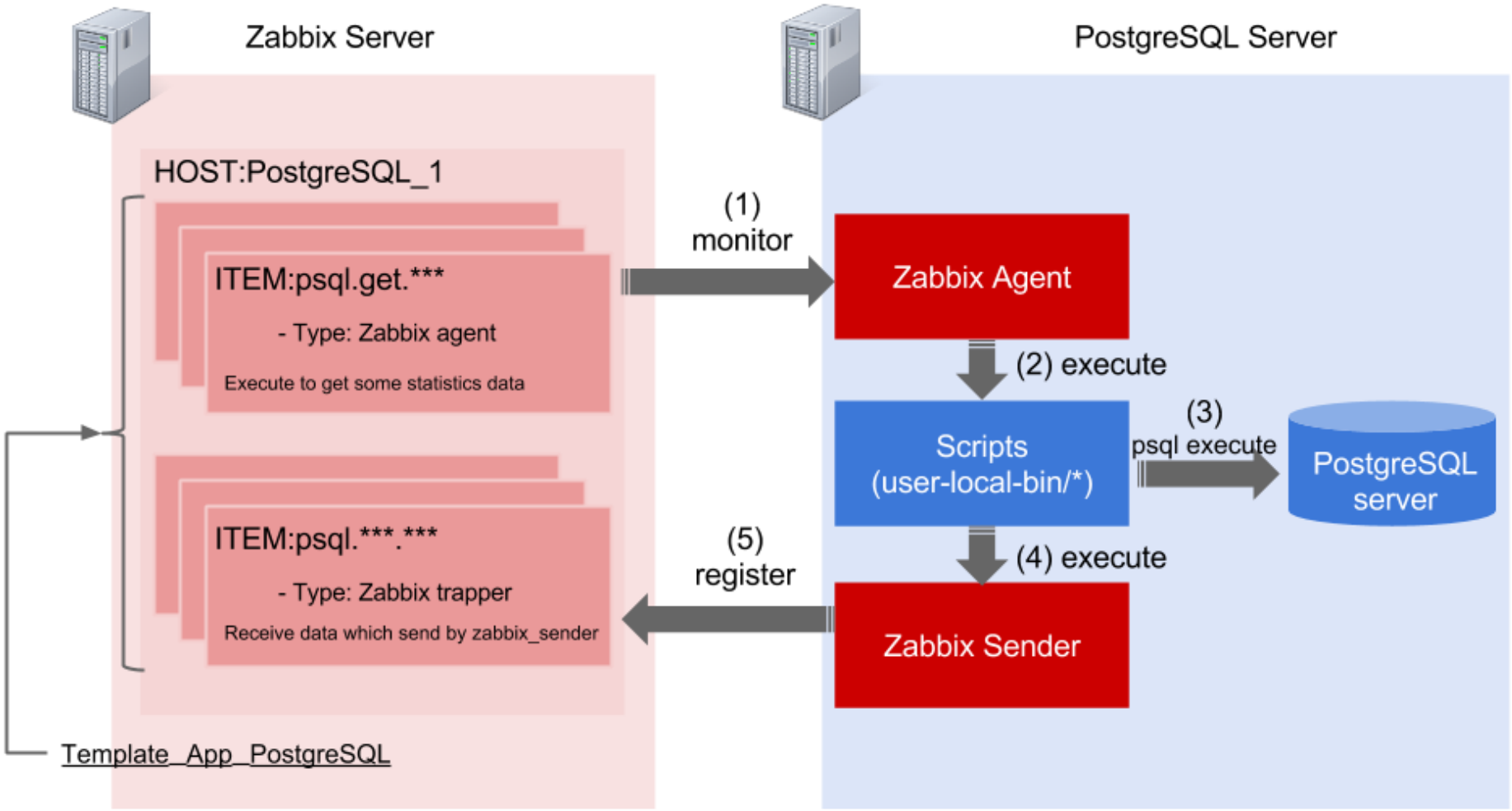
- クラスタとしてのサービス状態
  - Activeなpgpool-IIが複数存在する。
    もしくは1つも存在しない。

時間	ホスト	説明	ステータス	深刻度
2015/06/10 08:55:24	PostgreSQL Cluster	pgpool-II split-brain is occurring on pgpool group	障害	重度の障害
2015/06/10 08:50:58	pgpool01	pgpool-II server promote to master with activating delegate_ip on pgpool01	正常	情報
2015/06/10 08:47:13	pgpool01	pgpool-II server promote to master with activating delegate_ip on pgpool01	障害	情報

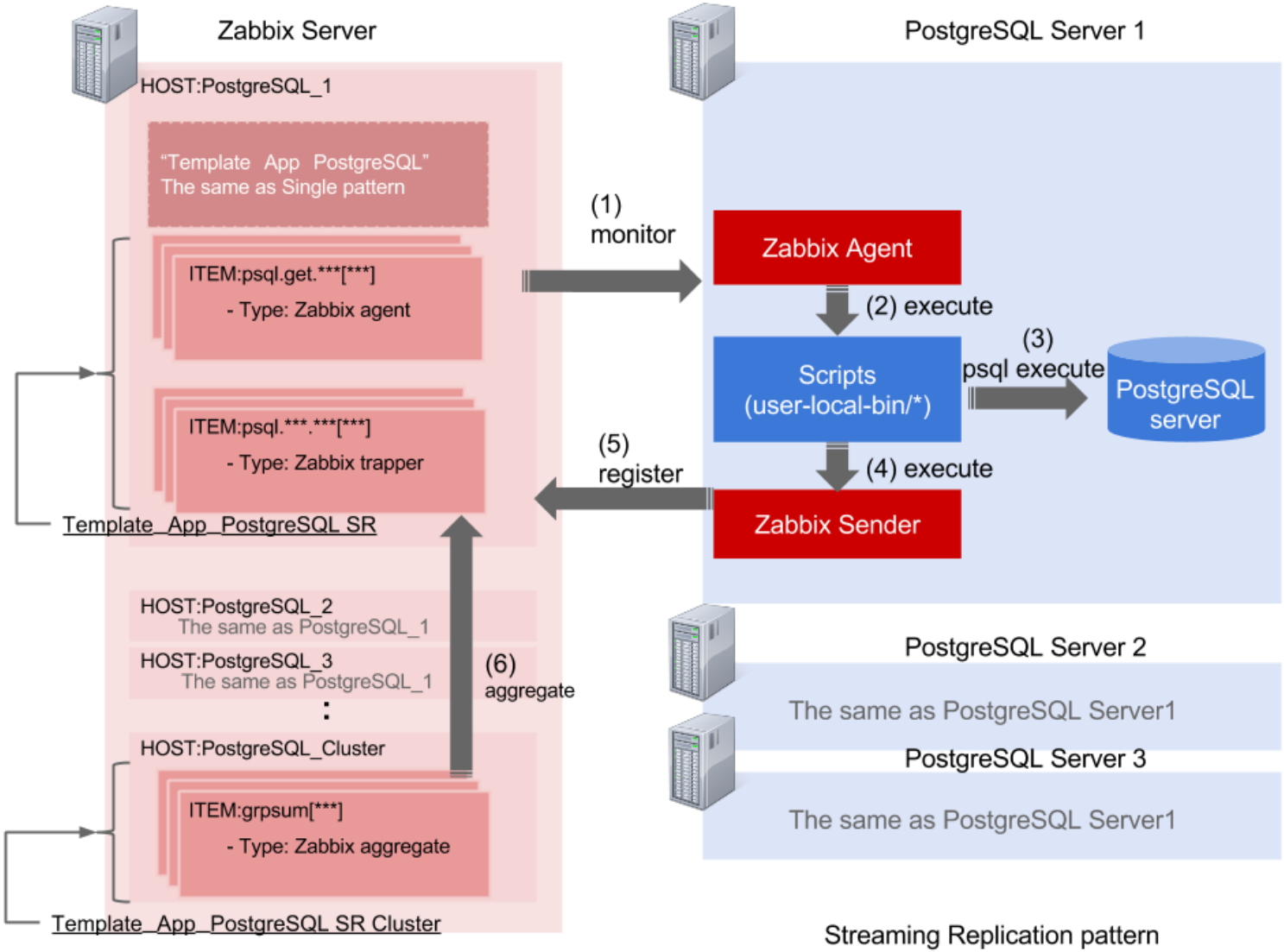
  - ストリーミングレプリケーションのPrimary
    が複数存在する。もしくは1つも存在しない。
  - 同期レプリケーションのスタンバイサーバが
    全てダウンしている（更新が不可能）。

サーバだけの監視では気付きにくい
クラスタ特有の障害事象に対応

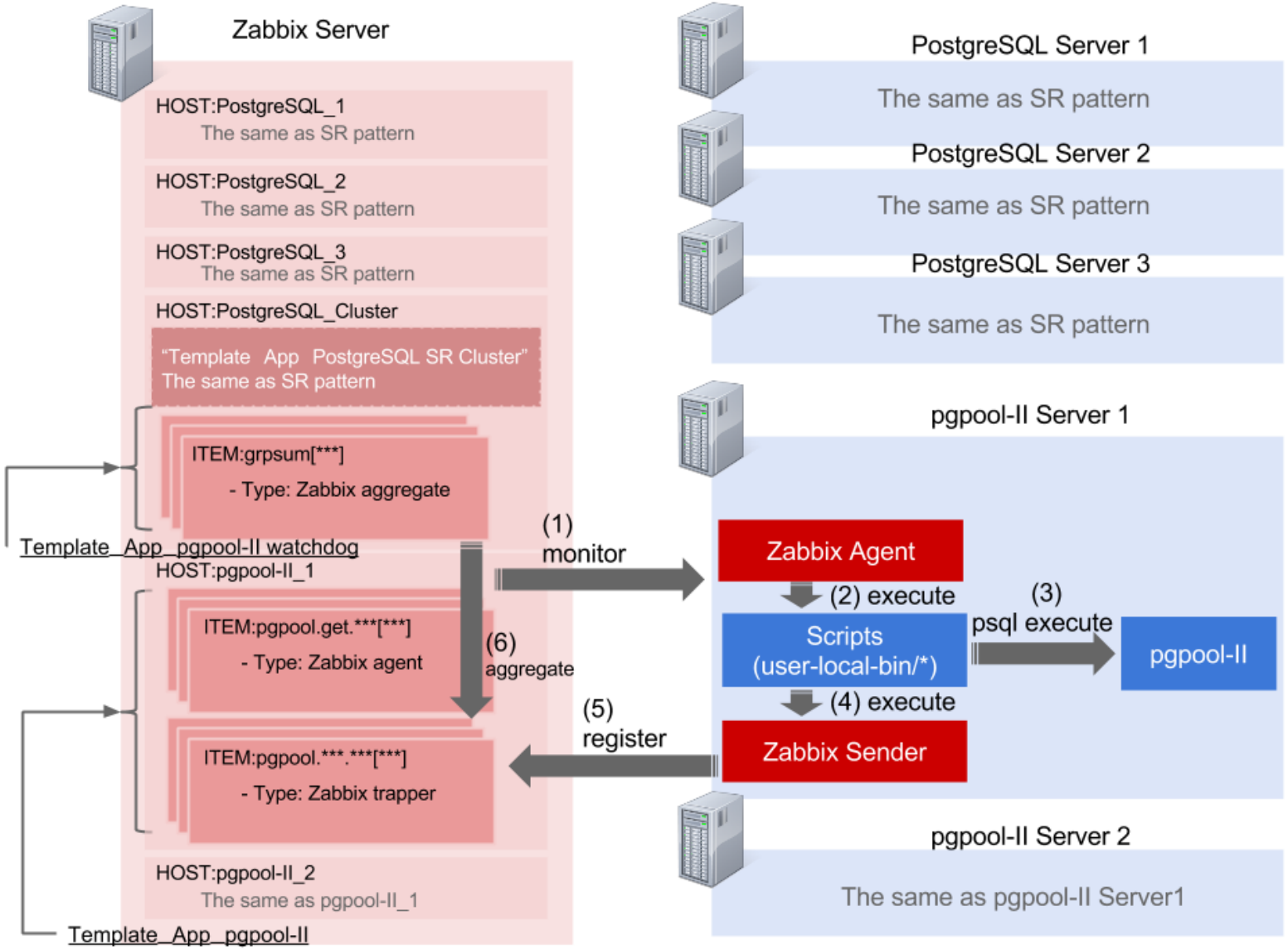
pg_monzのしくみ(Single)



pg_monzのしくみ(SR)



pg_monzのしくみ(pgpool-II)



pg_monzの特徴

TIS IT Holdings Group | Go Beyond

- 導入が容易
  - Zabbix標準の機能を活用
  - 特別なモジュールや設定変更は不要
- 導入手順
  - 監視先サーバへ設定ファイル,スクリプトを配置
  - Zabbixサーバへテンプレートをインポート
  - Zabbixのホストにテンプレートを割当て

これだけでPostgreSQL監視がスタート

pg_monzの特徴

- 環境に合わせた監視設定を自動で生成
  - Zabbixローレベルディスカバリ機能を活用
- 活用シーン
  - データベースの監視
  - Standbyサーバへのレプリケーション監視
  - pgpool-IIバックエンドサーバの状態監視

環境が変化した際の設定作業を削減

監視に使うPostgreSQLの機能

TIS IT Holdings Group | Go Beyond

- 稼働統計情報
- ログ

ログからわかること

TIS IT Holdings Group | Go Beyond

- 何らかの異常が発生したこと
  - 深刻なレベル(PANIC,FATAL)から特に影響ないレベル(INFO)まで
- 処理されたSQL
  - 設定時間を超過したSQL(スロークエリ)
  - 監査目的で全てのSQLを出力することも可能
- チェックポイント、自動VACUUMの実行
- デッドロックの発生 etc

PostgreSQLのログの難点

TIS IT Holdings Group | Go Beyond

- 古いログのメンテ機能がない
  - ログローテーションは可能
    - log_rotation_age, log_rotation_size
  - クリーンアップ機能はない
    - logrotate, cron 等を駆使してください
- 障害メッセージ出現時に警告できない
  - 別途、監視ツールを使って監視する
  - Zabbix + pg_monz

PostgreSQLのログの難点

TIS IT Holdings Group | Go Beyond

- ログをルーティングする機能がない
  - なんでも1つのログファイルに吐き出す。
  - エラーレベルやカテゴリで分けたりできないので、ログから見たい情報を探すのが面倒
  - PostgreSQLサーバが起動できないような深刻なエラーメッセージとスロークエリログを一緒にしないで。。。

pg_monz(Zabbix)でログ監視

TIS IT Holdings Group | Go Beyond

- 深刻なエラーレベルのログを検知して、警告を出すことはできる
  - PANIC, FATAL, ERRORの文字列を検知
- Zabbixで細かなログ監視はやりづらい
  - 特定の文字列が含まれるか、しかわからない
  - 細かな条件を指定した監視はできない

PostgreSQLのログを活用

TIS IT Holdings Group | Go Beyond

- ログにしか出せない情報を性能分析/チューニング用途で有効に使えたら
  - スロークエリのSQL文字列、処理時間
  - チェックポイントの実行時間
  - デッドロックの発生状況
  - ロック待ちで長時間かかったプロセス情報

ログを分析しやすい形に加工したい

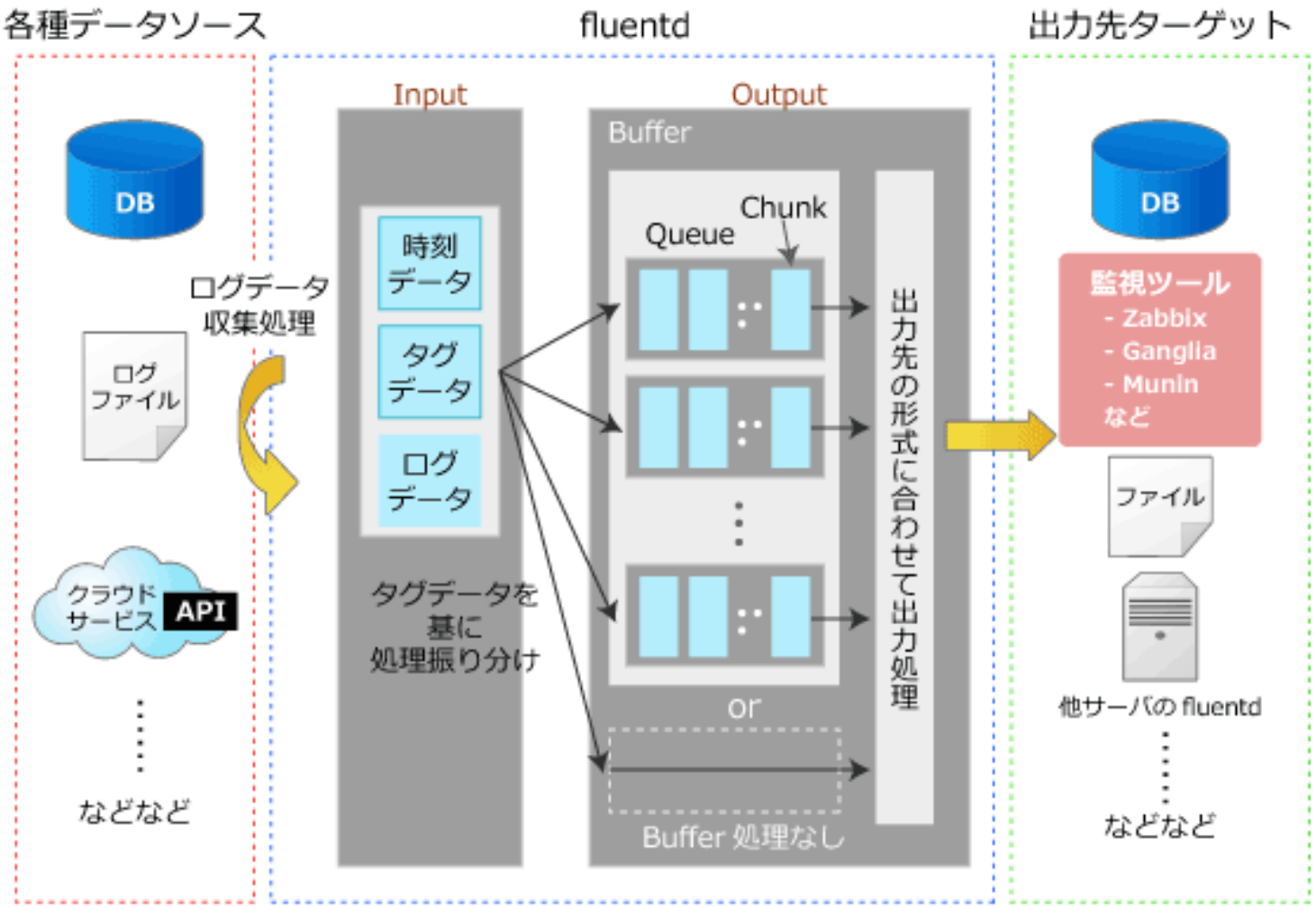
面倒なことは Fluentd に任せませんか？

Fluentd

TIS IT Holdings Group | Go Beyond

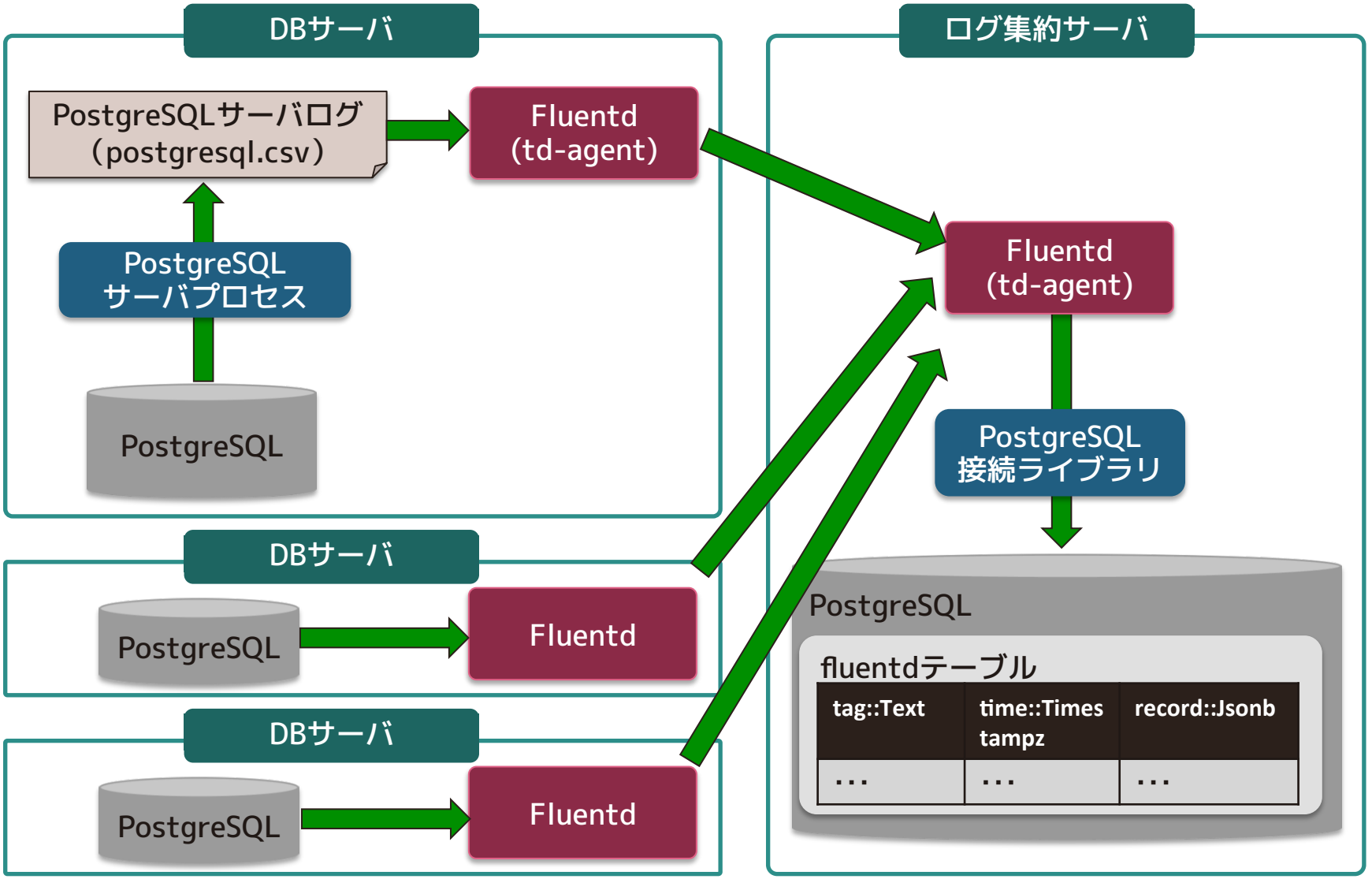
- Fluentdとは
  - 軽量でプラガブルなログ収集ツール
  - Apache License Version 2.0
- Fluentdの特徴
  - ログ管理を3つの層に分けて管理
    - インプット、バッファ、アウトプット
  - 各層がプラガブルなアーキテクチャ
    - 用途に応じたプラグインを追加するだけで使える

Fluentdの動作のしくみ

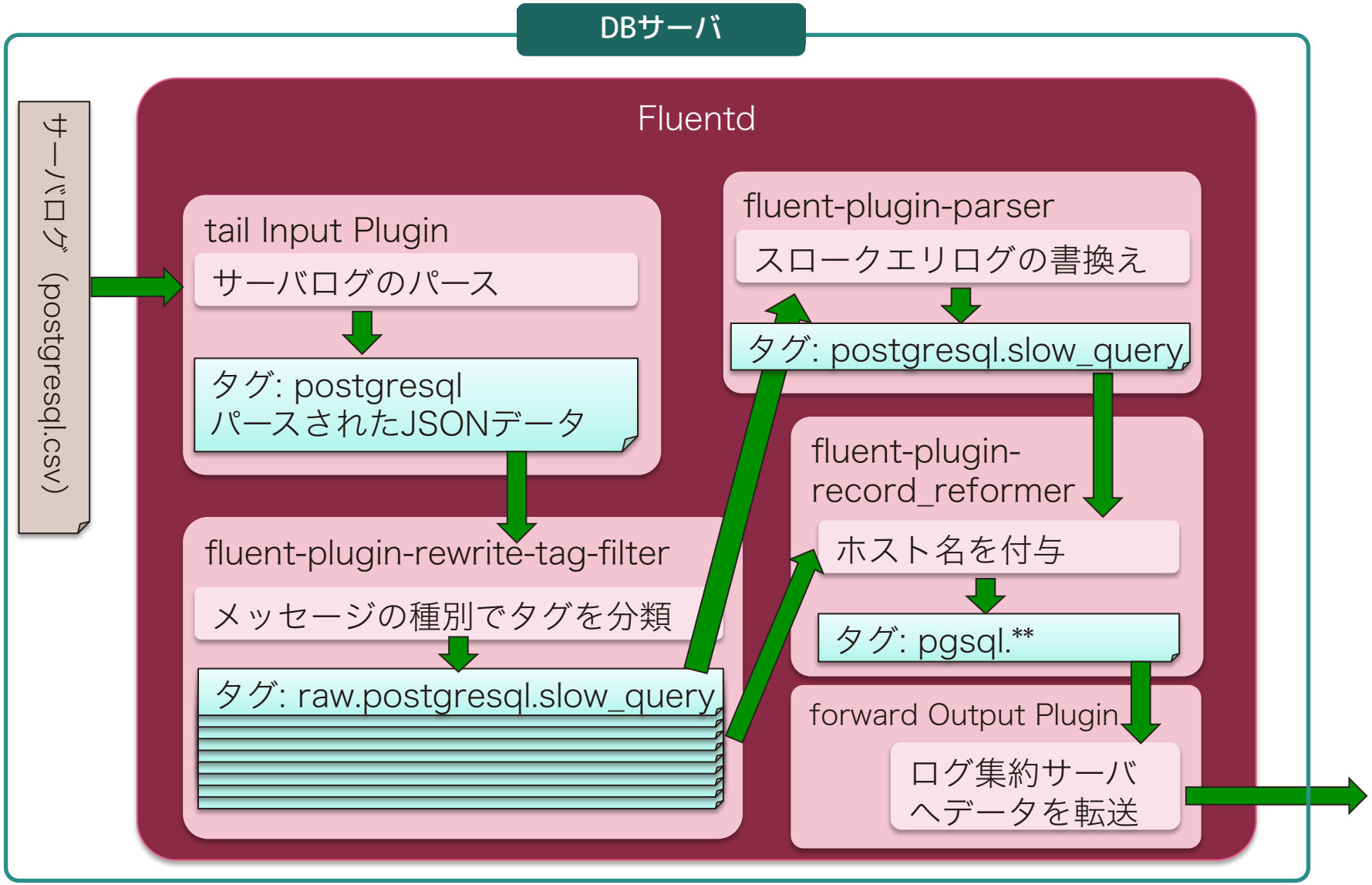


引用元: http://www.atmarkit.co.jp/ait/articles/1402/06/news007.html

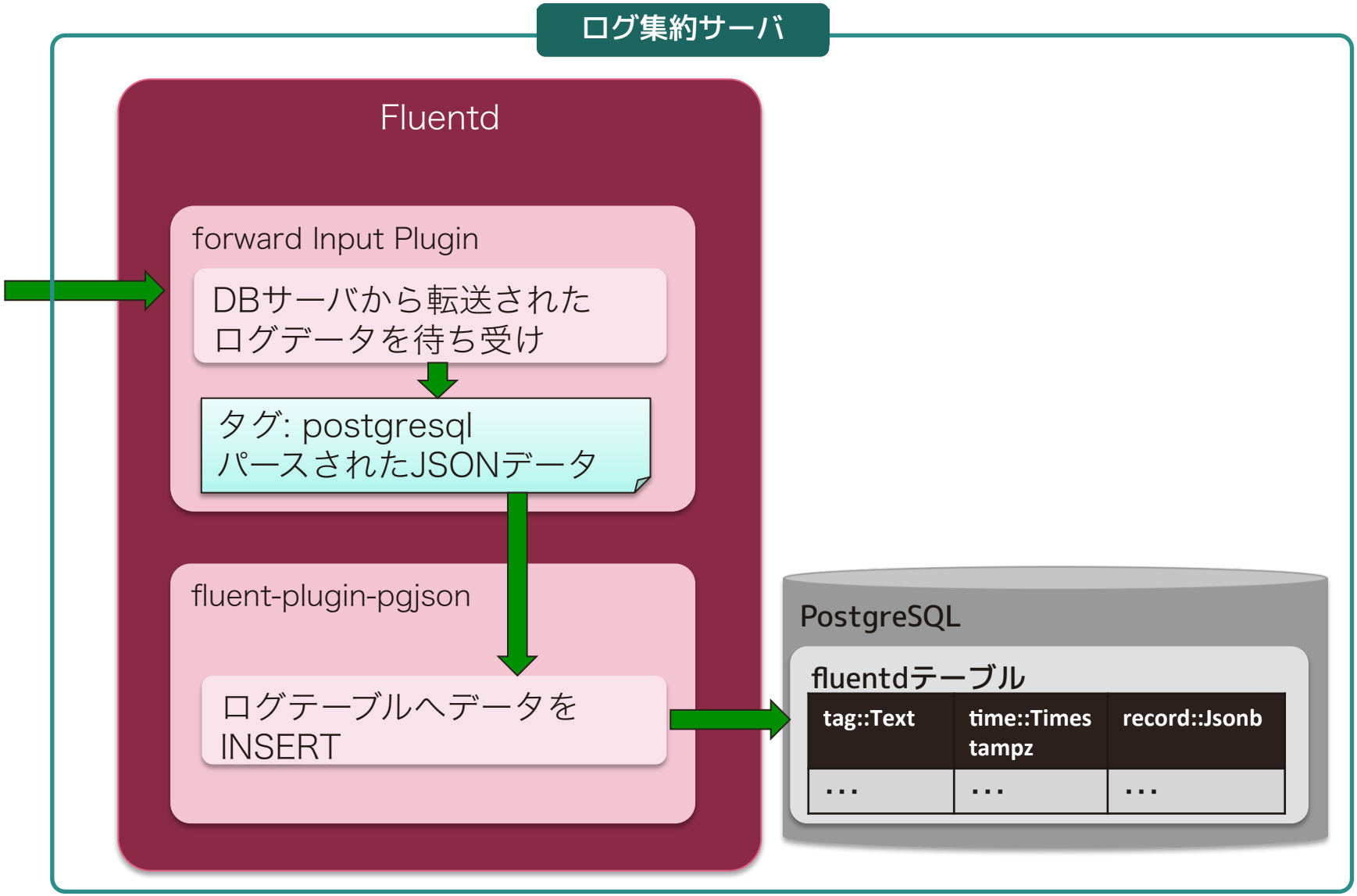
Fluentdでログを収集する(全体像)



Fluentdでの処理を抜粋してみる



Fluentdの処理に注目



tail Input Plugin

TIS IT Holdings Group | Go Beyond

- ログファイルに書き込まれた情報を
  常時取り込むインプットプラグイン
- PostgreSQLのCSVログをパースし、
  以下の様なJSONデータに変換する。

```
{
  "tag": "postgresql",
  "time": "2014-06-30 01:12:02+00",
  {"user_name":"","database_name":"","process_id":"27136","connection_from":"",
   "session_id":"53b0b980.6a00","session_line_num":"10","command_tag":"",
   "session_start_time":"2014-06-30 01:12:32 GMT","virtual_transaction_id":"",
   "transaction_id":"0","error_severity":"LOG","sql_state_code":"00000",
   "message":"checkpoint starting: xlog","detail":"","hint":"","internal_query":"",
   "internal_query_pos":"","context":"","query":"","query_pos":"","location":"",
   "application_name":""}
}
```

fluent-plugin-rewrite-tag-filter

TIS Go Beyond

- データ内容を条件に任意のタグに書き換えるプラグイン
- ログのメッセージ内容でタグを分別
  - Fluentdの設定

```
<match     postgresql>
  type       rewrite_tag_filter
  rewriterule1    message    ^duration:                          raw.postgresql.slow_query
  rewriterule2    message    ^checkpoints\sare\soccurring\stoo\sfrequently
                                                                 postgresql.checkpoints.frequently
  rewriterule3    message    ^checkpoint\sstarting:                  postgresql.checkpoint.start
  rewriterule4    message    ^checkpoint\scomplete:                  postgresql.checkpoint.complete
  rewriterule5    message    ^automatic                          postgresql.vacuum
  rewriterule6    message    ^temporary file:                    postgresql.tempfiles
  rewriterule7    message    ^process.*detected\sdeadlock            postgresql.deadlock
  rewriterule8    message    ^process.*(still waiting|acquired)      postgresql.lockwait
  rewriterule9    message    .*                                  postgresql.others
</match>
```

fluent-plugin-parser

TIS IT Holdings Group | Go Beyond

- データの任意のフィールドをさらに分解するプラグイン
- スロークエリログの処理時間,SQLを分解

```
{
  “tag”: “raw.postgresql.slow_query”,
  “time”: “2014-06-30 01:12:02+00”,“user_name”:“postgres”,“database_name”:“fluentd”,
  …, “message”:“duration: 4004.423 ms  statement: select pg_sleep(4);”, … ,
  "application_name":"psql"}
}
```

```
{
  "tag": "postgresql.slow_query",
  "time": "2014-06-30 01:12:02+00”,“user_name”:“postgres”,“database_name”:“fluentd”,
  …,“message”:“duration: 4004.423 ms  statement: select pg_sleep(4);”… ,
  "application_name":"psql”,"duration":"4004.423", "statement":"select pg_sleep(4);"}
}
```

fluent-plugin-record_reformer

TIS Go Beyond

- データにちょっとした処理をして
  フィールドを追加するプラグイン
- ログメッセージにホスト名を付与

```
{
  "tag": ”postgresql.checkpoint.start",
  "time": "2014-06-30 01:12:02+00",
  {"user_name":"","database_name":"","process_id":"27136","connection_from":"",
   …,"application_name":""}
}
```

```
{
  "tag": ”pgsql.checkpoint.start",
  "time": "2014-06-30 01:12:02+00",
  {"user_name":"","database_name":"","process_id":"27136","connection_from":"",
   …,"application_name":"","hostname":"pgsql01"}
}
```

fluent-plugin-pgjson

TIS IT Holdings Group | Go Beyond

- データをPostgreSQLのテーブルに出力するアウトプットプラグイン
  - 出力先にJSON型のカラムを利用
- Fluentdの設定
  - pgsqlで始まるタグのデータをfluentdデータベースのfluentdテーブルへINSERT

```
<match {postgresql.**}>
  type     pgjson
  host     localhost
  port     5432
  sslmode      prefer
  database     fluentd
  table    fluentd
  user     postgres
  password     postgres
  time_col     time
  tag_col tag
  record_col   record
</match>
```

fluent-plugin-pgjson

TIS IT Holdings Group | Go Beyond

- ログデータを格納するテーブル
  - fluentdデータベースにあらかじめテーブルを作成しておく。

```
CREATE TABLE fluentd (
    tag         Text,
    time        Timestamptz,
    record      Jsonb
);
```

PostgreSQLのJSON

TIS IT Holdings Group | Go Beyond

- PostgreSQLのデータ型として利用可能
  - 9.4ではJSON型, JSONB型が選択可
  - 詳しくは別セミナーへ
- スキーマレスなデータの格納先に使える
  - Fluentdで収集したログデータをJSON型のカラムに格納すれば、フォーマットが非定型なログデータでもSQLで検索／分析できる。
- JSONの一部分だけを更新できない。
  - 「ログの蓄積」は代表的なユースケース

収集したログをSQLで検索する

TIS IT Holdings Group | Go Beyond

- 深刻度がERRORのメッセージを検索

```
# SELECT time, record#>>'{user_name}' as user, record#>>'{database_name}' as
    database, record#>>'{error_severity}' as severity, record#>>'{message}' as
    message, record#>>'{query}' as query, record#>>'{query_pos}' as pos
  FROM  fluentd WHERE record#>>'{error_severity}'  =  'ERROR';

-[ RECORD 1 ]----------------------------------------------------------------------------
time          |  2014-06-30 09:49:12+00
user          |  postgres
database      |  fluentd
severity      |  ERROR
message       |  column ""datname"" does not exist
query         |  select count(*) from fluentd where tag = 'postgresql.slow_query'
and datname = 'postgres'
pos           |  70
```

収集したログをSQLで検索する

TIS IT Holdings Group | Go Beyond

- スロークエリを検索（時間が長い10件）

```
# SELECT time, record#>>'{hostname}' as host, record#>>'{duration}' as
    duration, record#>>'{statement}' as statement
  FROM fluentd WHERE tag = 'pgsql.slow_query'
  ORDER BY (record#>'{duration}') desc LIMIT 10;

              time              |   host    |  duration  |      statement
--------------------------------+-----------+------------+-------------------------
 2015-06-07 08:33:48+00 | pgsql02 | 2004.107 | select pg_sleep(2);
 2015-06-07 02:25:08+00 | pgsql01 | 2003.866 | select pg_sleep(2);
 2015-06-07 08:45:48+00 | pgsql03 | 2003.767 | select pg_sleep(2);
 2015-06-07 02:25:10+00 | pgsql01 | 2003.621 | select pg_sleep(2);
 2015-06-07 08:45:51+00 | pgsql03 | 2003.461 | select pg_sleep(2);
 2015-06-07 08:33:51+00 | pgsql02 | 2003.376 | select pg_sleep(2);
 2015-06-07 02:25:02+00 | pgsql01 | 2003.374 | select pg_sleep(2);
 2015-06-07 08:33:54+00 | pgsql02 | 2003.247 | select pg_sleep(2);
 2015-06-07 08:45:46+00 | pgsql03 | 2002.880 | select pg_sleep(2);
 2015-06-07 02:25:05+00 | pgsql01 | 2002.092 | select pg_sleep(2);
```

収集したログをSQLで検索する

TIS IT Holdings Group | Go Beyond

- チェックポイント頻発の警告メッセージ

```
# SELECT time, record#>>'{message}' as message  FROM  fluentd
  WHERE tag = 'pgsql.checkpoints.frequently';

             time              |                                  message
-------------------------------+-----------------------------------------------------------------------
 2014-06-30 01:30:21+00  |  checkpoints are occurring too frequently (8 seconds apart)
 2014-06-30 01:30:29+00  |  checkpoints are occurring too frequently (8 seconds apart)
 2014-06-30 01:30:37+00  |  checkpoints are occurring too frequently (8 seconds apart)
 2014-06-30 01:30:46+00  |  checkpoints are occurring too frequently (9 seconds apart)
 2014-06-30 01:30:55+00  |  checkpoints are occurring too frequently (9 seconds apart)
 2014-06-30 01:31:04+00  |  checkpoints are occurring too frequently (9 seconds apart)
 2014-06-30 01:31:14+00  |  checkpoints are occurring too frequently (10 seconds apart)
 2014-06-30 01:31:23+00  |  checkpoints are occurring too frequently (9 seconds apart)
```

収集したログをSQLで検索する

- チェックポイントの実行状況

```
# SELECT time, record#>>'{message}' as message
  FROM fluentd WHERE tag like 'pgsql.checkpoint.%';

             time             |                       message
----------------------- --------------
+------------------------------------------------------------------------------------------------------------
 2014-06-30 01:17:32+00  |  checkpoint starting: time
 2014-06-30 01:17:33+00  |  checkpoint complete: wrote 5 buffers (0.0%); 0
transaction log file(s) added, 0 removed, 0 recycled; write=0.403 s, sync=0.463 s,
total=0.885 s; sync files=3, longest=0.460 s, average=0.154 s
 2014-06-30 01:27:32+00  |  checkpoint starting: time
 2014-06-30 01:27:32+00  |  checkpoint complete: wrote 1 buffers (0.0%); 0
transaction log file(s) added, 0 removed, 0 recycled; write=0.000 s, sync=0.002 s,
total=0.022 s; sync files=1, longest=0.002 s, average=0.002 s
 2014-06-30 01:30:13+00  |  checkpoint starting: xlog
 2014-06-30 01:30:16+00  |  checkpoint complete: wrote 1006 buffers (6.1%); 0
transaction log file(s) added, 0 removed, 0 recycled; write=2.019 s, sync=0.272 s,
total=2.396 s; sync files=19, longest=0.161 s, average=0.014 s
 2014-06-30 01:30:21+00  |  checkpoint starting: xlog
 2014-06-30 01:30:24+00  |  checkpoint complete: wrote 1894 buffers (11.6%); 0
transaction log file(s) added, 0 removed, 1 recycled; write=2.223 s, sync=0.209 s,
total=2.518 s; sync files=9, longest=0.148 s, average=0.023 s
```

収集したログをSQLで検索する

- ロック待ちが発生したセッション情報

```
# SELECT time, record#>>'{message}' as message FROM fluentd WHERE tag = 'pgsql.lockwait';

            time            |                    message
------------------------------------
+------------------------------------------------------------------------------------------------------
 2014-06-29 12:04:57+00 |
process 13505 still waiting for RowExclusiveLock on relation 16385 of database 16384 after
1001.725 ms
 2014-06-29 12:05:12+00 |
process 13505 acquired RowExclusiveLock on relation 16385 of database 16384 after
16021.973 ms
 2014-06-29 12:06:18+00 |
process 13397 still waiting for RowExclusiveLock on relation 16385 of database 16384 after
1000.910 ms
 2014-06-29 12:06:29+00 |
process 13397 acquired RowExclusiveLock on relation 16385 of database 16384 after
11992.269 ms
 2014-06-29 12:06:52+00 |
```

収集したログをSQLで検索する

TIS IT Holdings Group | Go Beyond

- 一時ファイルの生成情報

```
# SELECT time, record#>>'{message}' as message, record#>>'{query}' as query
   FROM fluentd WHERE tag = 'pgsql.tempfiles';

-[ RECORD 1 ]----------------------------------------------------------------
time    | 2014-07-02 09:34:33+00
message | temporary file: path ""base/pgsql_tmp/pgsql_tmp8184.0"", size
10723328
query   | explain analyze select * from pgbench_accounts order by bid;
```

収集したログをSQLで検索する

TIS IT Holdings Group | Go Beyond

- デッドロックの発生情報

```
# SELECT  time, record#>>'{database_name}' as db, record#>>'{command_tag}'
      as command, record#>>'{message}' as message, record#>>'{query}' as
      query
   FROM fluentd WHERE tag = 'pgsql.deadlock';

-[ RECORD1 ]----------------------------------------------------------------------------------
time         | 2014-07-02 09:45:02+00
db           | fluentd
command      | LOCK TABLE waiting
message      | process 8184 detected deadlock while waiting for ExclusiveLock
on relation 16394 of database 16384 after 1000.235 ms
query        | LOCK TABLE pgbench_history IN EXCLUSIVE MODE;
```

Let's try!

pg_monzの入手、問合せ先

TIS IT Holdings Group | Go Beyond

- 入手先や使い方
  - http://pg-monz.github.io/pg_monz/
- 問い合わせ
  - pg_monz ユーザーグループ
  - pg_monz@googlegroups.com

参考情報

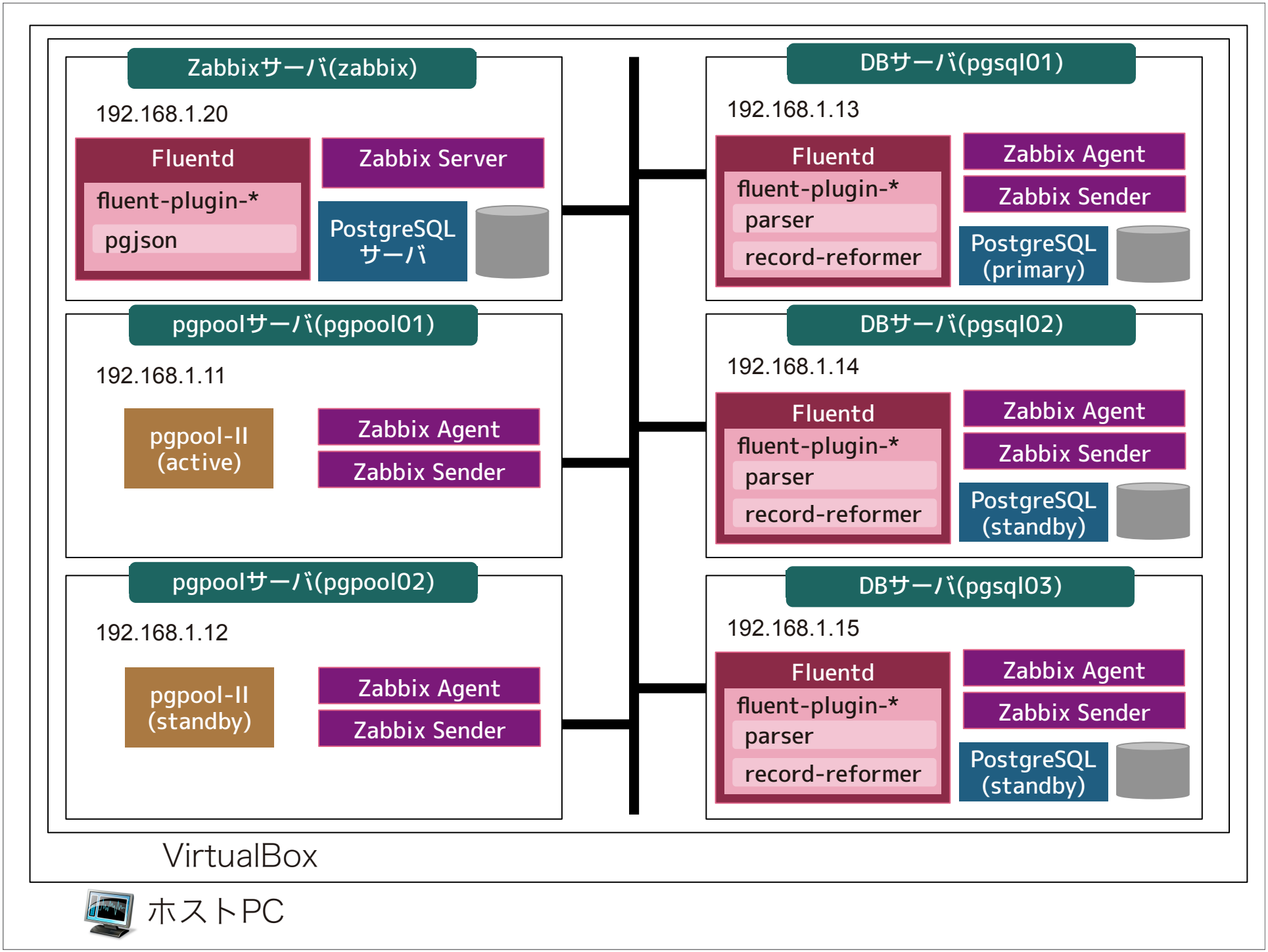
TIS IT Holdings Group | Go Beyond

- 明日から使えるPostgresql運用管理テクニック(監視編)
  - http://www.slideshare.net/kasaharatt/postgre-sql-26186128
- Let's Postgres - 稼動統計情報を活用しよう(1)
  - http://lets.postgresql.jp/documents/technical/statistics/1
- 今さら聞けないfluentd〜クラウド時代のログ管理入門
  - http://www.atmarkit.co.jp/ait/articles/1402/06/news007.html
- PostgreSQLのログをfluentdで回収する設定
  - http://chopl.in/blog/2013/06/07/postgresql_csv_log_with_fluentd.html

試行環境の構築手順まとめました

TIS IT Holdings Group | Go Beyond

- Qiitaに構築手順をアップしました。
  - pg_monz,Fluentdの試行環境を構築する
  - http://qiita.com/nakaniko/items/00c986f1fa867a627f53
- VirtualBox,Vagrantを入れたPCがあれば試すことができます。
  - あと、ほんのちょっとのやる気と
    ほんのちょっとのメモリが必要かも。。。


Zabbixサーバ(zabbix)
192.168.1.20
Fluentd
fluent-plugin-*
pgjson
Zabbix Server
PostgreSQL
サーバ
pgpoolサーバ(pgpool01)
192.168.1.11
pgpool-II
(active)
Zabbix Agent
Zabbix Sender
pgpoolサーバ(pgpool02)
192.168.1.12
pgpool-II
(standby)
Zabbix Agent
Zabbix Sender
DBサーバ(pgsql01)
192.168.1.13
Fluentd
fluent-plugin-*
parser
record-reformer
Zabbix Agent
Zabbix Sender
PostgreSQL
(primary)
DBサーバ(pgsql02)
192.168.1.14
Fluentd
fluent-plugin-*
parser
record-reformer
Zabbix Agent
Zabbix Sender
PostgreSQL
(standby)
DBサーバ(pgsql03)
192.168.1.15
Fluentd
fluent-plugin-*
parser
record-reformer
Zabbix Agent
Zabbix Sender
PostgreSQL
(standby)
VirtualBox
ホストPC

まとめ

- 運用で使えるPostgreSQLの標準機能
  - 稼働統計情報とログ
- 標準機能だけではいろいろダルい
  - 稼働統計情報：SQLを書くのがダルい
  - ログ：分析に必要な情報を取出すのがダルい
- ツールを組合せて楽をしよう
  - pg_monz で楽々監視
  - fluentd で楽々ログ管理
- pg_monzへのフィードバックを
  お待ちしています。

したっけね～！

※使い方がまちがっていたらすみません。


TIS
IT Holdings Group
Go Beyond