

ZabbixでPostgreSQLの監視を行おう ～pg_monz のご紹介～

SRA OSS, Inc. 日本支社
盛 宣陽

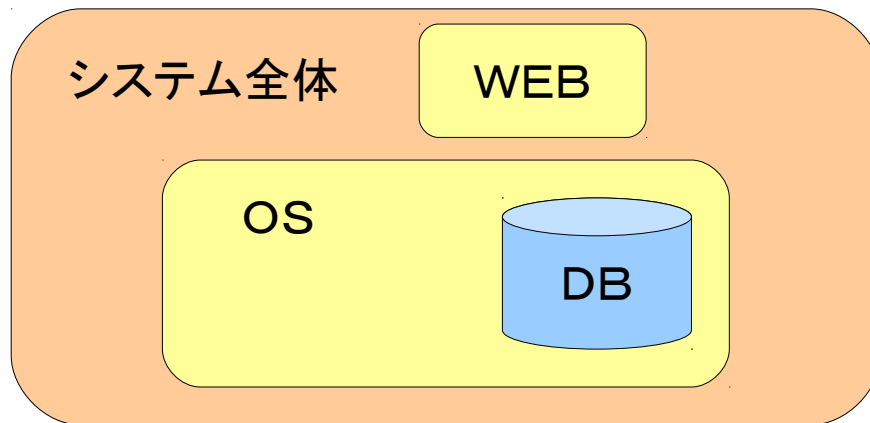
PostgreSQLの課題

- DBとしての基本機能、性能は商用DBと比べても引けをとらない
- 運用面には課題あり
 - どのようにして運用するのか？
 - 効果的な監視方法は？
 - シングル構成、クラスタ構成の場合は？

統合監視ソフトウェアの必要性

- DBの監視といっても

- DB内部状態
- OSの状態
- アプリケーションの状態
- 過去と現在の比較を行って将来を予測できるようなデータの蓄積
- などなど



様々な監視データと突き合わせて監視を行うためには
統合監視ソフトウェアがあると便利

統合監視ソフトウェアの問題点

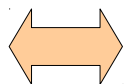
- 設定が大変
 - 知識と人手が必要
 - 特にデータベース監視では、データベース名やテーブル名などシステム固有な情報ありすぎる

~~複雑なのでプロセス監視、サービスポート監視、ログ監視だけ~~ 

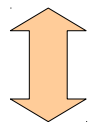
統合監視ソフトウェア Zabbix

- 人気があるオープンソース統合監視ソフトウェア
- 特徴
 - データ収集
 - 収集したデータの保存、傾向分析
 - アラート機能
 - 収集したデータを元にメールなどで障害通知
 - 可視化
 - Webインターフェースによるグラフィカル表示
 - 拡張監視
ZabbixエージェントのUserParameterによる監視の拡張
 - 設定作業の自動化
 - テンプレートの再利用
 - 監視対象の発見(ディスカバリ)機能
 - 監視対象ホストの内部情報取得機能(ローレベルディスカバリ)機能

Zabbix Server



データ収集 RDBMSに保管
ホストのディスクバリエーション
アラート通知



Zabbix Proxy

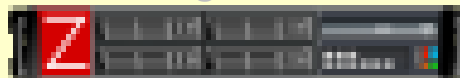


遠隔地の監視



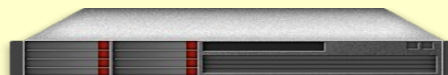
監視対象ホスト

Zabbix Agent: エージェントによる監視



ビルトイン監視
ログ監視
カスタムスクリプト監視
ローレベルディスクバリエーションで
ホスト固有情報を探索

エージェントレス: 専用機器



SNMP、IPMI、PINGなどで監視

プログラムからコマンド実行



zabbix_sender

pg_monzについて

- ZabbixのPostgreSQL監視テンプレート
- 開発元 TIS, SRA OSS, Inc. 日本支社
- GitHubにてApacheライセンスV2で公開
http://pg-monz.github.io/pg_monz
- 特徴
 - システム固有情報(DB名、テーブル名)を自動で取得
 - ZabbixのLLD(ローレベルディスカバリ)を利用
 - ZabbixのUserParameterを利用
 - PostgreSQL監視内容
死活監視 ログ監視 リソース監視 性能監視
PostgreSQLの稼働統計情報を活用
- 動作環境
Zabbix 2.0以上、PostgreSQL 9.2以上



pg_monz概念図

Zabbix サーバ



PostgreSQL監視
テンプレート

[*pg_monz_template.xml*](#)

LLD

DB名やテーブル名の要求

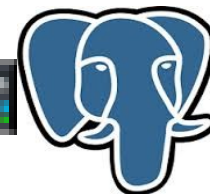
DB名、テーブル名の返却

監視

監視データの要求

監視データの返却

PostgreSQLサーバ



LLDスクリプト

[*find_dbname.sh*](#)

[*find_dbname_table.sh*](#)

Zabbix エージェント

PostgreSQL監視用

UserParameter

(スクリプトによる監視)

[*userparameter_pgsql.conf*](#)

psqlでDBの稼働統計情報に
問い合わせ

pg_monz動作原理

- ZabbixのLLDの利用
 - Zabbix Agentからスクリプトを実行してシステム固有情報をZabbixサーバにLLDマクロとして返却

```
例) postgresデータベースとDB1データベースが存在する場合  
{"data":  
  [{"#{DBNAME}":"postgres"}, {"#{DBNAME}":"DB1"}]  
}
```

- Zabbixサーバでは監視項目に変数を埋め込んでZabbixエージェントに監視要求を出す
({ \$で始まる変数 } => マクロ、 { #で始まる変数 } => LLDマクロ)

例) DBサイズ監視を行うアイテム

```
psql.db_size[{ $PGHOST }, { $PGPORT }, { $PGROLE }, { $PGDATABASE }, { #DBNAME }]
```

pg_monz監視項目 総数

- 1データベースクラスタの監視項目数
22項目
- 1データベースあたりの監視項目数
12項目
- 1テーブルあたりの監視項目数 (デフォルト無効)
14項目

pg_monz監視項目①

- 死活監視
 - postgresプロセス監視 & トリガー
 - SQL応答監視 & トリガー
- ログ監視
 - PANIC,FATAL,ERRORを含むメッセージ監視
- サイズ監視
 - 対象DBの容量 & トリガー & グラフ

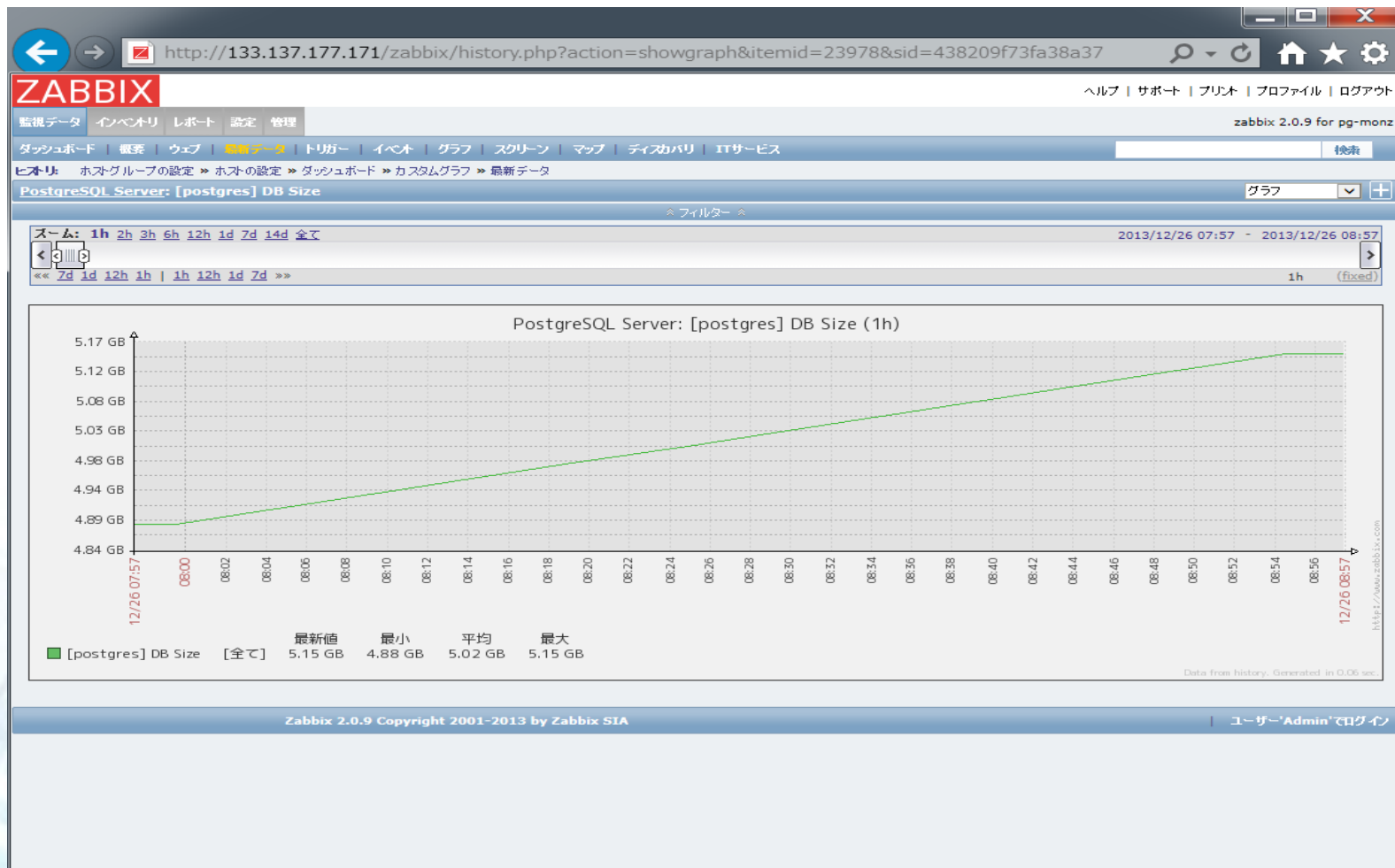


図: データベース別データサイズ

pg_monz監視項目②

- バックエンドプロセス監視
 - 総接続数とプロセス状態の内訳 & トリガー & グラフ
 - SQL処理中プロセス数
 - アイドルプロセス数
 - トランザクション内アイドルプロセス数
 - ロック待ちプロセス数
- チェックポイント実行状況
 - checkpoint_segments/checkpoint_timeout超過による
チェックポイント実行回数 & トリガー & グラフ

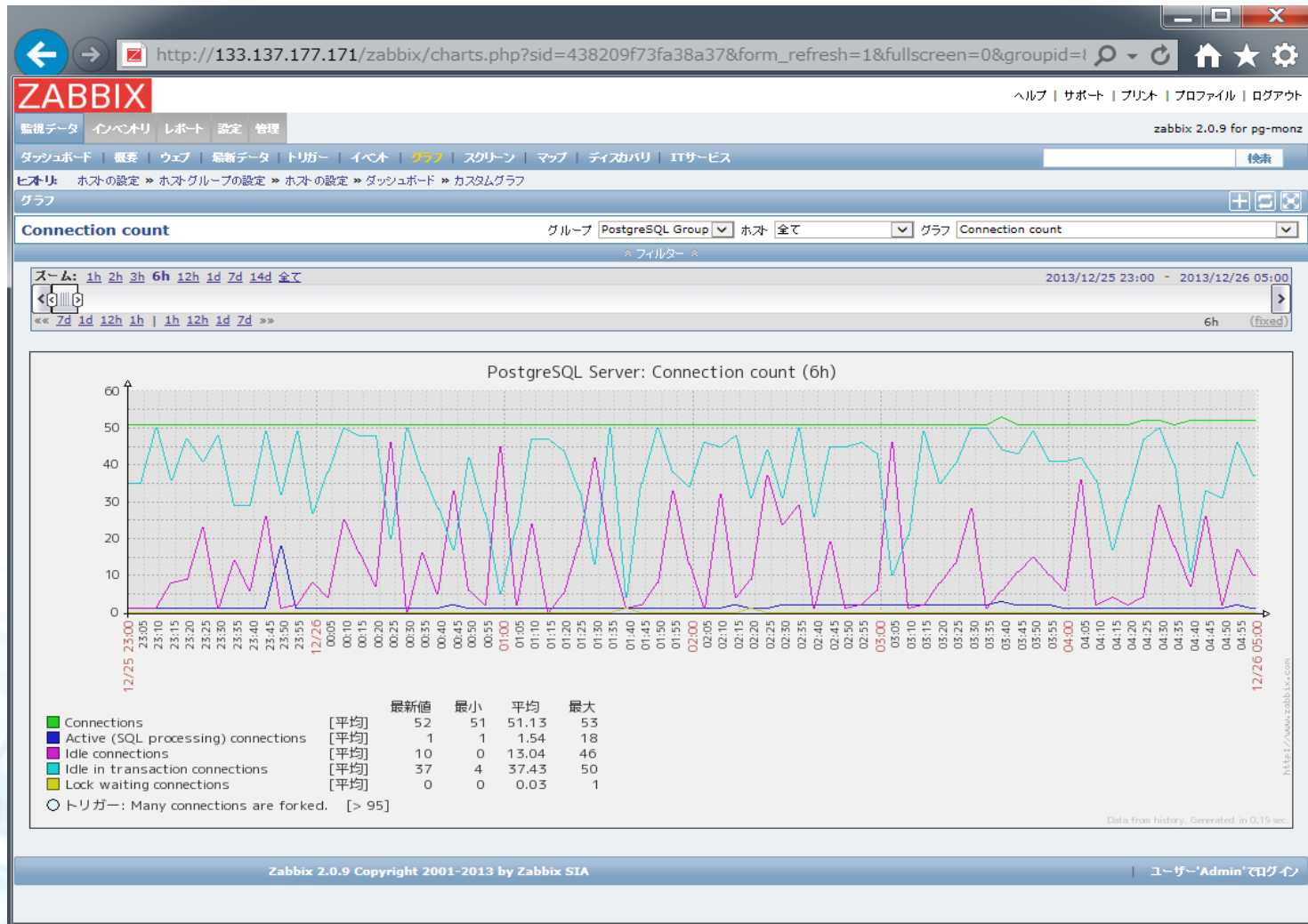


図: 総接続数とプロセス状態の内訳

pg_monz監視項目③

- キャッシュヒット率の監視
 - DB別にキャッシュヒット率の算出 & トリガー & グラフ
- トランザクション処理状況の監視
 - DB別のCOMMIT回数/s, ROLLBACK回数/s & グラフ
- 一時ファイル発生状況の監視
 - DB別一時ファイルの利用量 & トリガー & グラフ
- 滞留バックエンド処理の監視
 - 指定時間経過したクエリ数、SELECT処理数、DML数 & トリガー

pg_monz監視項目④

- テーブル単位の情報(デフォルト無効)
 - vacuum/analyze 実行回数
 - auto vacuum/analyze 実行回数
 - キャッシュヒット率
 - liveタプルとdeadタプル件数
 - シーケンシャルスキャンとインデックススキャン回数

pg_monz 検証

- 検証環境

- Zabbix サーバ、PostgreSQLサーバ共通

Zabbix 2.0.9

CPU 1 core Mem 2G OS CentOS 6.4 Vmware ESXi ゲストOS

DB : PostgreSQL 9.2.5 (Zabbix DBを含む)

- 参考: Vmware ESXi のスペック

バージョン 5.5

製品名 HP DL360p G8

プロセッサ Xeon E5650 2GHz 8Core x2

メモリ 96GB

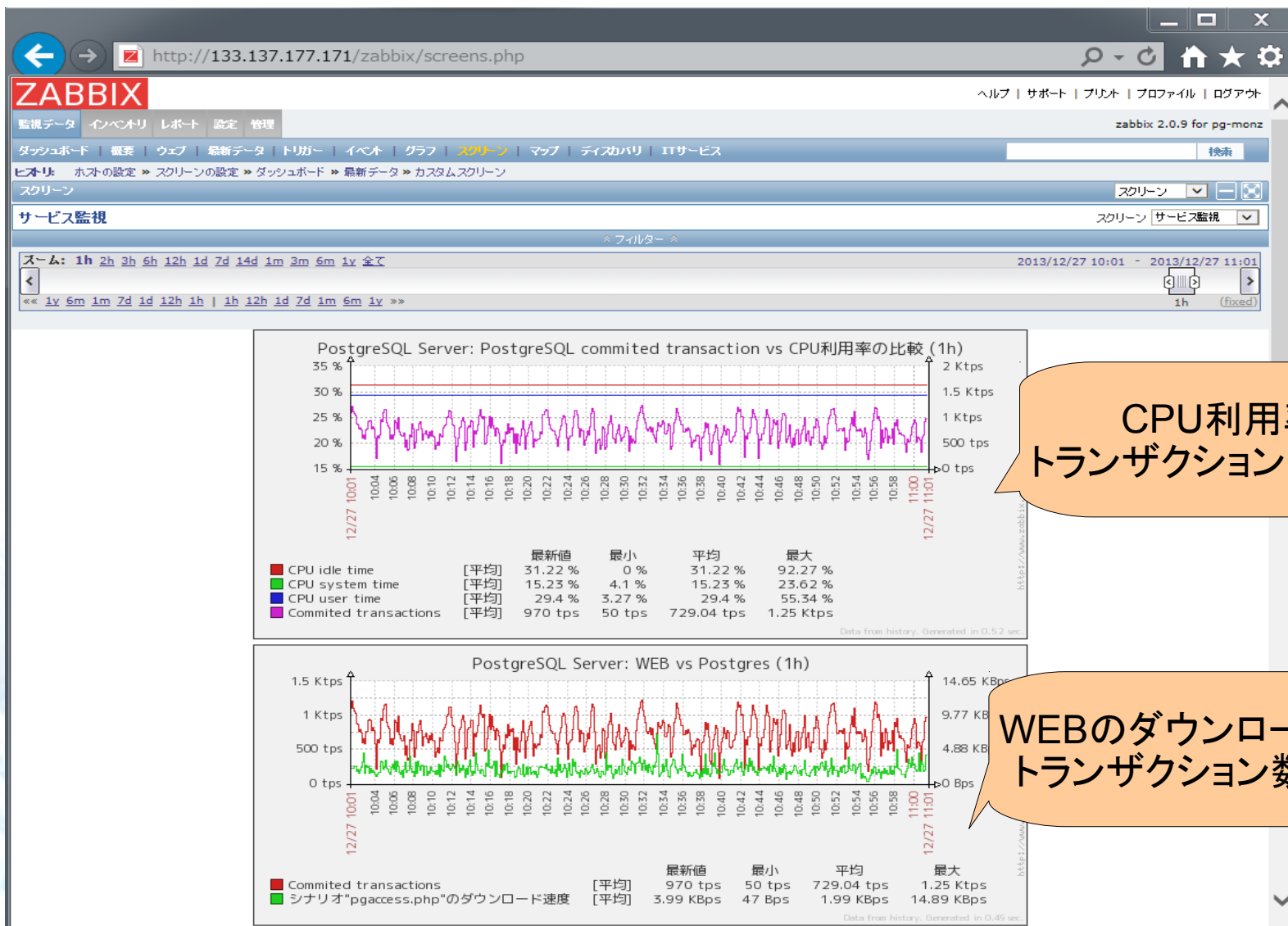
ハードディスク 600GB 10,000 回転 2.5inch SAS x4本 RAID5

pg_monz 検証結果

- PostgreSQLサーバ
 - pg_monzを利用し1データベースクラスタの中の300データベースを監視すると平均CPU利用率が17.22%上昇した
 - 1DBサーバ当たり平均CPU利用率 0.05%の負荷上昇となった
 - デフォルトの監視間隔は5分なので、仮に1分とした場合には5倍の負荷がかかる予想される。この場合1DBサーバ当たりの平均CPU利用率は0.25%上昇することになる
 - pg_monzを利用し1データベースクラスタ、1データベースに対し300テーブルを監視すると平均CPU利用率が23.09%上昇した
 - 30テーブル当たり平均CPU利用率2.3%上昇となった
 - 大量のテーブル監視をする場合には監視間隔を伸ばしたほうがよい。
- Zabbixサーバ
 - pg_monzを利用すると最大CPU利用率が1.67%上昇した

pg_monz 監視以外の応用活用

- PostgreSQLチューニング
 - 一時ファイルが多発 …work_memを増やす
 - checkpoint_segments超過によるcheckpoint多発
 - …checkpoint_segmentsを増やす
- Zabbix付属のTemplate_OS_Linuxと併用
 - OSのロードアベレージやCPU率、iowaitとDBのトランザクション量の相関関係を調べる
- WEB監視との併用
 - ZabbixのWEB監視を利用して、アプリケーションの応答時間とデータベースのCOMMIT/sの相関関係を調べる



CPU利用率と
トランザクション数の比較

WEBのダウンロード時間と
トランザクション数の比較

pg_monz 今後の改良予定

- PostgreSQL Streaming Replication監視対応
 - レプリケーション遅延状況
 - マスタ、スレーブの役割確認
- pgpool-IIの監視
 - プロセス確認
 - ノードのステータス確認
 - セッション確認
- PostgreSQLのバックグラウンドワーカプロセスの利用
 - UserParameterではなくPostgreSQLのサブプロセスにより監視を行って負荷軽減

pg_monz

- WEB

- 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