

Zabbix 2.2 検証レポート

2013年11月1日
SRA OSS, Inc. 日本支社

本レポートでは、近日リリース予定の Zabbix 2.2 の新機能を検証し、評価した結果を公開します。

はじめに
検証環境
使用したソフトウェア
Zabbix 2.2 の新機能
各機能の検証結果
まとめ

はじめに

Zabbix は、オープンソースの企業向けシステム監視ツールです。Web ベースの使いやすい管理インタフェース、大規模な監視への対応、高機能な通知、可視化機能などが特長です。

本レポートでは、近日リリース予定である Zabbix 2.2 の α 版である 2.1.6、 β 版の 2.1.7(2013年11月1日現在、 β 版の 2.1.8 がリリース済み)を使用し、実際にその新機能を検証し、評価した結果を記載しています。

検証環境

ここでは以下の仮想マシン環境に Zabbix を導入し、検証を行いました。

役割	Zabbix サーバ	Zabbix エージェント
OS	CentOS 6.3 (x86_64)	CentOS 6.3 (x86_64)
ホスト名 (IP アドレス)	zabbix-server (192.168.5.1)	zabbix-agent (192.168.5.2)
CPU	2.4GHz x1 コア (仮想 CPU)	2.4GHz x1 コア (仮想 CPU)
メモリ	742MB	742MB
HDD	6.9GB	6.9GB

使用したソフトウェア

以下のソフトウェアをソースコードからコンパイルし、使用しました。
その他のソフトウェアは CentOS 6 標準のものを利用しました。

- Zabbix 2.1.6/2.1.7
- PostgreSQL 9.2.4

Zabbix 2.2 の新機能

Zabbix 2.2 の主な新機能は以下になります。

- VMware 監視機能
- 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
- ヒストリデータの Cassandra サポート (2.2.x で実装予定)
- その他細かな改善点

各機能の検証結果

今回は以下の 5 項目について検証を行いました。

- VMware 監視機能
- 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

VMware 監視機能

Zabbix 2.2 では新たに VMware 監視機能が追加されました。
本機能では以下の監視が行えます。

- ハイパーバイザ(vSphere / ESXi)、ゲスト VM のリソース監視
- ディスカバリ機能による vSphere (ESXi)、VM の自動検知・登録

VMware 監視では以下のような項目が監視できます。

- 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)

本検証では以下のソフトウェアを使用し、VMware 監視の検証を行いました。

- Zabbix 2.1.7
- VMware ESXi 5.5
- VMware vCenter Server Appliance 5.5

VMware 監視の設定

VMware 監視を有効にするには、Zabbix を `--with-libcurl --with-libxml2` オプション付きでコンパイルする必要があります。また、`zabbix_server.conf` 設定ファイルの `StartVMwareCollectors` を 1 以上に設定する必要があります。

VMware 監視は以下の手順で行います。

- VMware ハイパーバイザ(vSphere/ESXi)もしくは vCenter のホストを作成する
- `{$URL}`, `{$USERNAME}`, `{$PASSWORD}` マクロを適切に設定する
 - `{$URL}` は `https://(vSphere or vCenter のホスト名/IP アドレス)/sdk` を指定
- ホストにテンプレート「Template Virt VMware」を適用する

- 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

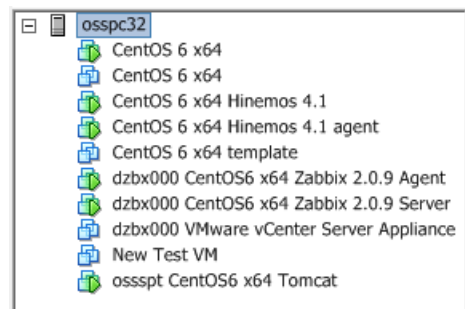
VMware 監視の動作

今回は以下の動作を確認しました。なお、ディスクバリの実行間隔はデフォルトでは 1 時間となっていますが、今回は検証のため 1 分間隔に変更しています。

- vSphere (ESXi)に Template Virt VMware テンプレートを適用し、ハイパーバイザと作成済みのゲスト VM が自動登録される
- vCenter に Template Virt VMware テンプレートを適用し、ハイパーバイザと作成済みのゲスト VM が自動登録される
- 各ハイパーバイザ、VM の監視データが取得できる
- 新たにゲスト VM を作成し、それが監視対象として自動で追加される
- ゲスト VM を削除すると、削除されたことが検出される(一定期間後に自動削除される)

VMware ESXi 上に右のような VM が存在する場合、Zabbix では以下のように監視対象が追加されます。

「Discover VMware hypervisors:」が自動検出された vSphere(ESXi) に対応するホスト、「Discover VMware VMs:」が自動検出されたゲスト VM になります。



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	🟢🟢🟢🟢

VMware 監視テンプレートの仕様について

VMware 監視テンプレートは以下のような仕様、制限があることに注意する必要があります。

- Template Virt VMware: vSphere/ESXi もしくは vCenter ホストに適用するテンプレート
- Template Virt VMware Guest: ディスカバリされたゲスト VM に自動適用されるテンプレート
- Template Virt VMware Hypervisor: ディスカバリされたハイパーバイザに自動適用されるテンプレート
- Template Virt VMware を vSphere/ESXi に適用した場合、ハイパーバイザのホストが手動作成したものと自動登録されたものが 2 重に登録される
- ディスカバリにより自動登録されたゲスト VM ホストのインターフェースは上位のホストのものが引き継がれる (VM の IP アドレス等は取得できない)
 - そもそもホストプロトタイプでインターフェースに関する設定ができない
- VMware 監視では VMware からみた情報のみ取得できる
- 各 VM 上で Zabbix エージェントを動作させており、各 VM のエージェント監視を自動的に開始させたい場合、VMware 監視の枠組内ではできず、別途ホストディスクバリで行う必要がある
 - この場合各ホストが 2 重登録になる
- Zabbix 2.1.6 から 2.1.7 で、「Template Virt VMware vSphere」という vSphere 単体に適用するテンプレートが削除されるなど、大きく仕様が変更されており、2.2 正式リリースまでにさらに仕様の変更される可能

性がある

データベースの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りましたが、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うことができます。

アップグレード方法としては、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 を起動するだけです。

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ります。

2.0 から 2.2 以降のバージョンへの自動アップグレードも将来的に可能となる予定です(2.0 → 2.4 等)。

アップグレードの実行

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ます。進捗は以下のようにログに逐次出力されます。

```
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
```

1.8 → 2.0 のときは履歴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程度の時間がかかることがありましたが、2.0 → 2.2 では履歴テーブルは変更されないため、それほど時間はかかりません。なお、途中で中断したりした場合は簡単には元の状態には戻せないため、事前に必ずバックアップを取っておく必要があります。

Web 監視の改良

Zabbix の Web 監視は、Web サイトに HTTP/HTTPS クライアントとしてアクセスして監視する機能です。Web ブラウザからのアクセスをシミュレートします。

複数ページにまたがって画面遷移を行う監視(Web シナリオ)が可能です。

Web サイトの死活監視、パフォーマンスの監視が行えます。

Zabbix 2.2 以前の Web 監視の問題点

以前のバージョンでは、Web 監視設定の再利用ができないという問題がありました。具体的には以下のような問題です。

- テンプレートに Web 監視を設定できないため、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

テンプレートの対応

テンプレートの設定		テンプレートの作成		インポート					
テンプレート		グループ		すべて					
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	テンプレート
<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)	-

Zabbix 2.2 ではテンプレートに Web 監視を設定できるようになりました。
ただし、検証に使用したバージョン 2.1.6 では Web 監視設定のエクスポートができないという問題があります。

マクロ対応

設定にマクロが使えるようになりました。これにより、テンプレートとマクロを利用し、設定の再利用が可能です。

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

その他改良点

Web シナリオ別にリトライ回数の指定が可能になりました。
Web シナリオ毎に HTTP プロキシの設定が行えるようになりました(2.0 以前では http_proxy 環境変数でグローバルに設定可能)。

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	<input checked="" type="checkbox"/>

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

ダッシュボード | 概要 | ウェブ | 最新データ | トリガー | イベント | グラフ | スクリーン | マップ | ディスカバリ | ITサービス

zabbix server 検索

ホスト: ホストグループの設定 >> ホストの設定 >> ホストグループの設定 >> ホストの設定 >> ウェブ監視の設定

検索: zabbix server

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)

1のうち1件を表示しています

ログの改良

ログにサーバ名、シナリオ名、ステップ名が表示されるようになりました。

```
cannot process step "index.php" of
web scenario "Zabbix frontend" on host
"Zabbix server": Couldn't connect to server
```

変数の改良

2.0 ではシナリオ全体に対して変数が指定できましたが、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りました。

変数で regex: で始まるものは、Web コンテンツに対して正規表現で値をマッチさせてから、部分一致した結果を変数に格納できるようになりました。

```
{session}=regex:name="sid" value="([0-9a-z])"
```

この結果を POST することで、動的な Web シナリオ確認が行えます。

ローダブルモジュール形式の監視機能拡張

2.2 から、ローダブルモジュールにより Zabbix の機能を拡張できるようになりました。

従来の外部コマンドを使用する UserParameter による拡張に比べ、性能面で有利です。

Zabbix エージェントで起動時にローダブルモジュールを読み込むには、zabbix_agentd.conf 内で LoadModulePath で指定したディレクトリから、LoadModule=xxxx.so のように共有ライブラリファイルを指定します。

ローダブルモジュールにより追加されたアイテムなどは、通常の Zabbix の機能と同様に扱うことができます。

ソースコード内の以下の場所にサンプルがあります。このファイルを元にして、モジュールを作成します。

```
zabbix-x.y.z/src/modules/dummy/dummy.c
```

以下のように C 言語で処理の内容を記述できます。この例では、keys[] 配列にアイテムキー名と呼び出す関数を指定し、その関数の処理を記述しています。

```
int zbx_module_pgsql_version(AGENT_REQUEST *request, AGENT_RESULT *result);

static ZBX_METRIC keys[] =
/* KEY FLAG FUNCTION TEST PARAMETERS */
{
    {"pgsql.version", CF_HAVEPARAMS, zbx_module_pgsql_version, NULL},
    {NULL}
};

int zbx_module_pgsql_version(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    ....
    SET_STR_RESULT(result, version);
    return SYSINFO_RET_OK;
}
```

今回は、以下のようなモジュールを試験的に作成し、動作を確認しました。以下の画面では、Last value に SQL の実行結果(PostgreSQL のバージョン情報)が取得できていることがわかります。

- アイテムキー: postgresql.version
- PostgreSQL に接続し、以下の SQL を発行し、結果を文字列として返す

```
select version();
```
- 接続は維持し、次回の実行時に再利用する

Name	Last check	Last value	Change
- other - (1 Item)			
PostgreSQL version (by loadable ...)	10 Oct 2013 18:23:16	PostgreSQL 9.2.4 on x86_64...	- History

ローダブルモジュールを利用すると拡張の自由度が高まりますが、モジュールの作成にはある程度の Zabbix と C 言語プログラミングの知識が必要となります。

今後、汎用的なモジュールがコミュニティからいろいろと出てくることが期待でき、一般のユーザにとってもメリットがあると考えられます。

また、専門的、特殊なものは有償で Zabbix ベンダーが作りこんで提供するという形態も考えられます。

パフォーマンスの改善

Zabbix のキャッシュの改善の歴史

Zabbix はバージョンが上がるにつれ、以下のようにキャッシュによる性能の改善が図られてきました。

- 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

2.2 では、特にヒストリデータの読み込みキャッシュが追加されたことが特徴です。その他にも、DB の更新量の削減などの性能改善が行われています。本レポートでは、これによりどの程度性能が向上したかを検証しました。

2.0 と 2.2 の性能比較

以下の条件で Zabbix 2.0.8 と 2.1.6 との比較を行いました。

- 100 ホスト (同一エージェント)
- 100 x 44 = 4400 アイテム
- 100 x 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、この設定のみ変更しています。

テスト 1: 通常の監視

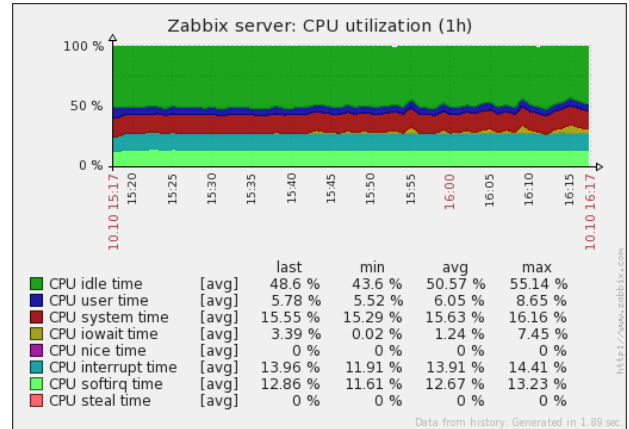
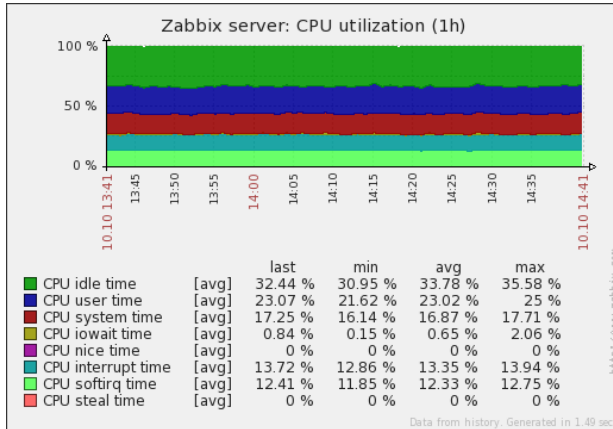
Zabbix にデフォルトで含まれている Template OS Linux テンプレートをほぼそのまま使用した監視です。

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します。

(例)

```
{system.cpu.util[,iowait].last(0)}>20
{vfs.fs.size[/,pfree].last(0)}<10
{system.uptime.change(0)}<0
```

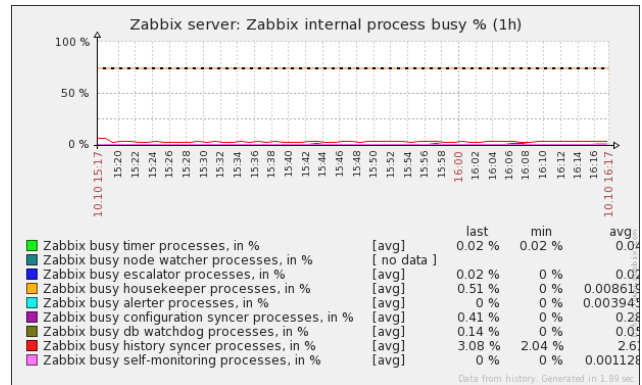
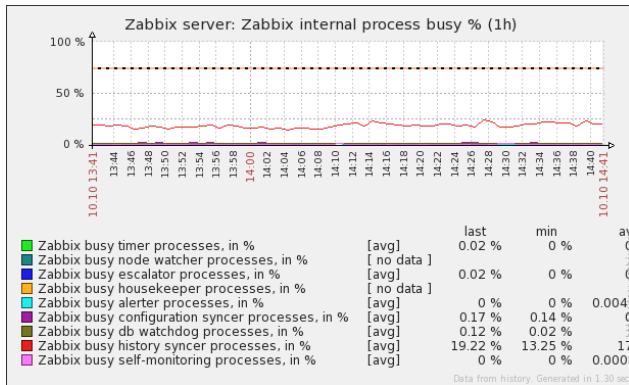
テスト 1: CPU utilization



(左) 2.0.8: CPU user time の負荷がやや高い

(右) 2.1.6: 負荷はかなり低い

テスト 1: Zabbix internal process busy



(左) 2.0.8: history syncer process の負荷がやや高い

(右) 2.1.6: 全体的に負荷は低い

通常の監視項目による監視でも、負荷が低くなっていることがわかります。

テスト 2: 広範囲なヒストリを参照する場合

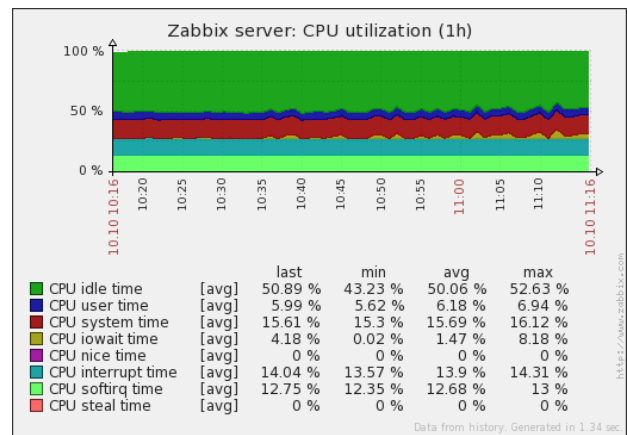
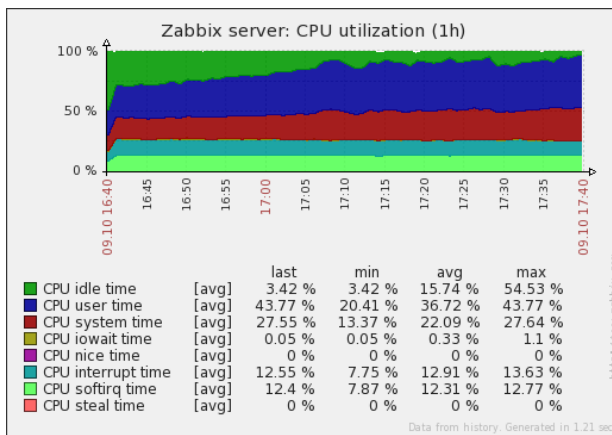
テスト 1 で使用したテンプレートのトリガーを修正し、過去のヒストリデータを広範囲に参照するようにした監視です。

トリガーで過去 30 分間の値の最大・最小・平均値を取得します。

(例)

```
{system.cpu.util[,iowait].avg(1800)}>20
{vfs.fs.size[/,pfree].min(1800)}<10
{system.swap.size[,pfree].max(1800)}<50
```

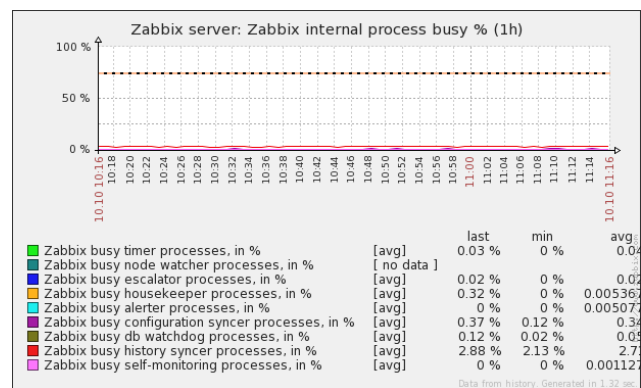
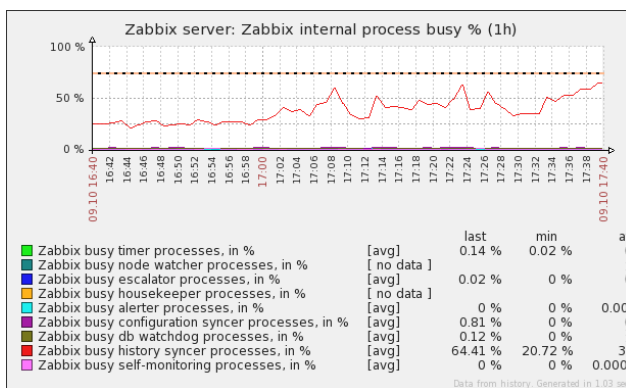

テスト 2: CPU utilization



(左) 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)

(右) 2.1.6: 負荷はかなり低い (テスト 1 の場合とほぼ同じ)

テスト 2: Zabbix internal process busy



(左) 2.0.8: history syncer process の負荷がかなり高い

(右) 2.1.6: 全体的に負荷は低い

2.0 では履歴データの参照範囲を広げると大幅に負荷が増えるのに対して、2.1 ではほとんど負荷が変わらないことがわかります。

性能比較結果について

2.0 に比べ、2.2 は監視時の負荷がかなり軽減されていることがわかりました。特に DB への負荷が軽減されています。

また、Value cache の効果により、2.2 ではトリガーで過去の履歴を広範囲に参照しても負荷がほとんど上がりません。2.0 では履歴の参照量がそのまま負荷として上乘せされます。

なお、2.2 で value cache を無効にした場合、2.0 より負荷が高くなってしまったため、無効にするのはおすすめしません。

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ます。このようなログが出た場合、ValueCacheSize の値を増やすことを検討してください。

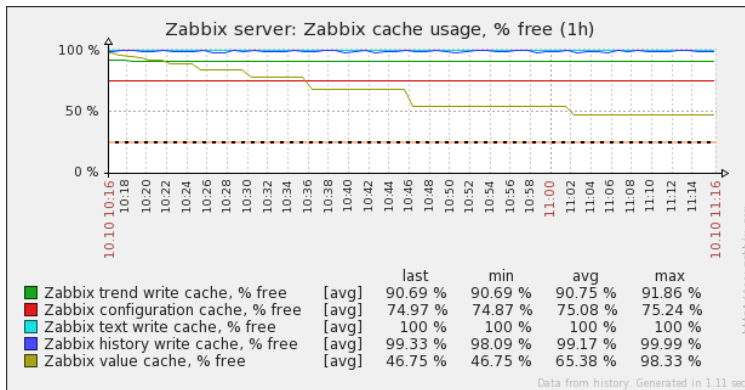
```
26791:20131009:144349.698 value cache is fully used: please increase ValueCacheSize
```

configuration parameter

value cache の情報は以下のアイテムで取得できます。

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

これらは Template App Zabbix Server にデフォルトで設定されています。
キャッシュの使用率をグラフで表示したものが以下になります。



まとめ

本レポートでは Zabbix 2.2 の α 版、 β 版を用いて、その新機能や改良点について検証を行いました。
Zabbix 2.2 では、以下の点が改良されたことが確認できました。

- 大幅な性能向上が行われた
- データベースの自動アップグレードによりアップデートが省力化された
- Web 監視の利便性が大きく向上した
- VMware 監視が実装され、簡単に監視を開始できる
- ローダブルモジュールにより性能を犠牲にしない機能拡張が可能になった