Supervision

Bruno BEAUFILS

2021/2022

Supervision

- Rappels des principes
  - FCAPS (**Fault**, Configuration, Accounting, **Performance**, Security)
  - Tâches à faire pour les outils de supervision
    - récolter les données (sondes, *poller*)
    - planifier la récolte (ordonnanceur de sondages, *scheduler*)
    - stocker les données (bases de données)
    - **utiliser** les données
- Système de supervision
  - 1 moniteur (client)
  - plusieurs agents (serveurs)
- 2 utilisations différentes
  1. métrologie : **mesurer** la perfomance
  2. surveillance : **vérifier** que le système fonctionne comme prévu

Supervision

- Rappels des principes
  - FCAPS (**Fault**, Configuration, Accounting, **Performance**, Security)
  - Tâches à faire pour les outils de supervision
    - récolter les données (sondes, *poller*)
    - planifier la récolte (ordonnanceur de sondages, *scheduler*)
    - stocker les données (bases de données)
    - **utiliser** les données
- Système de supervision
  - 1 moniteur (client)
  - plusieurs agents (serveurs)
- 2 utilisations différentes
  1. métrologie : **mesurer** la perfomance
  2. surveillance : **vérifier** que le système fonctionne comme prévu

Supervision

- Rappels des principes
  - FCAPS (**Fault**, Configuration, Accounting, **Performance**, Security)
  - Tâches à faire pour les outils de supervision
    - récolter les données (sondes, *poller*)
    - planifier la récolte (ordonnanceur de sondages, *scheduler*)
    - stocker les données (bases de données)
    - **utiliser** les données
- Système de supervision
  - 1 moniteur (client)
  - plusieurs agents (serveurs)
- 2 utilisations différentes
  1. métrologie : **mesurer** la perfomance
  2. surveillance : **vérifier** que le système fonctionne comme prévu

Supervision

- Rappels des principes
  - FCAPS (**Fault**, Configuration, Accounting, **Performance**, Security)
  - Tâches à faire pour les outils de supervision
    - récolter les données (sondes, *poller*)
    - planifier la récolte (ordonnanceur de sondages, *scheduler*)
    - stocker les données (bases de données)
    - **utiliser** les données
- Système de supervision
  - 1 moniteur (client)
  - plusieurs agents (serveurs)
- 2 utilisations différentes
  1. métrologie : **mesurer** la perfomance
  2. surveillance : **vérifier** que le système fonctionne comme prévu

Métrologie

Objectifs

- collecter des données sur le fonctionnement du système/réseau
- pour suivre les performances dans le temps

Exemples

- traffic réseau (entrant/sortant), requêtes HTTP (quantité, charge), etc.
- entrées/sorties disques, charge CPU, espace disque/mémoire (utilisé)

Tâches

- Récolter : SNMP très bien adapté
- Planifier : cron peut suffir
- Stocker : grandes séries temporelles
- Utiliser : présenter des graphes

Enregistrer et tracer des graphes

- **RRD** : Round Robin Database
  - un format de fichier pour **stocker efficacement** des séries de données temporelles
  - taille des fichiers (*archives*) **fixe**
    - déterminée par le nombre de points et l'intervalle de durée
    - les nouvelles données remplacent les anciennes
    - assure une consistance sur le système de fichier
  - données **consolidables** : conserver des informations sur les anciennes données
    - réduction de la résolution (fréquence)
    - fonctions de consolidation (moyenne, minimum, etc.)
  - fichiers **indépendant de la source** des données
  - utilisé par la plupart des outils de supervision
- RRDtool
  - écrit par Tobias OETIKER (1999)
  - 2 services offerts
    - gestion des archives RRD
    - création de graphes
  - utilisé par **beaucoup** d'outils de graphes
- Historique : MRTG (Multi Router Traffic Grapher)
  - écrit par Tobias OETIKER (1995)
  - essentiellement captation SNMP

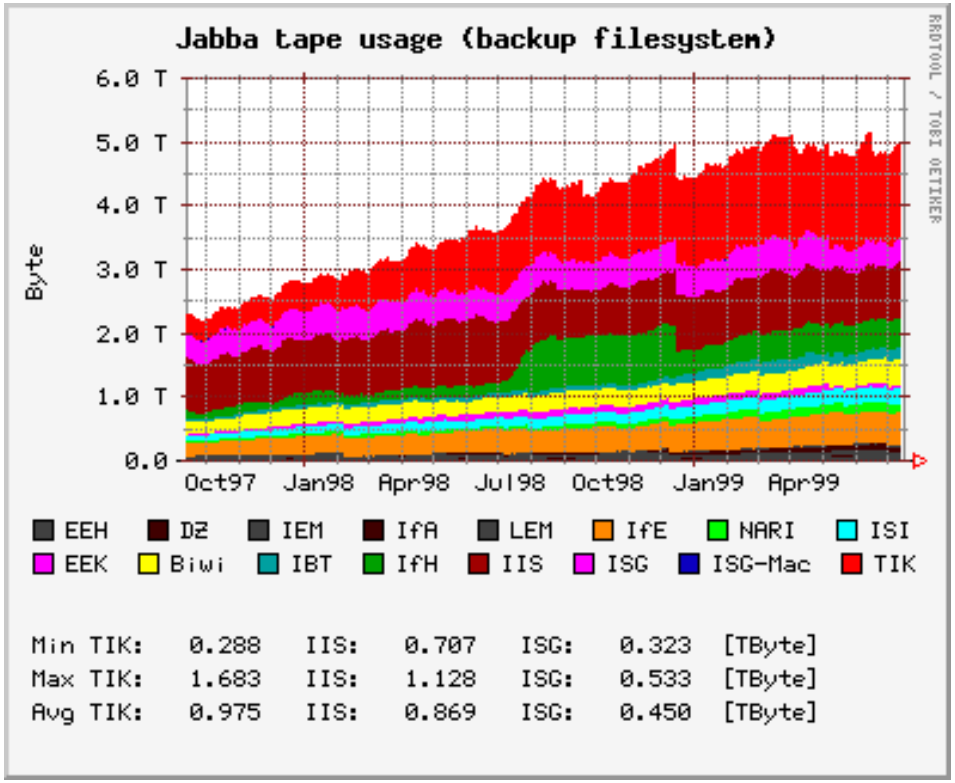
Enregistrer et tracer des graphes

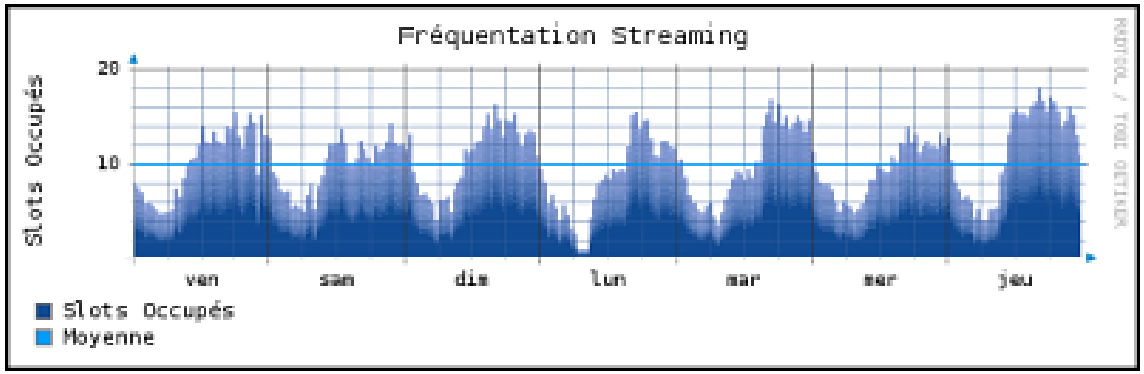
- **RRD** : Round Robin Database
  - un format de fichier pour **stocker efficacement** des séries de données temporelles
  - taille des fichiers (*archives*) **fixe**
    - déterminée par le nombre de points et l'intervalle de durée
    - les nouvelles données remplacent les anciennes
    - assure une consistance sur le système de fichier
  - données **consolidables** : conserver des informations sur les anciennes données
    - réduction de la résolution (fréquence)
    - fonctions de consolidation (moyenne, minimum, etc.)
  - fichiers **indépendant de la source** des données
  - utilisé par la plupart des outils de supervision
- RRDtool
  - écrit par Tobias OETIKER (1999)
  - 2 services offerts
    - gestion des archives RRD
    - création de graphes
  - utilisé par **beaucoup** d'outils de graphes
- Historique : MRTG (Multi Router Traffic Grapher)
  - écrit par Tobias OETIKER (1995)
  - essentiellement captation SNMP

Enregistrer et tracer des graphes

- **RRD** : Round Robin Database
  - un format de fichier pour **stocker efficacement** des séries de données temporelles
  - taille des fichiers (*archives*) **fixe**
    - déterminée par le nombre de points et l'intervalle de durée
    - les nouvelles données remplacent les anciennes
    - assure une consistance sur le système de fichier
  - données **consolidables** : conserver des informations sur les anciennes données
    - réduction de la résolution (fréquence)
    - fonctions de consolidation (moyenne, minimum, etc.)
  - fichiers **indépendant de la source** des données
  - utilisé par la plupart des outils de supervision
- RRDtool
  - écrit par Tobias OETIKER (1999)
  - 2 services offerts
    - gestion des archives RRD
    - création de graphes
  - utilisé par **beaucoup** d'outils de graphes
- Historique : MRTG (Multi Router Traffic Grapher)
  - écrit par Tobias OETIKER (1995)
  - essentiellement captation SNMP

RRDtool : exemples





Crédits : Tobias OETIKER

Panorama des outils (intégrés, basés sur RRDtool)

- SmokePing - https://oss.oetiker.ch/smokeping
  - écrit en Perl
  - architecture simple, configuration par fichiers
  - nombreuses sondes
- Munin - http://munin-monitoring.org
  - écrit en Perl
  - architecture simple, configuration par fichiers
  - extension par plugings facile
- Cacti - https://www.cacti.net
  - écrit en PHP
  - spécialisé dans la métrologie réseau
  - application complexe, configuration par interface web,

Panorama des outils

- collectd - https://collectd.org
  - écrit en C
  - ne s'intéresse qu'à la collecte (délégue le reste)
  - beaucoup d'extensions
- Graphite – https://graphiteapp.org
  - écrit en Python
  - mêmes services/objectifs que RRDTool
  - pas de collectes des données (stockage et graphes uniquement)
- Prometheus – https://prometheus.io
  - écrit en Go
  - solution complète (collecte, stockage, utilisation)
  - collecte via HTTP
  - langage de requête des données
- Grafana - https://grafana.com
  - écrit en Go
  - application web de visualisation, alertes (pas de collecte)
  - utilisé sur différentes bases (RRD, Prometheus, etc.)
  - permet de définir des tableaux de bords (exportables)

Supervision

- Rappels des principes
  - FCAPS (**Fault**, Configuration, Accounting, **Performance**, Security)
  - Tâches à faire pour les outils de supervision
    - récolter les données (sondes, *poller*)
    - planifier la récolte (ordonnanceur de sondages, *scheduler*)
    - stocker les données (bases de données)
    - **utiliser** les données
- Système de supervision
  - 1 moniteur (client)
  - plusieurs agents (serveurs)
- 2 utilisations différentes
  1. métrologie : **mesurer** la perfomance
  2. surveillance : **vérifier** que le système fonctionne comme prévu

Surveillance

Objectifs

- vérifier activement que le système/réseau fonctionne comme prévu
- détecter les problèmes avant les utilisateurs
- **alerter**

Exemples :

- vérifier qu'une machine sur un segment réseau est accessible
- essayer de récupérer une page web et vérifier la réponse

Tâches

- Récolter : s'intéresse plus aux services qu'aux ressources
  - besoin de sondes ad-hoc
- Planifier :
- Stocker : état
- Utiliser : alerter (tableaux de bords, email, SMS)

Suites logicielles

- **Intégration**
  - collecte, stockage, utilisation
  - faciliter l'extension
- Services offerts
  - métrologie
  - alertes
  - rapport
  - réponses automatiques
- Quelques conventions
  - console web
    - facilite la construction de tableau de bord
  - code couleur
    - vert = OK
    - orange = dégradé
    - rouge = KO
  - alertes par email (ou SMS)
- Futur
  - liaison avec la **gestion de configuration** (*configuration management*)

Suites logicielles

- **Intégration**
  - collecte, stockage, utilisation
  - faciliter l'extension
- Services offerts
  - métrologie
  - alertes
  - rapport
  - réponses automatiques
- Quelques conventions
  - console web
    - facilite la construction de tableau de bord
  - code couleur
    - vert = OK
    - orange = dégradé
    - rouge = KO
  - alertes par email (ou SMS)
- Futur
  - liaison avec la **gestion de configuration** (*configuration management*)

Suites logicielles

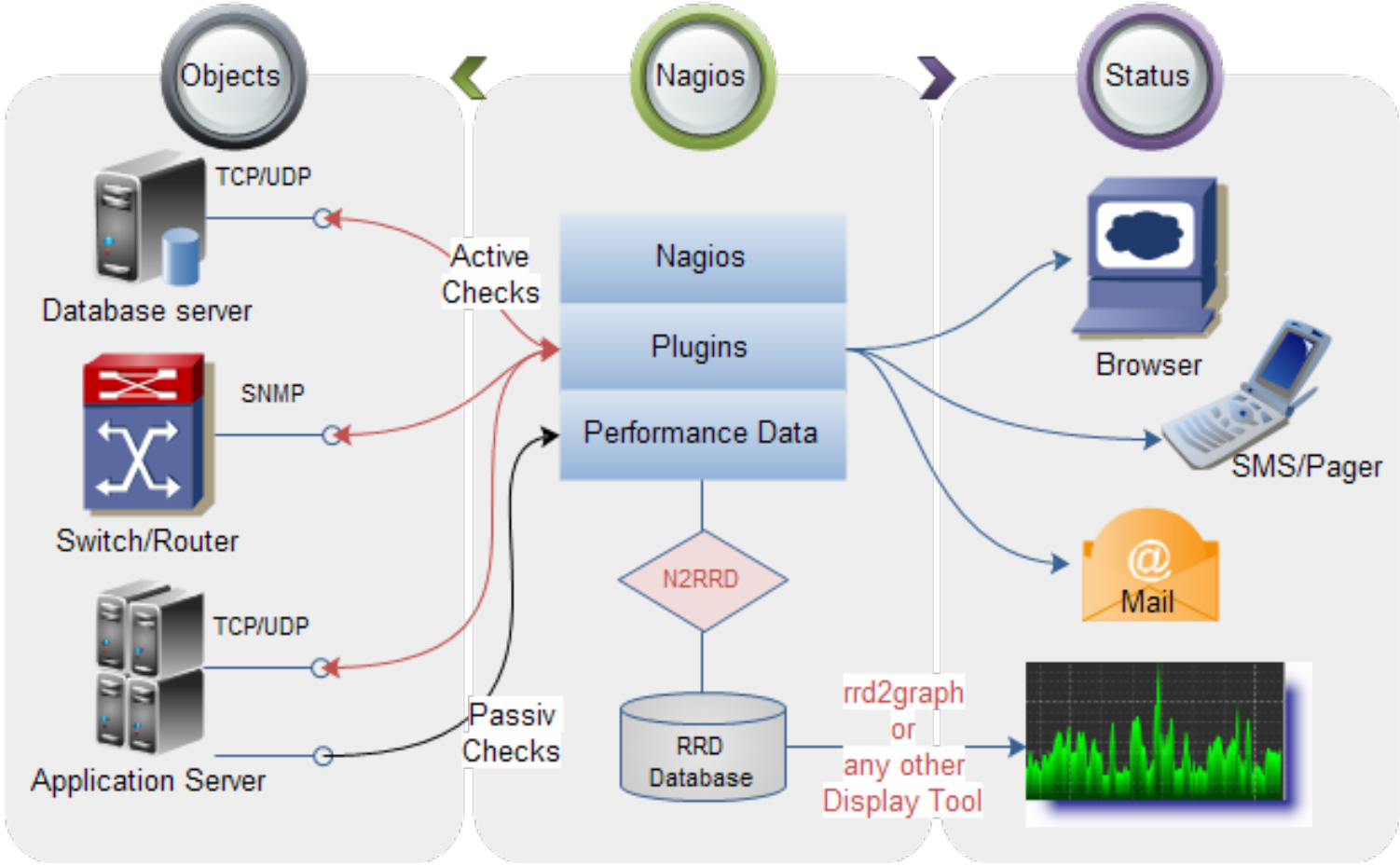
- **Intégration**
  - collecte, stockage, utilisation
  - faciliter l'extension
- Services offerts
  - métrologie
  - alertes
  - rapport
  - réponses automatiques
- Quelques conventions
  - console web
    - facilite la construction de tableau de bord
  - code couleur
    - vert = OK
    - orange = dégradé
    - rouge = KO
  - alertes par email (ou SMS)
- Futur
  - liaison avec la **gestion de configuration** (*configuration management*)

Nagios la référence historique

Nagios - http://www.nagios.org

- Histoire
  - débuté en 1999 (NetSaint) logiciel libre
  - ~ 2009 mécontentement de la communauté sur la gestion du projet
  - création de nombreux forks (dérivé)
  - le code historique passe en modèle *Open Core*
    - le cœur est nommé Nagios Core
- **LA** référence
  - ses principes techniques ont inspirés la plupart des solutions actuelles
  - le coeur (*core*) est libre le reste pas forcément (Open Core)
    - une partie est gérée par la communauté
    - une autre par une entreprise

Nagios Core : principes



Crédits : Wikipedia (Public Domain)

Nagios Core : composants principaux

- NRPE (*Nagios Remote Plugin Executor*)
  - plugin `check_nrpe`
  - récupération de données via l'exécution de scripts à distance
- Nagios Plugins
  - conventions d'écriture de plugins
  - utilisés par beaucoup de systèmes dérivés
  - Exemple : Nagvis – `http://www.nagvis.org`
    - représentation graphique

Panorama des outils

Basés sur Nagios

- Icinga - https://www.icinga.com
  - fork de Nagios en 2009
  - v1 proche de Nagios, v2 ré-écriture complète
- Shinken - http://www.shinken-monitoring.org
  - ré-écriture de Nagios en python
  - architectures distribuées
- Centreon - https://www.centreon.com/solutions
  - entreprise française (populaire en France)
  - même concepts que Nagios

Autres concepts que Nagios

- OpenNMS - https://www.opennms.org
- Zabbix - https://www.zabbix.com

Privateurs

- IBM Tivoli
- HP OpenView

Comparaison

https://en.wikipedia.org/wiki/Comparison_of_network_monitoring_systems