

MAGIC OF THE NEW ZABBIX AGENT

Alexey Pustovalov

Zabbix LLC サポートエンジニア兼トレーナー

ZABBIX 2019
Conference
JAPAN

#ZabConfJp2019

What is Zabbix Agent ?

MAGIC OF THE
NEW ZABBIX AGENT

ZABBIX 2019
Conference
JAPAN

#ZabConfJp2019

Zabbixエージェントとは

- 各OS内部のメトリクスを収集するデーモンプロセス
- UNIX系OSやWindowsなど様々なOSに対応
- 低リソースで動作可能
- 監視間隔は柔軟にカスタマイズ可能

Zabbixエージェント監視機能

- ディスク使用率、ネットワークの使用率...
- メモリ使用率、CPU使用率...
- ホスト名やOSのバージョンのような汎用的な情報
- ポート監視

Zabbixエージェント監視機能

- ログ監視やファイルのコンテンツ
- Windows イベントログ
- Zabbixサーバー、Zabbixプロキシのパフォーマンス情報
- など

詳細はドキュメントを参照

監視機能の拡張

カスタム
モジュール

ユーザー
パラメータ

system.run
[*]

カスタムモジュール

ジェダイのようなコーディング

- ・ネイティブZabbix Agentメトリックと同じぐらいの速さ
- ・C 言語の学習が必要



```
main {  
    printf("How to C learn, young padawan!")  
}
```

ユーザーパラメータ

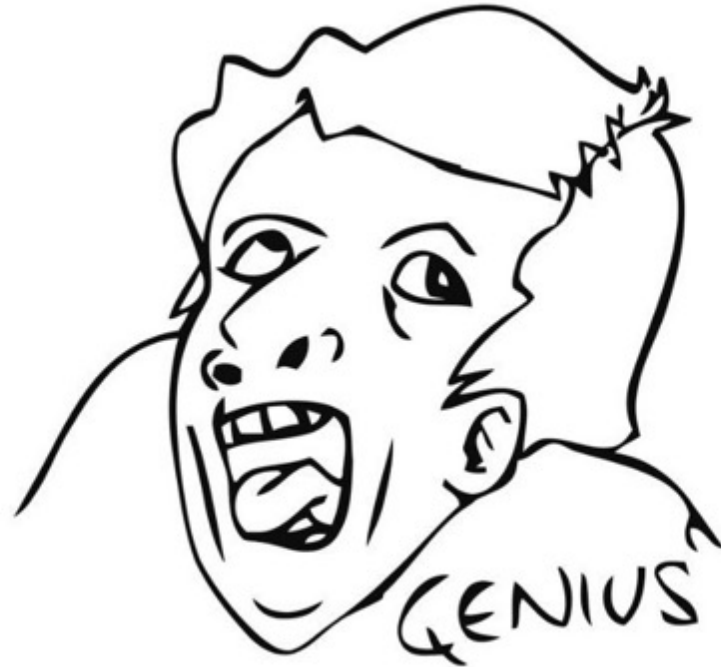
忍者のようにスクリプトを組み込める！

- Zabbixエージェントからコマンド実行することが可能
- フォークしてスクリプト実行
- よりエージェントを使いこなせる！



ユーザーパラメータ

```
UserParameter=awesome.command, sleep 45 &&  
echo "I'm a scripting genius!"
```



system.run[*] – シンプルだけどパワフル

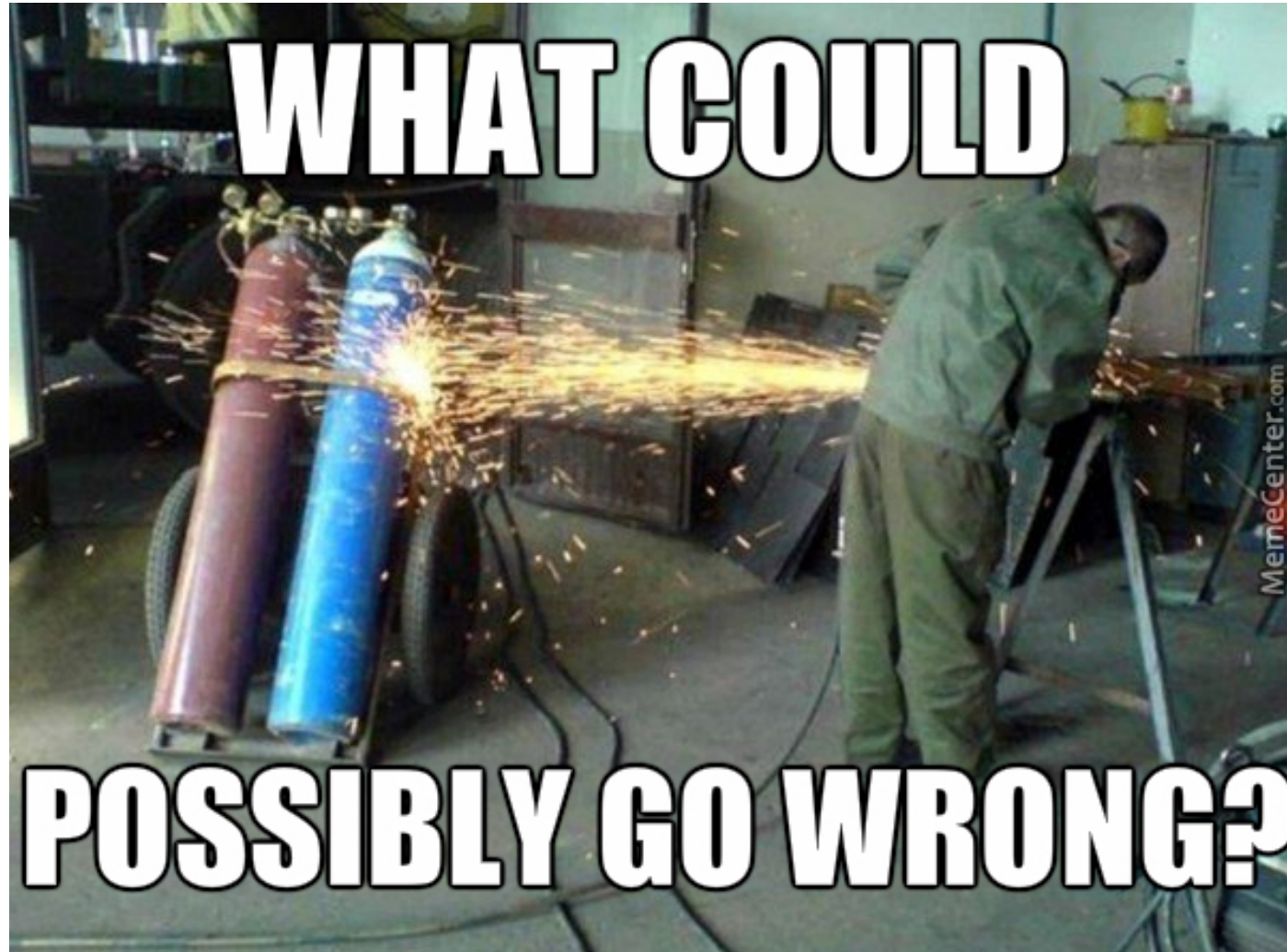
- ユーザパラメーターまたはファイルをあらかじめエージェントに設定しなくても利用できる
- コマンド変更する場合でもZabbixエージェントを再起動する必要がない
- 「大いなる力には大きな責任が伴う」



system.run[*]

- Server=0.0.0.0/0
 - EnableRemoteCommands=1
 - LogRemoteCommands=0
 - User=root
 - AllowRoot=1
-
- shell> iptables -F
 - shell> systemctl stop firewalld
 - shell> setenforce 0

system.run[*]



Current challenges ?

**MAGIC OF THE
NEW ZABBIX AGENT**

ZABBIX 2019
Conference
JAPAN

#ZabConfJp2019

Zabbixエージェントの機能制限

- 他のソフトウェアからのデータの受け取り
- アクティブチェックは監視間隔のカスタマイズに未対応
- 1つのアクティブチェックのプロセスが、1つのサーバー/プロキシを担当
- モジュール開発やソースの理解にはC言語の学習が必要

Meet the brand new Agent

**MAGIC OF THE
NEW ZABBIX AGENT**

ZABBIX 2019
Conference
JAPAN

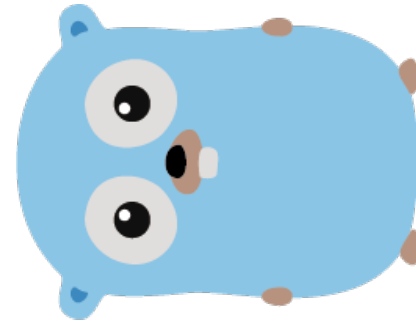
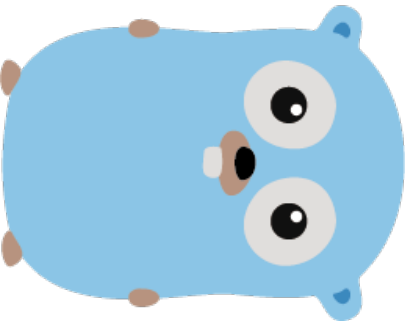
#ZabConfJp2019



ZABBIX 2019
Conference
JAPAN

NEW !

- C 言語ではない
- Go! 言語



Why Go?

- コンパイルして単一バイナリで実行!*
 - 近年急成長プログラミング言語で人気が高い
 - C 言語よりも簡単
-
- * 場合により他のソフトウェアへの依存関係が発生する場合があります。

拡張性高いZabbixエージェント2

- アクティブチェックでも監視間隔をカスタマイズ可能
- 現行の.confファイルのサポート
- ログファイル読み取りの並行動作
- すぐに使えるsystemdモニタリング！
- プラグイン単位のタイムアウトが実装可能

コンポーネント -1

- Connector
 - Zabbixサーバーとの通信を管理
 - アイテム設定
 - 監視データのバッファ
 - 1つのServerActiveにつき1つのコネクター

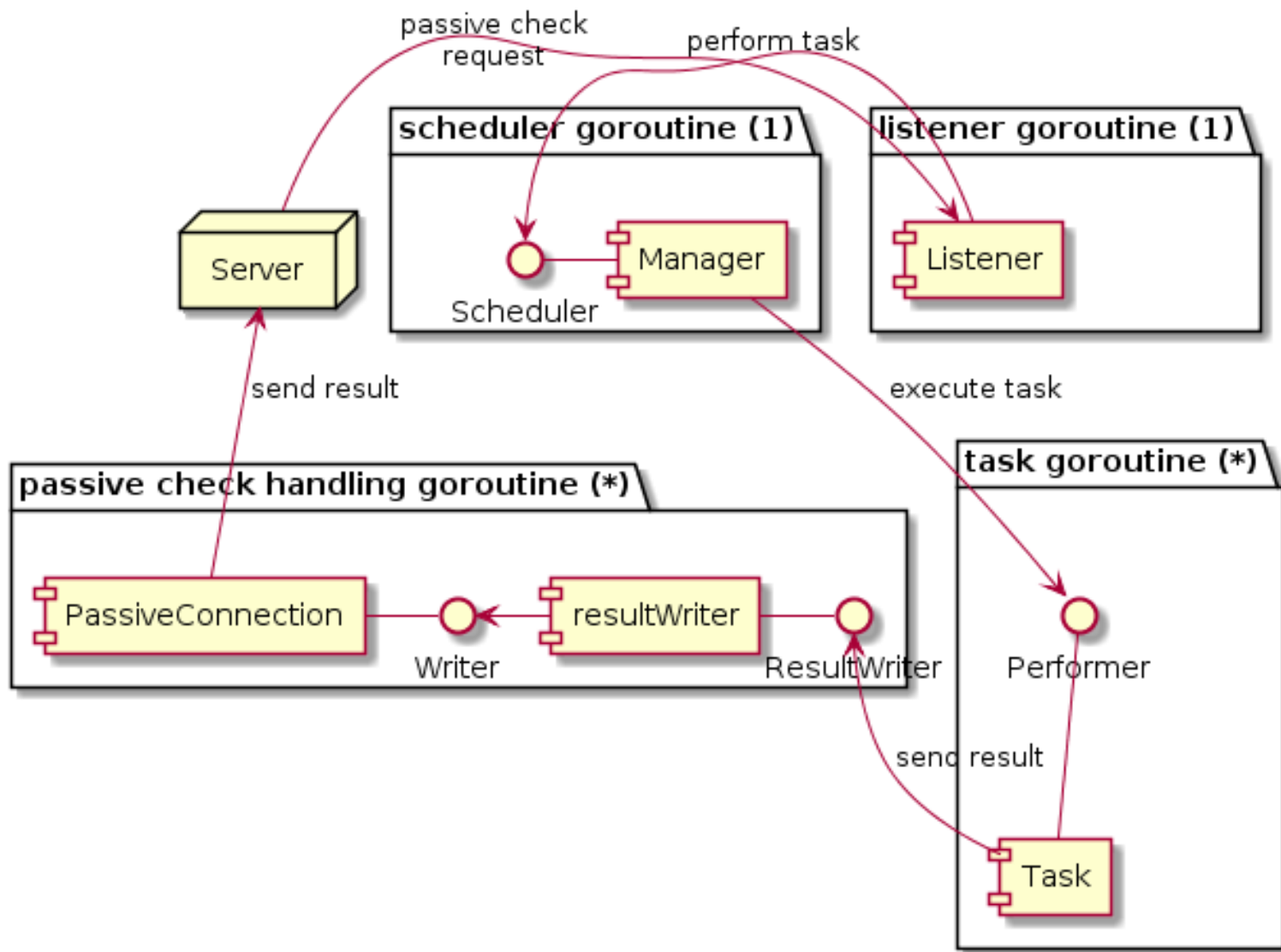
コンポーネント -2

- Listener
 - パッシブチェックのリクエストを受け取ってスケジューラーに渡す

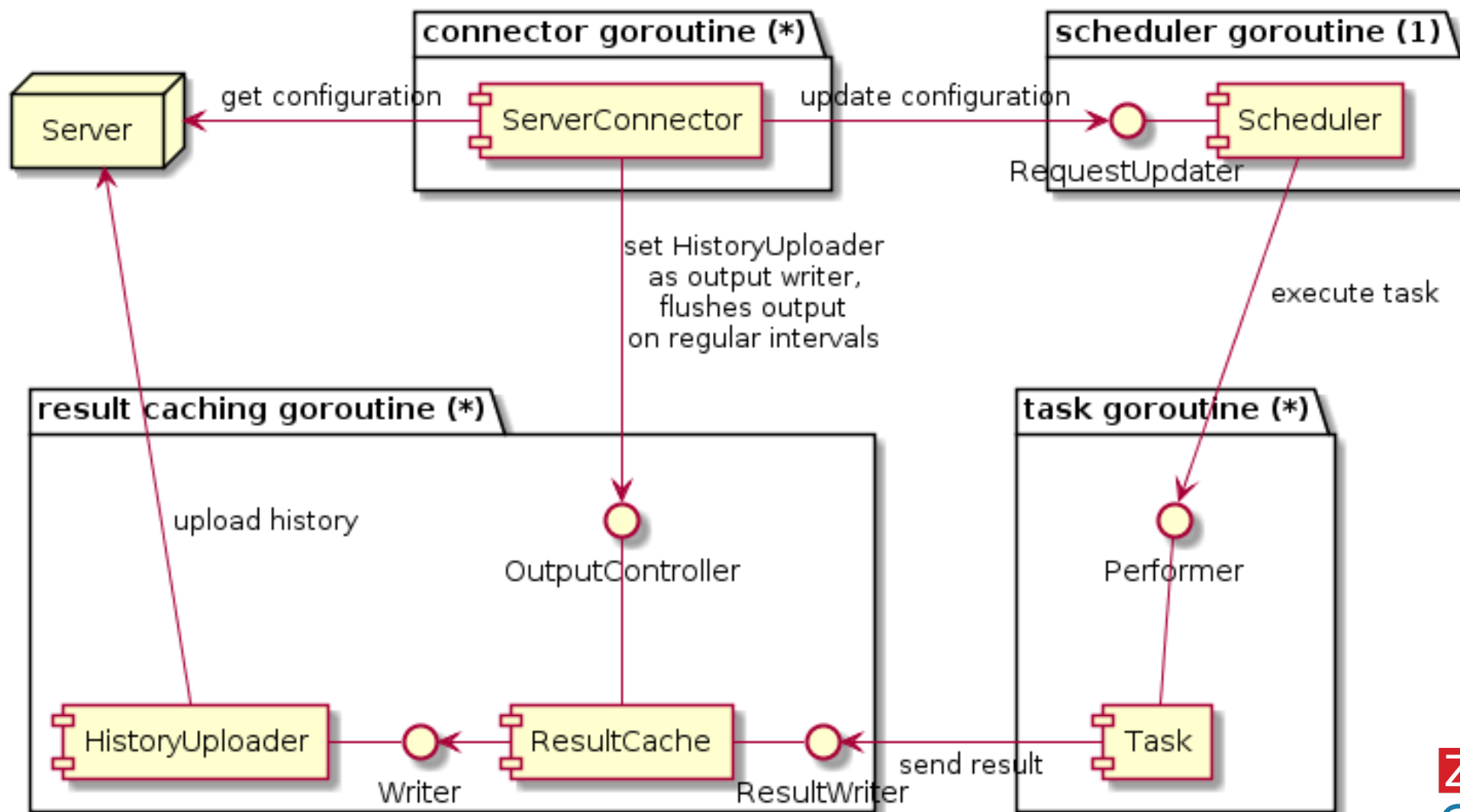
コンポーネント -3

- Scheduler
 - スケジュールに基づくタスクキューと、タスクの並列実行を管理
 - エージェントごとに1つのスケジューラー

パッシブチェック(ポーリング)

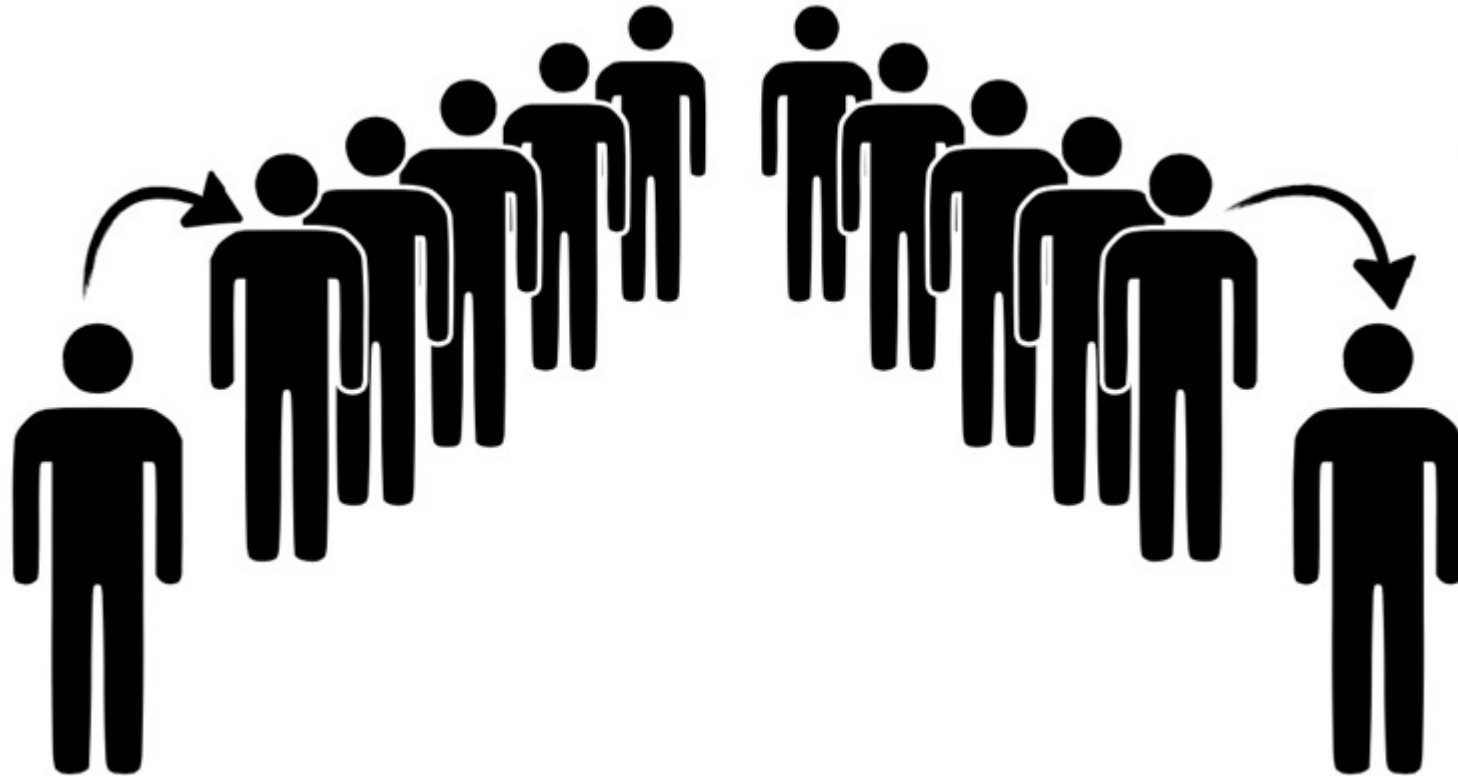


アクティブチェック



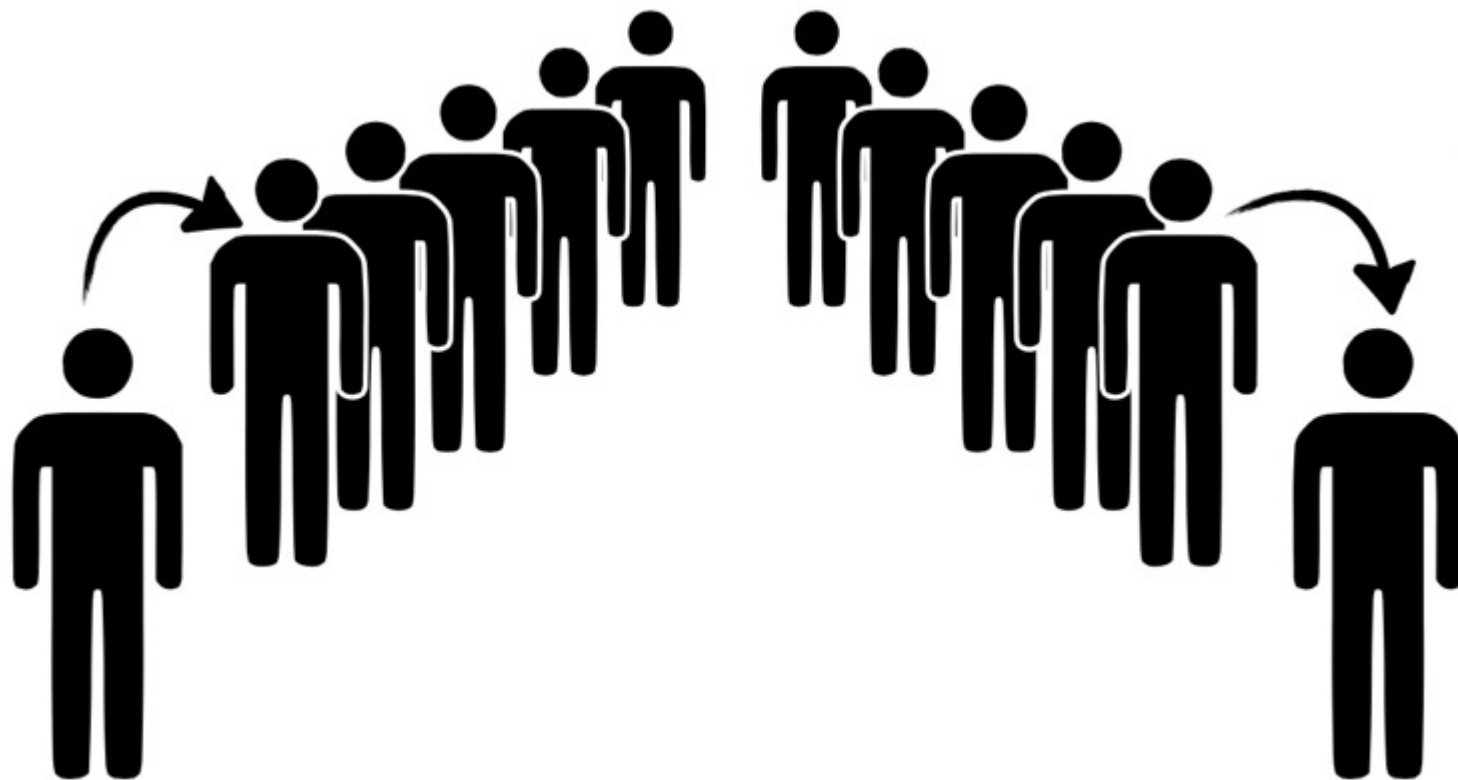
タスクキュー

- 2つのキュー:
 - プラグインごとの監視処理待ちのキュー
 - スケジューラーのアクティブなプラグイン全体のキュー



タスクキュー

- 並列処理の上限により監視処理を実行できない場合、プラグインはスケジューラーのキューを取得し、順番に処理を実行



It's all about the plugins

**MAGIC OF THE
NEW ZABBIX AGENT**

ZABBIX 2019
Conference
JAPAN

#ZabConfJp2019

完全に新しいプラットフォーム

- 単なる別実装のZabbixエージェントではなく、
- 完全に新しいプラットフォーム

だれもが開発者

- Zabbixエージェントの独自のプラグインを作成するのは簡単ではなかった
- 異なるタイプのタスクに対応する、複数のプラグインインターフェイス

様々な目的のためのフレームワーク

- Exporter
- Runner
- Watcher
- Collector

Exporter

- ポーリングを実行し値を返す
 - 何も返さない
 - エラー
 - 単一の値 / 複数の値

Watcher

- 独自の監視データの収集方法を実装
 - 特定のポートをオープンする
 - 異なったトラップベースの通信方法を実装することが可能

Collector

- たとえばCPUのリソース使用状況の監視など、定期的な監視データの収集のために利用できる

Runner

- たとえばバックグラウンドのgoroutineの開始/停止を実装することができる

プラグインの開発方法

1. コードを書く

```
return string("Hello World!")
```

2. メトリクスを設定

```
plugin.RegisterMetrics(&impl, "HelloWorld", "helloworld", "My  
hello to the World!")
```

3. ソースフォルダにプラグインを置く

4. リストにプラグインを加える

```
_ "zabbix/plugins/helloworld"
```

5. コンパイル

```
./configure --enable-agent2 && make install
```

6. 実行し、現行の.confファイルを再利用

```
package helloworld

import (
    "zabbix/pkg/plugin"
)

type Plugin struct {
    plugin.Base
}

var impl Plugin

func (p *Plugin) Export(key string, params []string, ctx plugin.ContextProvider) (result
interface{}, err error) {

    return string("Hello World!"), nil
}

func init() {
    plugin.RegisterMetrics(&impl, "helloworld", "helloworld", "My hello to the World!")
}
```

開発状況

- 名称は Zabbix エージェント2
- Zabbix 4.4 ではテスト的に実装
- Zabbix 5.0より正式サポート予定
- 現時点ではLinux系のOSのみに対応
- Windows向けZabbixエージェント2 – 開発中!
- ドキュメントも近日中にGit上に公開予定

Thank you !

MAGIC OF THE
NEW ZABBIX AGENT

ZABBIX 2019
Conference
JAPAN

#ZabConfJp2019