

L'art d'ordonnancer

avec JobScheduler

François BAYART

30 Octobre 2010



[Look a box](#)

Introduction



Qui suis-je ?

François Bayart

consultant système en solution libre et propriétaire

Redhat (1996), SuSE (1997) et Debian (1999)



Projet JobScheduler

- Projet sous 2 licences
OpenSource GPL
Commercial (sans les contraintes GPL)
- Développé par SOS-Berlin
(Software- und Organisations-Service GmbH)
- Scheduler pour Unix / Linux / Windows



Explication

1 [2] 34567 explication



Définition d'un "scheduler"

Ordonnanceur = scheduler != planificateur

- Définition simpliste :
 - Traitement : Action ou suite d'action à effectuer
 - Planification : Plage horaire de démarrage
 - Ressources : Eléments externes conditionnant le démarrage
 - Machines : Physique ou virtuelle, voir process dans une file d'attente
 - Utilisateur : Compte de soumission



Les petits plus

Adapter l'ordonancement suite à des évènements externes non prévisible

Redémarrage automatique et reprise lors d'incidents

Alerter et informer l'équipe des opérations / exploitation

Générer un rapport d'incident



Crontab forever !

Planificateur pure

- Planification = déclencheurs
 - Pas de queue
 - Pas de contraintes
 - Pas d'enchaînement



qu'est ce que le scheduling

Au minima

- Pouvoir définir un “workflow” et les dépendances
- Permettre de gérer l'exécution des tâches
- Surveiller l'exécution des tâches
- Pouvoir définir les priorités et “queue” des jobs



Technologies

L'application de base:

- ✓ Java, C++
- ✓ WebServices (XSLT-Stylesheets)
SOAP par défaut
- ✓ API : Java, Perl, VBScript, JavaScript

Scripting possible :

- ✓ JavaScript
- ✓ PHP
- ✓ Perl
- ✓ VB Scripts

GUI Editor:

- ✓ Java

Web Editor:

- ✓ PHP

Managed Jobs:

- ✓ PHP
- ✓ SQL
(MySQL, PostgreSQL, MSSQL, Oracle)



La Communauté

- ✓ projet sur SourceForge
 - <http://jobscheduler.sourceforge.net/>
- ✓ forum de SourceForge (très vivant !)
 - Help , Support
- ✓ support de SOS Berlin
 - Très présent



Architecture



JobScheduler : StandAlone

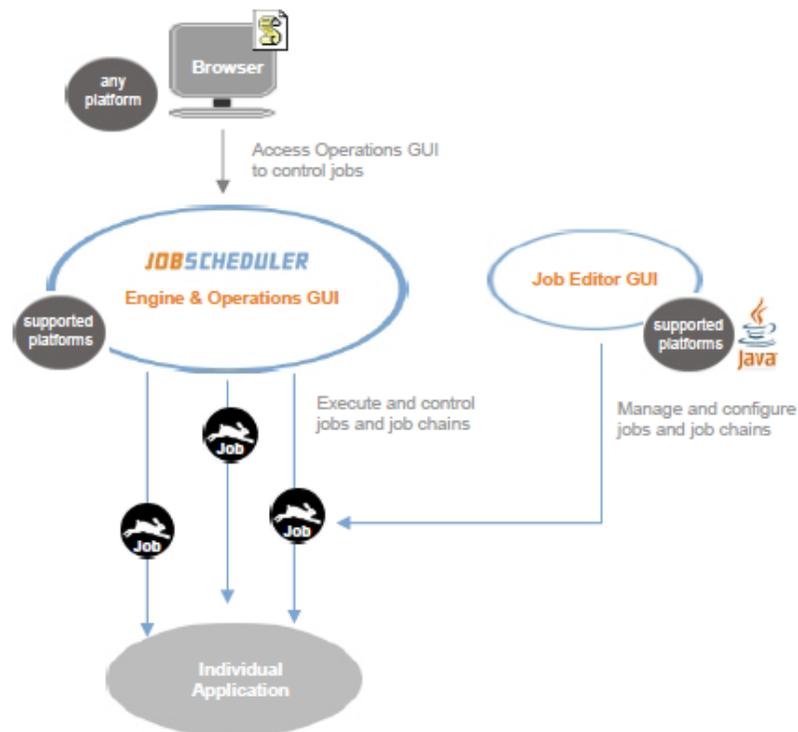
- ✓ fichiers XML
- ✓ éditeur graphique
- ✓ serveur HTTP intégré



JobScheduler : StandAlone

Architecture

Single Server Operation



At a Glance:

- You can operate the Job Scheduler on a single server without a database.
- If you want to operate Job Scheduler without a database, no additional software is required.
- The implementation of the Job Scheduler includes a built-in Operations GUI.
- You can use the Job Editor GUI to configure jobs that are stored as XML files.
- Use any of the supported platforms.

By the way!

- You can use any editor of your choice to manage job configurations in XML files.



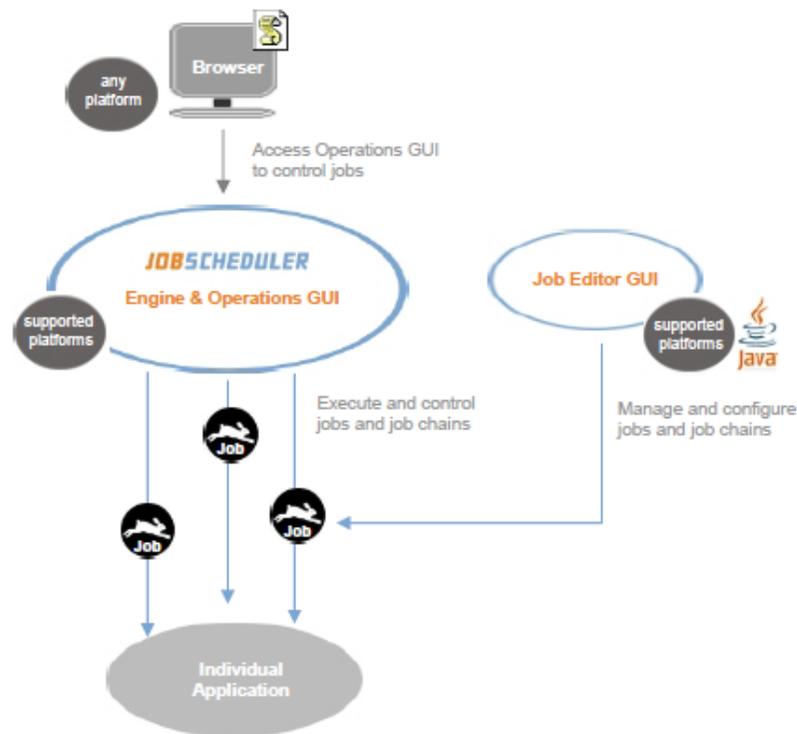
JobScheduler : StandAlone

fichiers XML
éditeur graphique

Serveur HTTP intégré

Architecture

Single Server Operation



At a Glance:

- You can operate the Job Scheduler on a single server without a database.
- If you want to operate Job Scheduler without a database, no additional software is required.
- The implementation of the Job Scheduler includes a built-in Operations GUI.
- You can use the Job Editor GUI to configure jobs that are stored as XML files.
- Use any of the supported platforms.

By the way!

- You can use any editor of your choice to manage job configurations in XML files.

Software- und Organisations-Service GmbH

► www.sos-berlin.com



JobScheduler : Managed / multi-serveur

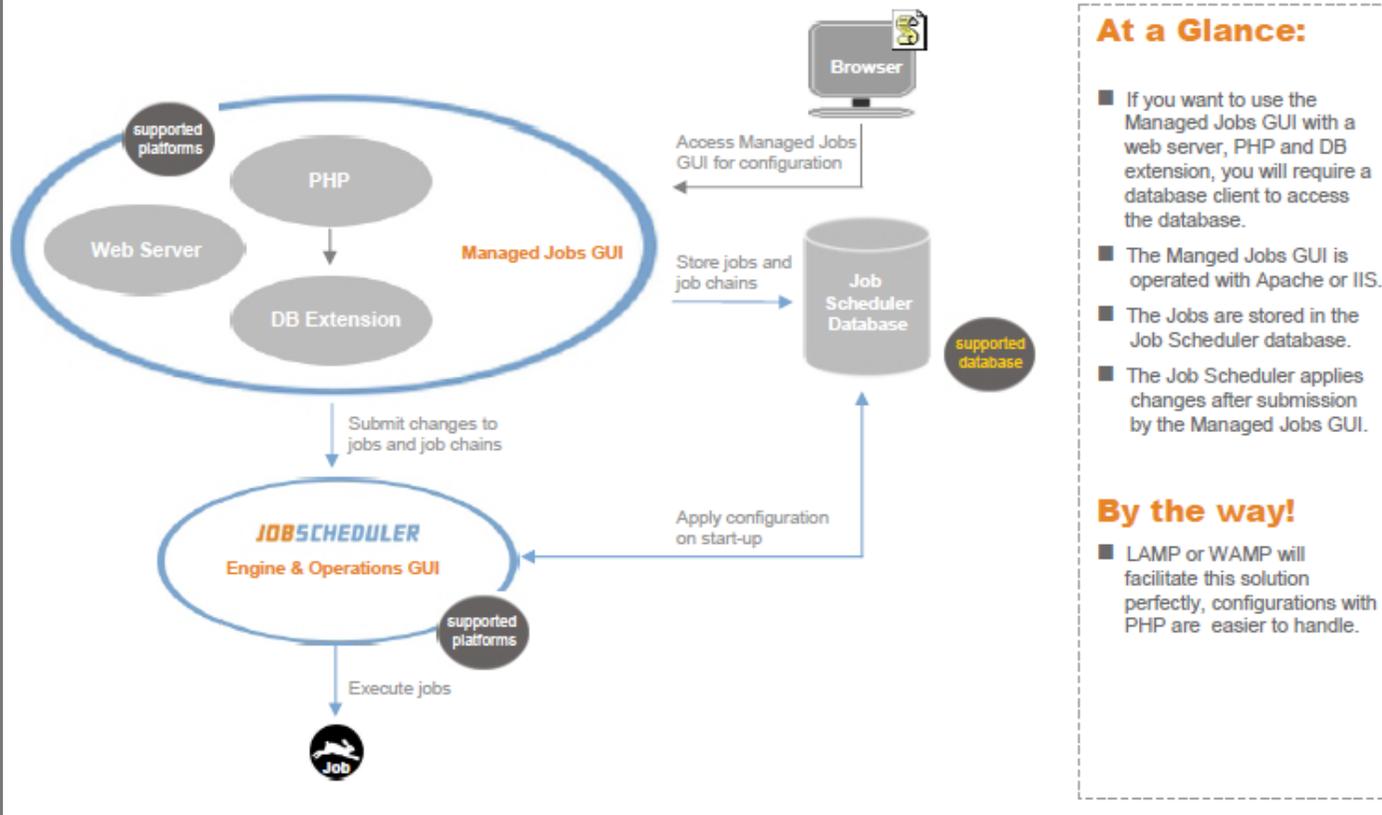
- ✓ frontend PHP
- ✓ connexion database
- ✓ agent ou client ou engine au choix



JobScheduler : Managed / multi-serveur

Job Configuration

Using Managed Jobs GUI



Software- und Organisations-Service GmbH

► www.sos-berlin.com



Quelques termes

supervisor : donneur d'ordre

engine & operations GUI : exécutant

Managed Jobs GUI : LAMP

workload :

” scheduler ” : agenda / planning

Job Editor

Hot Folders



Démonstration

En vrai c'est mieux !



Contexte de la démonstration

- JobScheduler Standalone
UM01 : Engine & Operations GUI

- JobScheduler Managed / mutli-serveurs
UM01 : Frontend PHP + DB + Supervisor & Operations GUI
UM02 : Engine & Operations GUI
UM03 : Engine & Operations GUI



Définition d'un job

Création d'un job « indépendant » de base

Création d'un job « indépendant » distant

Création d'une chaîne simple

Création d'une chaîne multi-serveurs

Gestion des « ressources » et des « schedule »



Définition des accès – ACL

Engine :

filtrage selon l'IP ou FQDN

none / signal / info / no_add / all

Frontends :

similaire à Unix (rwx / ???) avec 3 grands groupes
administrator , developer , operator



Mais encore ...

1234 [5] 67 mais encore ...



Architectures avancées

Load Balancing

H.A. Cluster



Monitoring

envoyer ses statuts vers un outils de monitoring

recevoir ses ordres de déclenchement depuis l'outil de monitoring qui aurait détecter un "event" anormal sur lequel il faut réagir



Transferts de fichiers

- ✓ vérifier la présence du fichier chez un tiers
- ✓ envoyer des fichiers chez un tiers
- ✓ Protocoles supportés : FTP, SFTP, FTPS, SCP (pas DAU)



Malheureusement

;-)



Aspects négatifs

- JUM 32 bits obligatoire
- Ergonomie de l'interface
- Documentation pas évidente
- Howto et Screencast Out of date



Le(s) mot(s) de la fin

123456 [?] le(s) mot(s) de la fin



Planifier et ordonnancer les tâches peut représenter beaucoup de travail, surtout lorsqu'ils s'exécutent sur plusieurs serveurs.

JobScheduler permet de simplifier cela.

BY JAMES MOHR



Liens Internet

- Cette présentation sur SlideShare
- SOS Berlin : <http://www.sos-berlin.com/>
- article linux mag :
<http://www.linux-magazine.com/Issues/2008/97/JOB-SCHEDULER-A-LA-CARTE>
- Wikipedia :
 - http://en.wikipedia.org/wiki/Job_scheduler
 - http://en.wikipedia.org/wiki/Open_Source_Job_Scheduler
- Regroupement de spécialistes en ordonnancement
<http://ordonnancement.org/>



Pas de question !

merci ...

