

新バージョン！ Zabbix 2.2 新機能検証結果 DB ベンチマーク結果のご紹介

SRA OSS, Inc. 日本支社
盛 宣陽

概要

- Zabbix 2.2の新機能
- 各機能の検証結果
- Zabbix DB MySQL vs PostgreSQLの比較

Zabbix 2.2のリリース！

- リリースが遅れましたが
2013年11月12 日にリリースしました



Zabbix 2.2の新機能

- VMWare監視機能
 - ハイパーバイザ・ゲストVMのリソース監視
 - vSphere (ESXi)、VMの自動検知・登録
- データベースの自動アップグレード機能
- Web監視の改良
- ローダブルモジュール形式の監視機能拡張
- パフォーマンスの改善
 - 複数timerプロセス、履歴データのメモリキャッシュ、他
- その他細かな改善点多数

今回のご報告内容

- データベース自動アップグレード機能
- Web監視の改良
- VMware監視
- パフォーマンスの改善

データベースの自動アップグレード機能

データベース自動アップグレード機能

- Zabbix 2.0までは、添付のスクリプトを使って手動でデータベースをアップグレードする必要があった
- Zabbix 2.0から2.2へのデータベースのアップグレードは自動で行ってくれる
- 2.0で使用していたデータベースを参照するように設定して2.2を起動するだけでOK
- 1.8から2.2にアップグレードするには、一度手動で1.8から2.0にアップグレードした後、2.2への自動アップグレードを行う必要がある
- 2.2以降のバージョンへの自動アップグレードもOK(予定)
(2.0 → 2.4等)

データベース自動アップグレード機能

- Zabbix サーバ起動時にデータベースのバージョンがチェックされ、アップグレードが必要であれば実行される
- 進捗はログに逐次出力される

```
30068:20131004:154611.244 Starting Zabbix Server. Zabbix 2.1.6
(revision 38841).
30068:20131004:154611.293 current database version
(mandatory/optional): 02010000/02010000
30068:20131004:154611.293 required mandatory version: 02010176
30068:20131004:154611.294 starting automatic database upgrade
30068:20131004:154611.314 completed 0% of database upgrade
30068:20131004:154611.332 completed 1% of database upgrade
30068:20131004:154611.353 completed 2% of database upgrade
... (中略) ...
30068:20131004:154612.171 completed 99% of database upgrade
30068:20131004:154612.172 completed 100% of database upgrade
30068:20131004:154612.172 database upgrade fully completed
```

データベース自動アップグレード機能

- 1.8 → 2.0のときはヒストリテーブルの変更があったため、データ量によっては数時間かかった
- 2.0 → 2.2ではヒストリテーブルは変更されないの
で、それほど時間はかからない
- 途中で中断したりした場合は戻せないの
で、事前に必ずバックアップをとる

Web監視の改良

Web監視

- WebサイトにHTTP/HTTPSクライアントとしてアクセスして監視
 - Webブラウザからのアクセスをシミュレート
- 複数ページにまたがった監視(Webシナリオ)が可能
- 死活監視、パフォーマンスの監視

Zabbix 2.2以前のWeb監視の問題点

- Web監視設定の再利用ができない
 - ホスト名が異なるだけで同じWebシナリオで監視設定をする場合でも再設定が必要
 - 設定のエクスポート&インポートができない
 - マクロが効かない

Zabbix 2.2でのWeb監視(1)

- テンプレートの対応



テンプレートの設定

テンプレート

グループ

Displaying 1 to 30 of 30 found

<input type="checkbox"/>	テンプレート ↓	アプリケーション	アイテム	トリガー	グラフ	スクリーン	ディスカバリ	ウェブ	ラ
<input type="checkbox"/>	Template_WEB	アプリケーション (1)	アイテム (0)	トリガー (0)	グラフ (0)	スクリーン (0)	ディスカバリ (0)	ウェブ (1)	-

- 設定のエクスポート&インポートもできる
しかし・・・Webシナリオのエクスポートができない

Zabbix 2.2でのWeb監視(2)

- 設定にマクロが使えるようになった
テンプレートとマクロを利用し設定の再利用ができる

名前	タイム アウト	URL	要求文 字列	ステータスコード
1: index.php	15 秒	http://{HOST.IP} /zabbix22/index.php		200
2: login	15 秒	http://{HOST.IP} /zabbix22/index.php	Zabbix SIA	200
3: dashboard.php	15 秒	http://{HOST.IP} /zabbix22/dashboard.php	y-mori	200

Zabbix 2.2でのWeb監視 その他の改良点(1)

シナリオ ステップ

親Webシナリオ [Template_WEB](#)

名前 Zabbix frontend

アプリケーション WEB

アプリケーションの作成

認証 なし

更新間隔(秒) 5

リトライ 1

エージェント Internet Explorer 10.0

HTTPプロキシ http://[username[:password]@]proxy.example.com[:port]

変数
{user}=Admin
{password}=zabbix

有効

Webシナリオ別に

リトライ回数の指定

HTTPプロキシの設定

2.0以前では
http_proxy環境変数でグローバル
に設定された

Zabbix 2.2でのWeb監視 その他の改良点(2)

- グローバルサーチによる検索結果で
Web監視が追加
(2.0以前では数が多いと探すのが大変)

グローバルサーチ
ホスト名入力

The screenshot shows the Zabbix 2.2 web interface. At the top, there is a navigation menu with items like 'ダッシュボード', '概要', 'ウェブ', '最新データ', etc. Below the menu, there is a search bar containing 'zabbix server' and a '検索' button. The search results are displayed in a table with columns: 'ホスト', 'IPアドレス', 'DNS', '最新データ', 'トリガー', 'イベント', 'スクリーン', 'ウェブ', 'アプリケーション', 'アイテム', 'トリガー', 'グラフ', 'ウェブ'. The 'ウェブ' column for the 'Zabbix server' host shows 'ウェブ (1)'. There are green boxes highlighting the search bar and the 'ウェブ' column header and value. Arrows point from the text '取得値' to the 'ウェブ' column header and from '設定値' to the 'ウェブ (1)' value.

ホスト	IPアドレス	DNS	最新データ	トリガー	イベント	スクリーン	ウェブ	アプリケーション	アイテム	トリガー	グラフ	ウェブ
Zabbix server	127.0.0.1		最新データ	トリガー	イベント	スクリーン	ウェブ	アプリケーション (12)	アイテム (70)	トリガー (42)	グラフ (12)	ウェブ (1)

取得値

設定値

Zabbix 2.2でのWeb監視 その他の改良点(3)

- ログの改良

サーバ名、シナリオ名、ステップ名の表示

cannot process step "**index.php**" of

web scenario "**Zabbix frontend**" on host

"**Zabbix server**": Couldn't connect to server

Zabbix 2.2でのWeb監視 その他の改良点(4)

- 変数の改良
 - 2.0ではシナリオ全体に変数が指定できたが、2.2からはシナリオの各ステップで変数指定ができるようになった
 - 変数でregex:で始まるものは正規表現で値をマッチさせてから変数に格納できる
`{session}=regex:name="sid" value="([0-9a-z])"`
この結果をPOSTすることで、動的なWebシナリオ確認ができる

VMware監視

VMware管理機能

- ハイパーバイザ、ゲストVMのリソース監視
- ディスバリ機能によりvSphere、VMの自動検知・登録
- 監視項目
 - CPU に関する情報 (VM/HV)
 - メモリに関する情報 (VM/HV)
 - 電源 ON / OFF 状態 (VM)
 - ストレージに関する情報 (VM/HV)
 - 稼働時間 (VM/HV)
 - ハイパーバイザに関する情報 (HV)
 - イベントログ (HV)

登録も簡単

- 事前準備
 - Zabbix を --with-libcurl --with-libxml2 オプションでコンパイルしておく (ソースからのコンパイルの場合のみ)
 - zabbix_server.conf 設定ファイルの設定
StartVMwareCollectors を 1 以上に設定
- 設定
 1. VMware vSphere もしくは vCenter のホストを作成する
 2. {\$URL}, {\$USERNAME}, {\$PASSWORD} マクロを適切に設定する
 - 注 {\$URL} は https://(vSphere or vCenterのホスト名/sdk
 3. ホストにテンプレート「Template Virt VMware」を適用する
 4. ディスカバリによりハイパーバイザ、ゲスト VM が自動登録され、監視が開始される

監視開始

- ディスカバリによりホストが登録
- 監視後に新規作成ゲストも自動登録



Name	Applications	Items	Triggers	Graphs	Discovery	Web	Interface	Templates	Status	Availability
Discover VMware VMs: CentOS 6 x64	Applications (8)	Items (29)	Triggers (0)	Graphs (0)	Discovery (3)	Web (0)	10050	Template_Virt_VMware_Guest	Monitored	
Discover VMware VMs: CentOS 6 x64	Applications (8)	Items (21)	Triggers (0)	Graphs (0)	Discovery (3)	Web (0)	10050	Template_Virt_VMware_Guest	Monitored	
Discover VMware VMs: CentOS 6 x64 Hinemos 4.1	Applications (8)	Items (29)	Triggers (0)	Graphs (0)	Discovery (3)	Web (0)	10050	Template_Virt_VMware_Guest	Monitored	
Discover VMware VMs: CentOS 6 x64 Hinemos 4.1 agent	Applications (8)	Items (29)	Triggers (0)	Graphs (0)	Discovery (3)	Web (0)	10050	Template_Virt_VMware_Guest	Monitored	
Discover VMware VMs: dzbx000 CentOS6 x64 Zabbix 2.0.9 Agent	Applications (8)	Items (29)	Triggers (0)	Graphs (0)	Discovery (3)	Web (0)	10050	Template_Virt_VMware_Guest	Monitored	
Discover VMware VMs: dzbx000 CentOS6 x64 Zabbix 2.0.9 Server	Applications (8)	Items (29)	Triggers (0)	Graphs (0)	Discovery (3)	Web (0)	10050	Template_Virt_VMware_Guest	Monitored	
Discover VMware VMs: dzbx000 VMware vCenter Server Appliance	Applications (8)	Items (21)	Triggers (0)	Graphs (0)	Discovery (3)	Web (0)	10050	Template_Virt_VMware_Guest	Monitored	
Discover VMware VMs: New Test VM	Applications (8)	Items (17)	Triggers (0)	Graphs (0)	Discovery (3)	Web (0)	10050	Template_Virt_VMware_Guest	Monitored	
Discover VMware hypervisors: osspc32.sra.co.jp	Applications (6)	Items (21)	Triggers (0)	Graphs (0)	Discovery (1)	Web (0)	10050	Template_Virt_VMware_Hypervisor	Monitored	
Discover VMware VMs: ossspt CentOS6 x64 Tomcat	Applications (8)	Items (29)	Triggers (0)	Graphs (0)	Discovery (3)	Web (0)	10050	Template_Virt_VMware_Guest	Monitored	
vSphere	Applications (2)	Items (1)	Triggers (0)	Graphs (0)	Discovery (3)	Web (0)	10050	Template_Virt_VMware	Monitored	

パフォーマンスの改善

パフォーマンスの改善

Zabbixのキャッシュの改善の歴史

- **1.8:** 収集したデータをDBに書き込むときにまとめて書き込むようになった(書き込みキャッシュ)
- **2.0:** 設定(アイテム、トリガー等)のキャッシュが追加された
- **2.2:** ヒストリデータ(監視データの履歴)の読み込みキャッシュ(value cache)が追加された
 - トリガー、計算アイテム、累計アイテムの処理が高速化

2.0と2.2の性能比較

- Zabbix 2.0.8 vs 2.1.6
- 100ホスト (同一エージェント)
- $100 \times 44 = 4400$ アイテム
- $100 \times 17 = 1700$ トリガー
- 5秒間隔で取得
- historyが空の状態から1時間監視
- ValueCacheSize=16M (Zabbix 2.1.6)
 - デフォルトの8Mだと不足するため

Test 1: 通常の監視

- トリガーは最新の値、もしくは最新と1つ前の値との差を取得

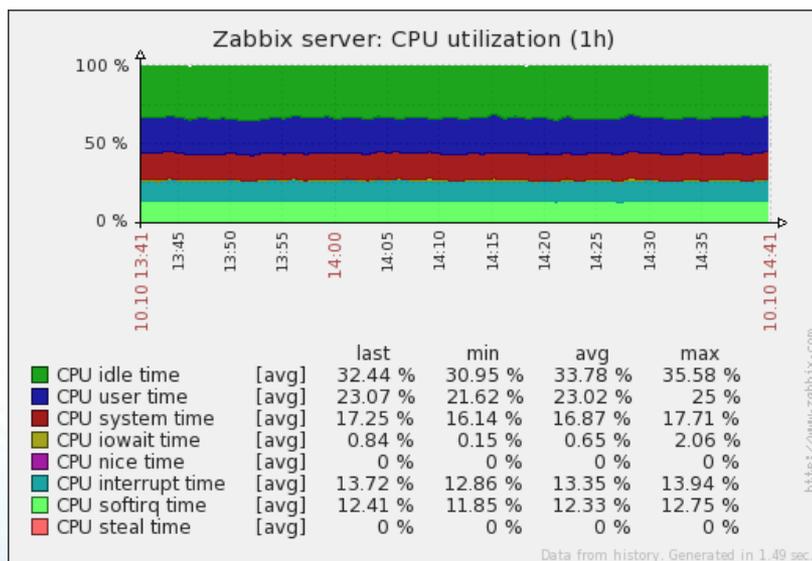
(例)

```
{system.cpu.util[,iowait].last(0)}>20
```

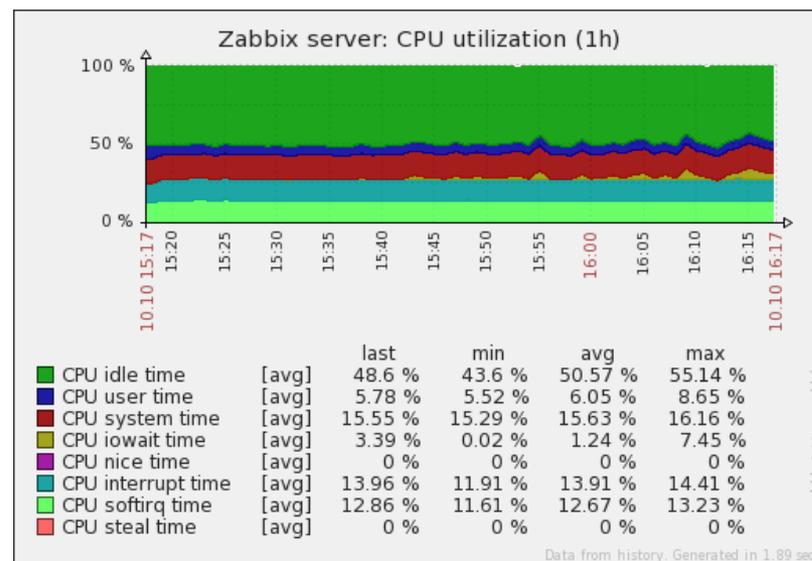
```
{vfs.fs.size[/,pfree].last(0)}<10
```

```
{system.uptime.change(0)}<0
```

Test 1: CPU utilization

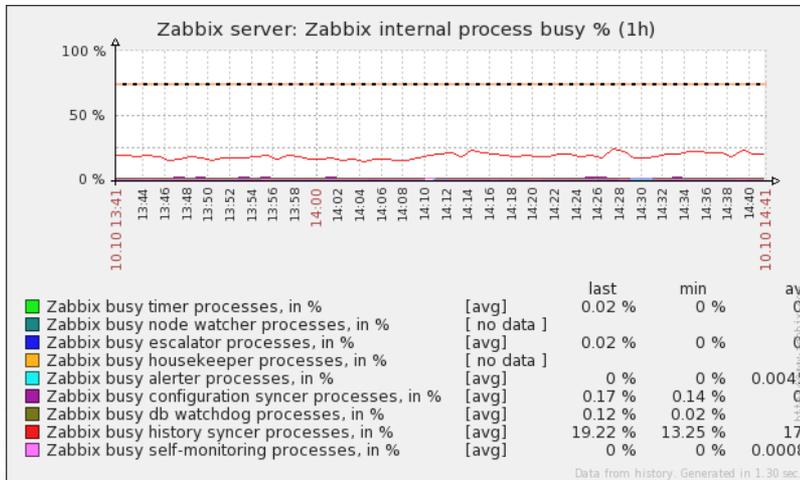


2.0.8: CPU user timeの負荷がやや高い

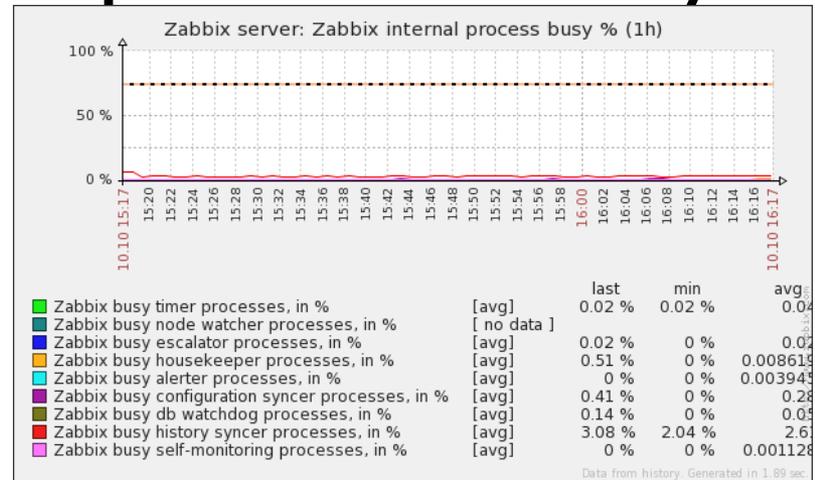


2.1.6: 負荷はかなり低い

Test 1: Zabbix internal process busy



2.0.6: history syncer processes
の負荷がやや高い



2.1.6: 全体的に負荷は低い

Test 2: 広範囲なヒストリを参照する場合

- トリガーで過去30分間の値の最大・最小・平均値を取得

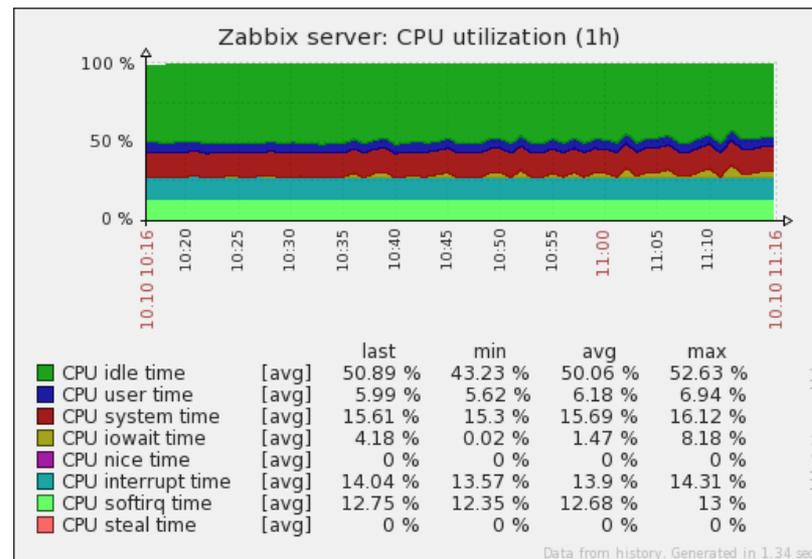
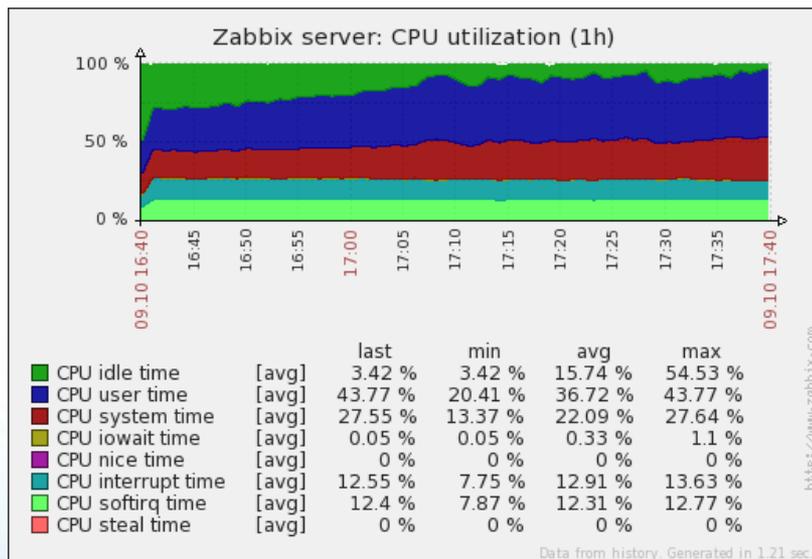
(例)

```
{system.cpu.util[,iowait].avg(1800)}>20
```

```
{vfs.fs.size[/,pfree].min(1800)}<10
```

```
{system.swap.size[,pfree].max(1800)}<50
```

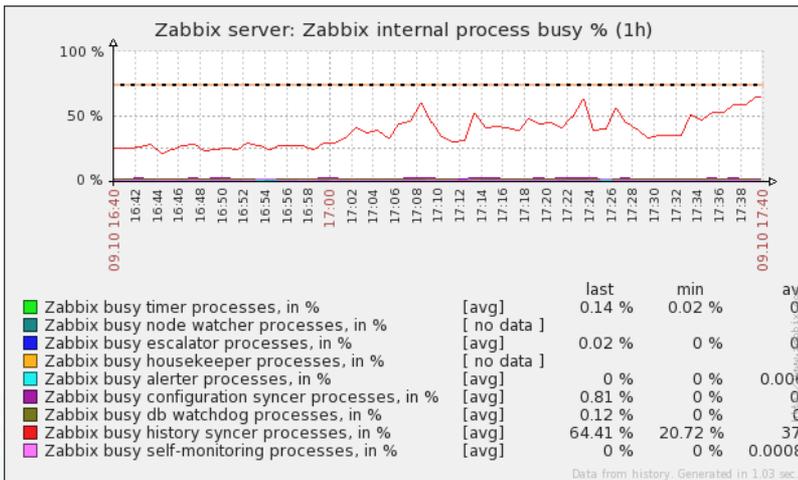
Test 2: CPU utilization



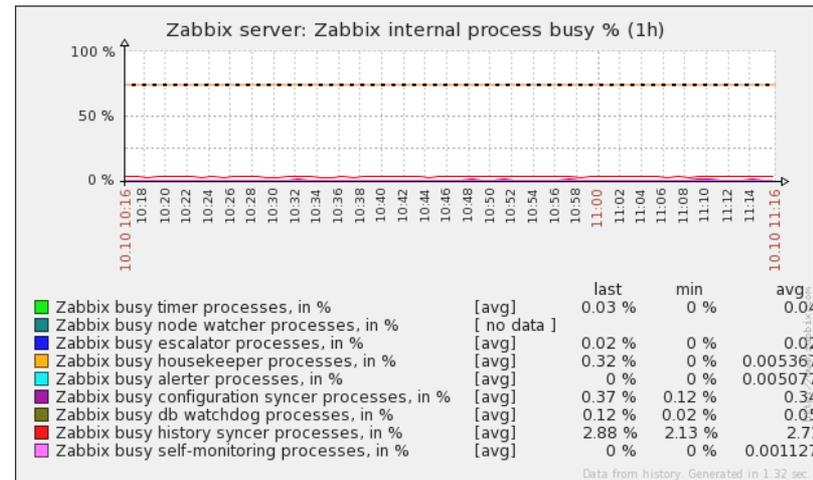
2.0.8: CPU user timeの負荷が
かなり高い
(トリガーの履歴参照負荷が
上乘せされた)

2.1.6: 負荷はかなり低い
(Test 1の場合とほぼ同じ)

Test 2: Zabbix internal process busy



2.0.6: history syncer processesの負荷がかなり高い



2.1.6: 全体的に負荷は低い

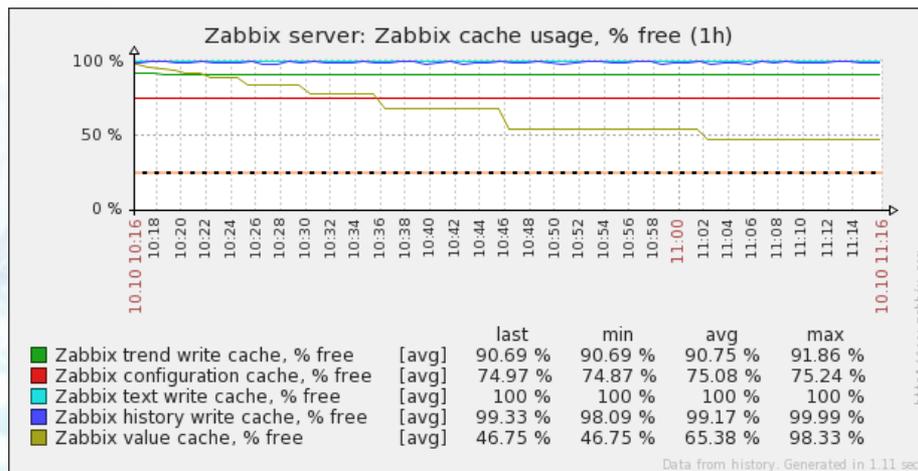
パフォーマンスの改善

- 2.0に比べ、2.2は監視の負荷が激減
 - DBへの負荷が軽減
 - 同一性能のマシンであれば2.2の方がより多くの監視が可能
- Value cacheの効果により、2.2ではトリガーで過去のヒストリを広範囲に参照しても負荷が軽い
- ちなみにvalue cacheを無効にすると2.0より負荷が高くなってしまいますのでおすすめしない

Value cacheのチューニング

- ValueCacheSizeが足りない場合、以下のログが出力される

26791:20131009:144349.698 value cache is fully used:
please increase ValueCacheSize configuration parameter



value cacheの情報は
以下のアイテムで取得できる

zabbix[vcache,buffer,pfree]
zabbix[vcache,cache,hits]
zabbix[vcache,cache,misses]

(Template App Zabbix Server
にデフォルトで設定されている)

Zabbix DB MySQL vs PostgreSQLの比較

検証環境

- ソフトウェア

Zabbix 2.1.1 (アルファリリース版)

DB MySQL 5.6.12/PostgreSQL 9.2.4

- 環境

Amazon EC2 M1 medium instance

- 3.75 Gib RAM
- 2ECU (1 core)
- 410 GB instance storage
- OS: Amazon Linux AMI 2013.03 (64bit)
- 3 servers

(1 for Zabbix server and DB ,2 for Zabbix Agent)

テストターゲット

- 600 hostsをシミュレート (2 agent servers)
- 26400 items
(44 items/1 host, 5秒間隔, 保存期間1日)
- 10200 triggers (17 triggers / 1 host)
- 空のhistoryデータからスタート
- 測定時間:
 - 4 時間 (short run)
 - 36 時間 (long run)
- Zabbix:
変更点 `CacheSize=64M` in `zabbix_server.conf`

テスト内容(short run)

- Test 1
基本的にデフォルトDB設定
- Test 2
バッファとトランザクションログの量を増やす
- 測定内容
4時間測定
CPU利用率とZabbixパフォーマンスを測定

Test 1: デフォルトDB設定

- MySQL

`character-set-server=utf8`

`skip-character-set-client-handshake`

`innodb_file_per_table`

- PostgreSQL

全てデフォルト

PostgreSQLは1つのテーブルに複数のファイルを利用

Test 1 CPU利用率

MySQL

PostgreSQL



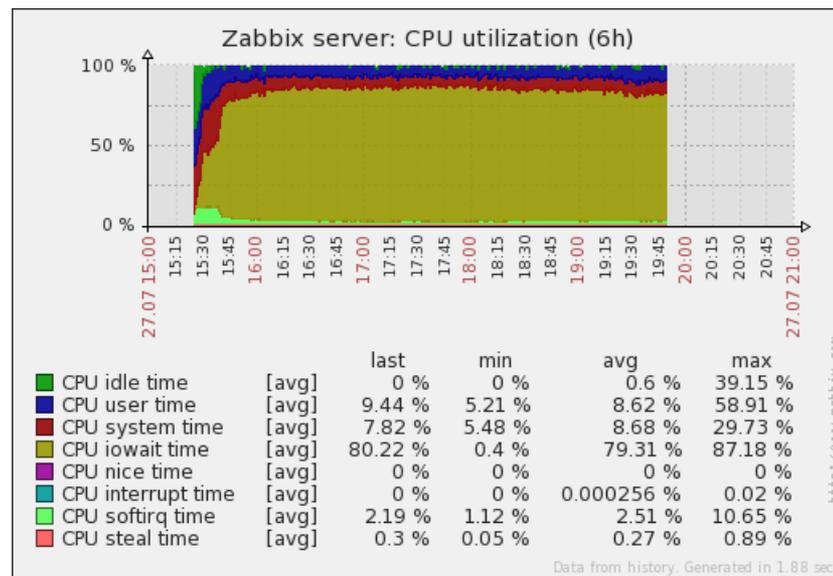
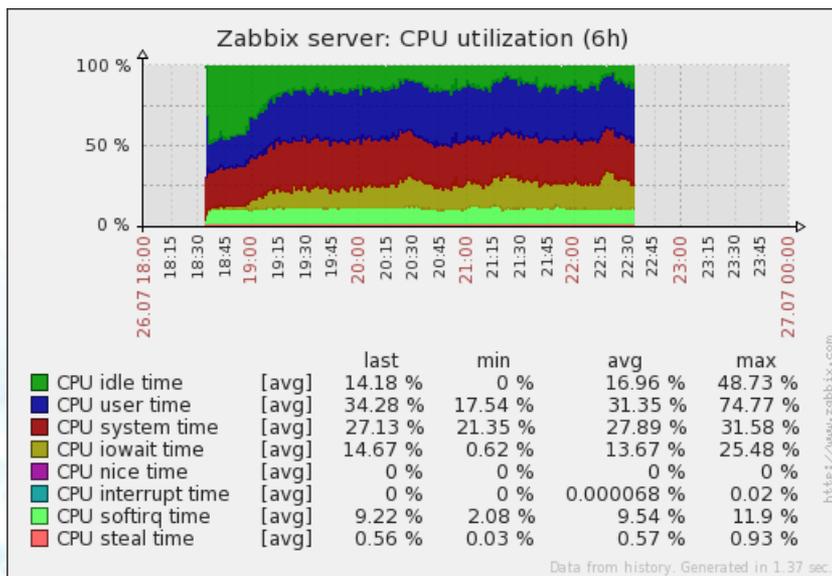
User



System



iowait



PG: 大量のio wait

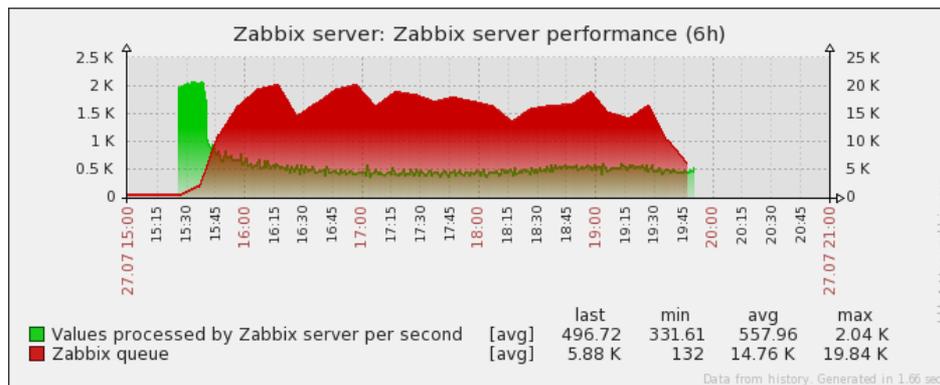
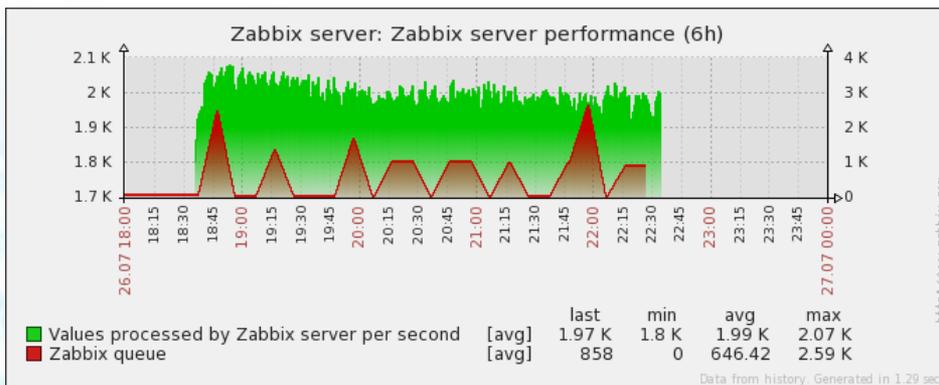
Test 1: Zabbixパフォーマンス

MySQL

PostgreSQL

 zabbix[wcache,values] :Values processed by Zabbix server per second

 zabbix[queue]:Zabbix queue



PG: Zabbix処理が遅延

Test 2: Tuning

- バッファサイズとトランザクションログサイズの変更
- MySQL
 - `Innodb_befffer_pool_size = 512MB`
 - `Innodb_log_file_size = 256MB`
x `innodb_log_files_in_group=2`(default)
= log size 512MB
- PostgreSQL
 - `Shared_buffers = 512MB`
 - `checkpoint_segments = 32`
16MB/each segmnet → log size 512MB

Test 2: CPU利用率

MySQL

PostgreSQL



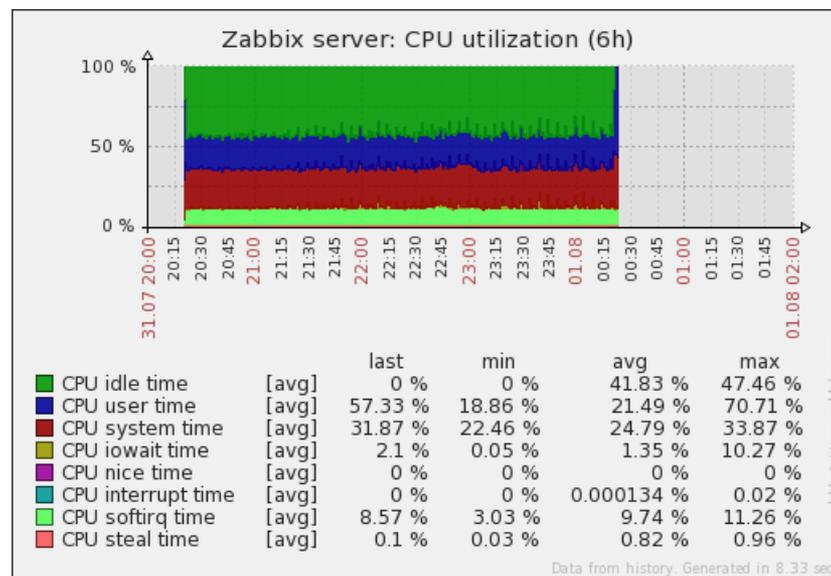
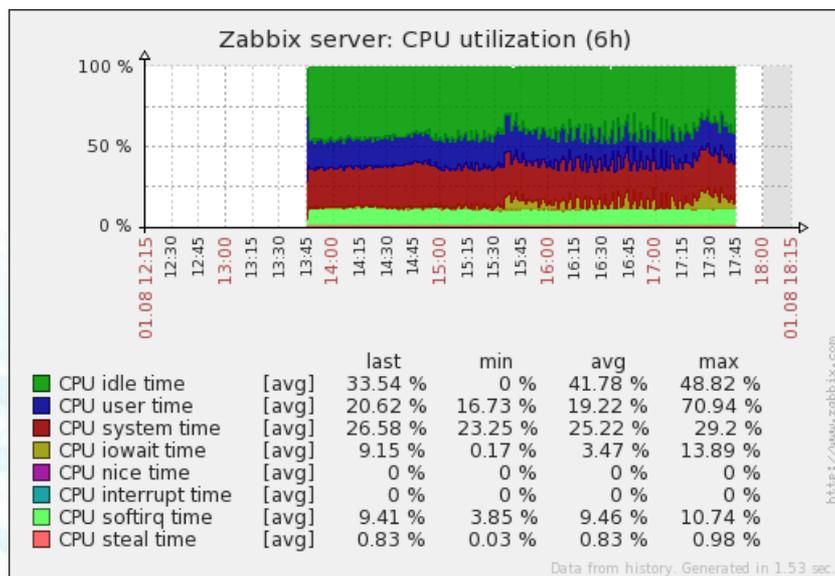
User



System



iowait



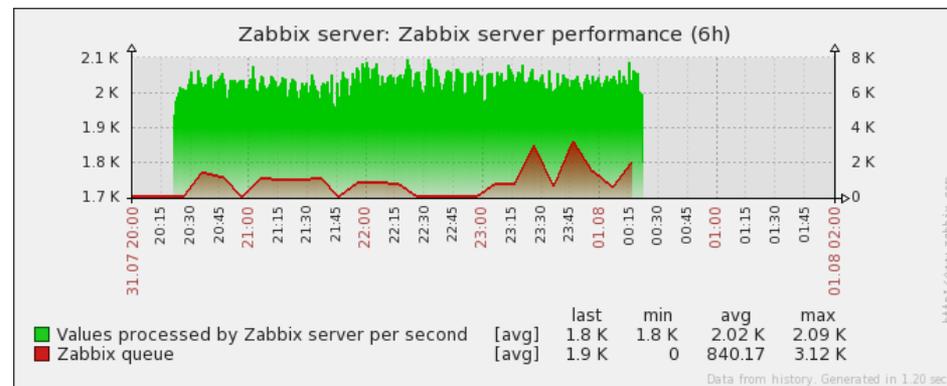
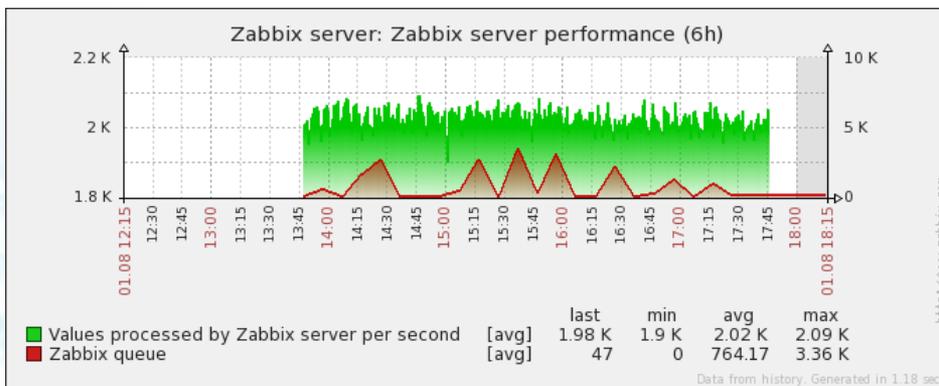
ほとんど同じ負荷状況 (PG: 若干ioが低く安定)

Test 2: Zabbixパフォーマンス

MySQL PostgreSQL

 zabbix[wcache,values] :Values processed by Zabbix server per second

 zabbix[queue]:Zabbix queue



ほとんど同じ性能

DBサイズ比較

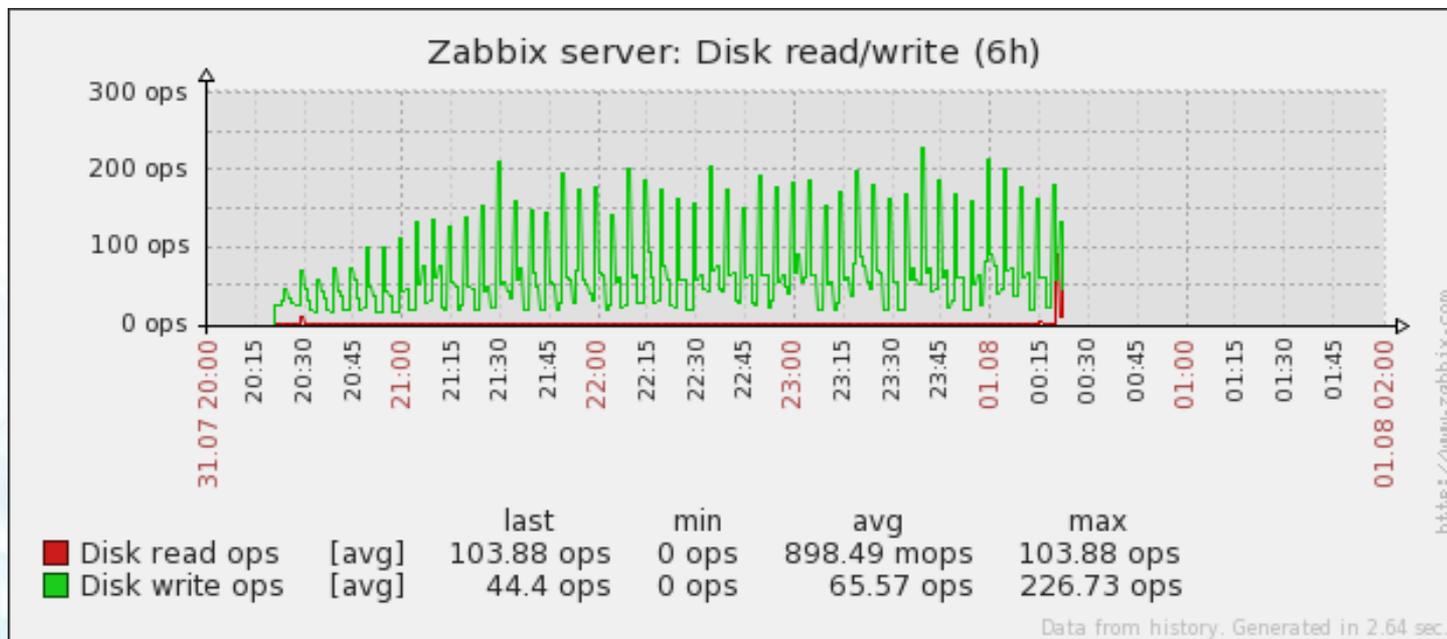
- Test 2
 - トランザクションログサイズは除外
- MySQL : 3.3 GB
- PostgreSQL : 3.2 GB
- ほとんど同じ

ここまでのまとめ

- デフォルトの設定では、MySQLの方が良好
- バッファサイズとトランザクションログサイズを増やすことで、MySQLとPostgreSQLはほぼ同じ性能が出せる
- DBサイズ増加傾向もほとんど同じ

Zabbixの特性

- 書き込みが多く読み込みが少ない特性
(2.2以降ではValue Cacheが効く)



チューニングのポイント

- 書き込みが多いケースでは
Checkpointのチューニングが重要
- 誤った設定は書き込み負荷が増大
- チェックポイントの契機
 - バッファあふれ
 - トランザクションログあふれ
- チェックポイントのチューニングは、バッファとトランザクションログのチューニングが基本中の基本

Zabbix DBのチューニング

- Zabbix DBは基本的に同じ頻度で書き込みを行う
- DBサイズの成長速度を測る
今回のテストでは
3.2GB/4h \doteq 70MB/5min
- DBサイズの成長速度から、十分なバッファ量とトランザクションログ量を設定する

今回のテストでは、下記の設定にて劇的な改善
バッファ512MB トランザクションログ量512MB

まとめ

- デフォルトの設定ではMySQLが速い
 - PostgreSQLのデフォルト設定が悪すぎ
- チューニングすればMySQLとPostgreSQLの性能はほとんど同じ

その他のベンチマーク結果

- 長時間テスト

housekeeper起動するとMySQLはZabbixのパフォーマンスが落ちやすい傾向

- パーティショニングテスト

PostgreSQLはCPU利用率が高い

- テストの詳細

http://www.sraoss.co.jp/event_seminar/2013/20130829_Zabbix_PG_vs_MySQL.pdf

参考URL

- Zabbix
 - <http://www.zabbix.com/jp/>
- ZABBIX-JP
 - <http://www.zabbix.jp/>
- SRA OSS, Inc. 日本支社
 - <http://www.sraoss.co.jp>
- 弊社で公開中のZabbix技術情報
 - <http://www.sraoss.co.jp/technology/zabbix/>