

## ***Termes généralement utilisés***

- Logiciels de supervision réseau et système
- Outils de monitoring IT
- Solutions de surveillance des infrastructures

## ***Les solutions « legacy » (historiques)***

- Historiquement, outils différents
  - Pour la supervision de l'infrastructure réseau.
  - Pour la supervision système.
- Pourquoi ?
  - Équipements réseaux :
    - Difficile/impossible de déployer des agents.
    - Supervision via des protocoles et outils normalisés : SNMP, ping, Netflow, syslog, ssh, TR-069.
  - Systèmes, serveurs :
    - Informations disponibles via SNMP insuffisantes pour les admin sys.
    - Difficile de trouver des personnes compétentes en SNMP pour le faire évoluer en fonction des besoins.
    - Facilité pour un admin sys de développer des scripts sur mesure (bash, perl, python).
    - Développement et déploiement d'agents de supervision : exemples de données supervisés : suivi des utilisateurs logués, suivi des processus, utilisation CPU et mémoire.
- Depuis 15 ans, logiciels de supervision (Zabbix, Centreon, Nagios) gèrent « convenablement » réseaux et systèmes. Ils reposent principalement sur un agent à déployer et supportent relativement bien SNMP.

# Les logiciels de supervision système et réseau

- Nagios/Zabbix : avant tout des **ordonnanceurs**
  - Lance à intervalles réguliers des plugins (sondes)
  - Récupère les informations renvoyées par les sondes
    - Notamment la valeur de retour (0 → OK)
  - Analyse les informations recueillies :
    - Si elles ne dépassent pas des limites fixées par l'administrateur → OK
    - Si dépassent les limites → envoie de notifications (mail, SMS ...)
  
- La modularité/flexibilité de Nagios/Zabbix grâce aux plugins :
  - Plugins : logiciels / scripts bash, Python, Perl
    - Conception assez simple
  - Exemple : pour le test d'accès à un annuaire ldap, il n'est pas très compliqué d'écrire un script qui se connecte à un annuaire, effectue une recherche et
    - Renvoie 0 si OK.
    - Temps d'exécution de la commande.
  - Plugins développés par la « communauté » et mis à disposition.
  - Des plugins sont également définis pour avertir ou réagir (envoi d'email, de sms, actions à entreprendre).

## ***Les solutions « récentes »***

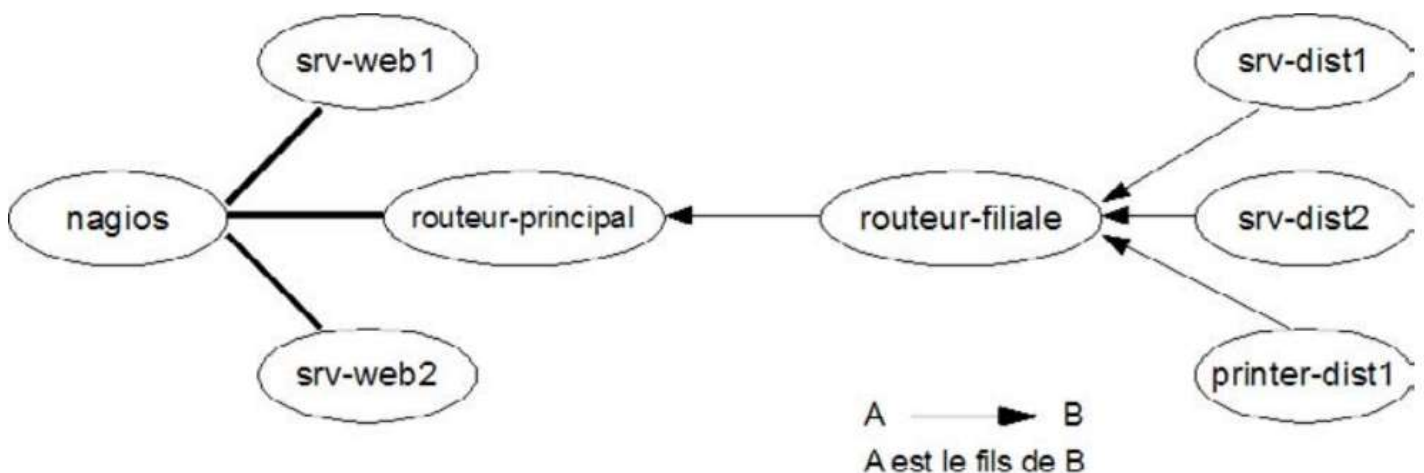
- Evolution des SI :
  - Architectures microservices
  - Infrastructures dynamiques (orchestrateurs) : les services migrent automatiquement d'un hôte vers un autre.
- Evolution de la supervision :
  - Supervision traditionnelle : collecte de métriques systèmes (CPU, mémoires, logs) insuffisante.
  - Apparition de nouveaux système de supervision : Prometheus.
  - Plus récemment : analyse en temps réel des interactions entre les services : voir opentelemetry. Dynatrace
- Prometheus
  - Conçu pour les architectures modernes (cloud, microservices, conteneurs, Kubernetes).
  - Peut fonctionner de manière décentralisée et distribuée.
  - Prometheus met en œuvre :
    - Time-Series Database (TSDB)
    - Langage de requête : promql
  - La visualisation des données (dashboard) repose généralement sur Grafana.
  - La gestion des alertes repose sur Alertmanager

## ***Comment choisir un logiciel de supervision? Exemple de critères de choix.***

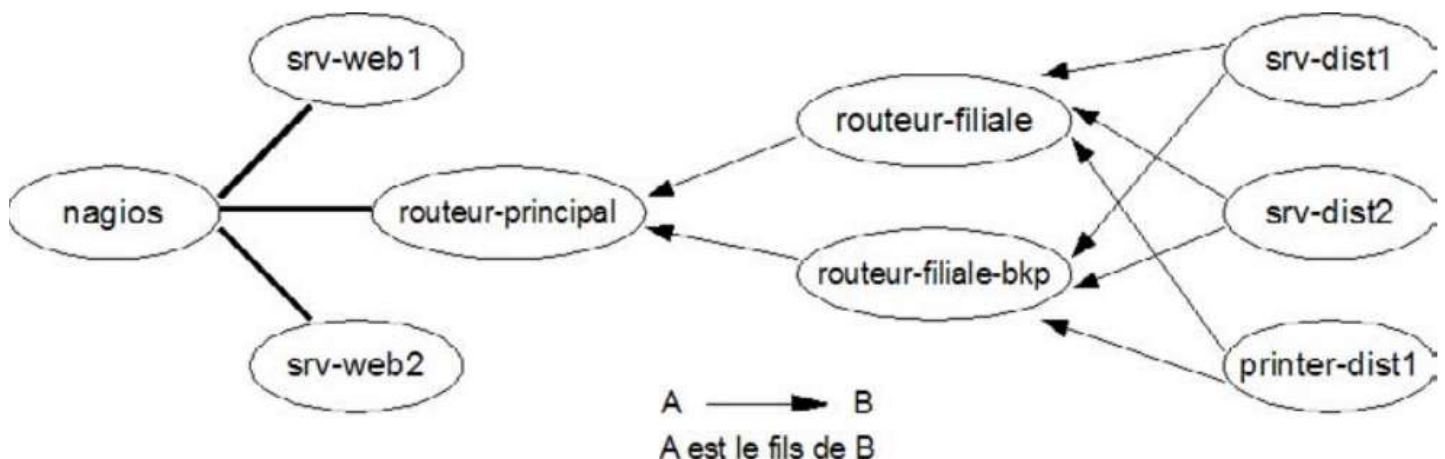
- Le logiciel est-il en adéquation avec
  - L'infrastructure ?
  - Avec les objectifs listés dans la phase de planification ?
- Le logiciel doit-il être open source ?
- Si Open source :
  - Logiciel stable ? performant ?
  - Importance de la communauté et du support ?
  - Référence dans le monde de la supervision Open Source ?
- Si propriétaire ?
  - La solution implique-t-elle une forte dépendance à l'éditeur ?
  - Quelle est la flexibilité/évolutivité de la solution ?
  - Quelle est la sécurité ? Code auditable ?
- Caractéristique essentielle : capacités à gérer un parc important :
  - Performances → Combien de tests par seconde ?
  - Facilité de configuration et d'ajout de métriques ?
  - Gestion des configurations via des modèles ?
  - Analyse et synthèse intelligente des alertes ?
  - Haute disponibilité ?
  - Répartition de charge (scalabilité) : deux aspects essentiels pour des très gros SI à superviser.

## Analyse et Synthèse des alertes

- SI : éléments inter-dépendants. Si l'un d'eux rencontre des problèmes, répercussion sur d'autres éléments.
- Un simple dysfonctionnement ==> un grand nombre d'alertes.
- Difficultés à trouver, parmi toutes les alertes, la **cause initiale du problème**.
- Le logiciel de supervision gère-t-il des relations de dépendances. Ces relations peuvent être :
  - physiques (exemple des liens réseaux)
  - virtuelles (comme c'est le cas entre une application et une base de données).



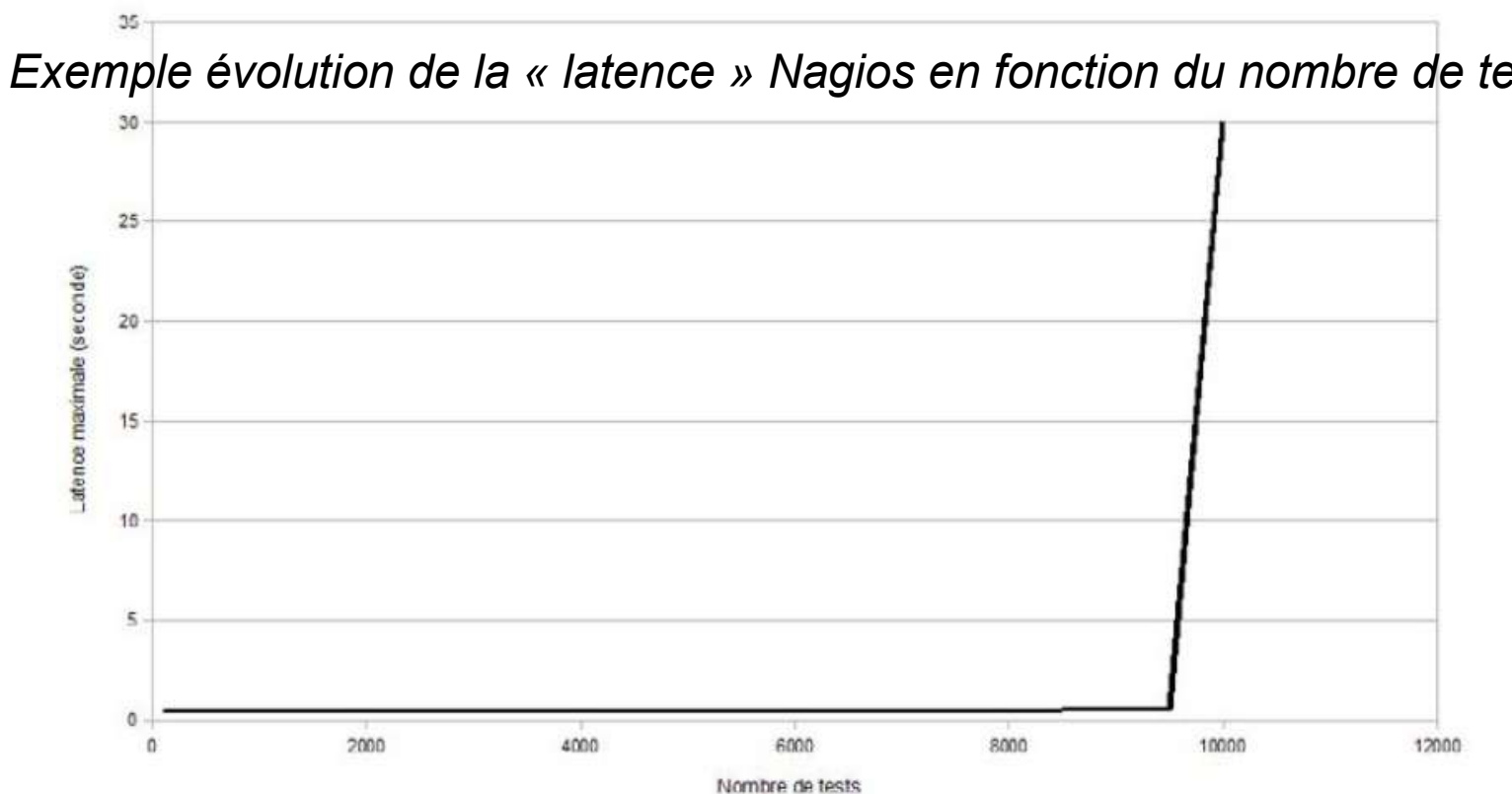
## Cas de relations à parents multiples



## ***Capacité à gérer un parc important de machines***

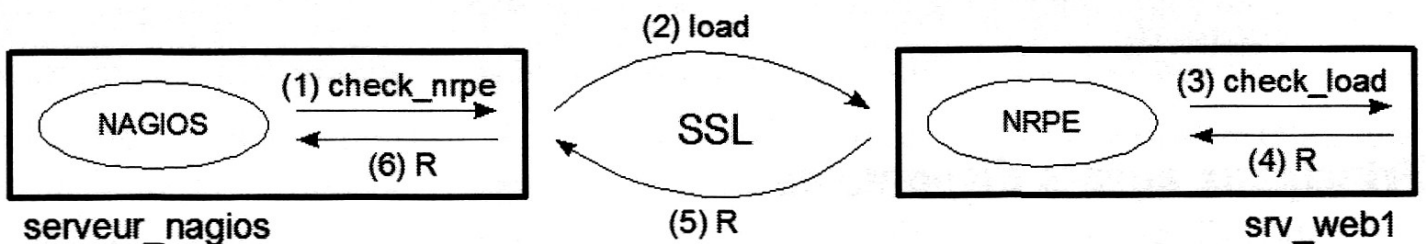
### ➤ Les performances

- Supervision = lancer régulièrement un nombre élevé de vérifications sur des hôtes distants.
- Ordonnancement différent en fonction de l'élément surveillé.
  - Supervision de la charge CPU différente de celle des espaces disques. CPU varie très vite (un serveur qui ne fait pas grand chose à 8H50 peut être surchargé à 9H). La probabilité est beaucoup plus faible de voir un disque ayant 2To d'espace libre se remplir en moins de 5 minutes.
- Lancer un test a un coût. La charge d'un test est très faible, celle de plusieurs milliers élevée.
- Le logiciel doit pouvoir lancer un grand nombre de tests par minute. S'il n'y arrive pas, il va prendre du retard.



# Interrogation des systèmes distants

- Généralement : installation d'un agent sur les serveurs supervisés.
- Cas de Nagios :
  - Agent Nagios sur les hôtes NRPE- Nagios Remote Process Executor.
    - NRPE agent à installer sur les machines distantes. Agent : logiciel qui réceptionne les demandes de Nagios, exécute un script récupère l'information et lui renvoie.
    - NRPE permet d'exécuter à distance des plugins Nagios classiques déposés par l'administrateur.



```
check_nrpe -H srv-web1 -c load
```

```
OK - Charge moyenne: 0.51, 0.28, 0.28|load1=0.510;1.000;2.000;0;
load5=0.280;1.000;2.000;0; load15=0.280;1.000;2.000;0;
```

```
echo $?
0
```

- NRPE : solution la plus souvent utilisée.
- Inconvénient : existe-t-il une version de l'agent pour tous les systèmes du SI ?

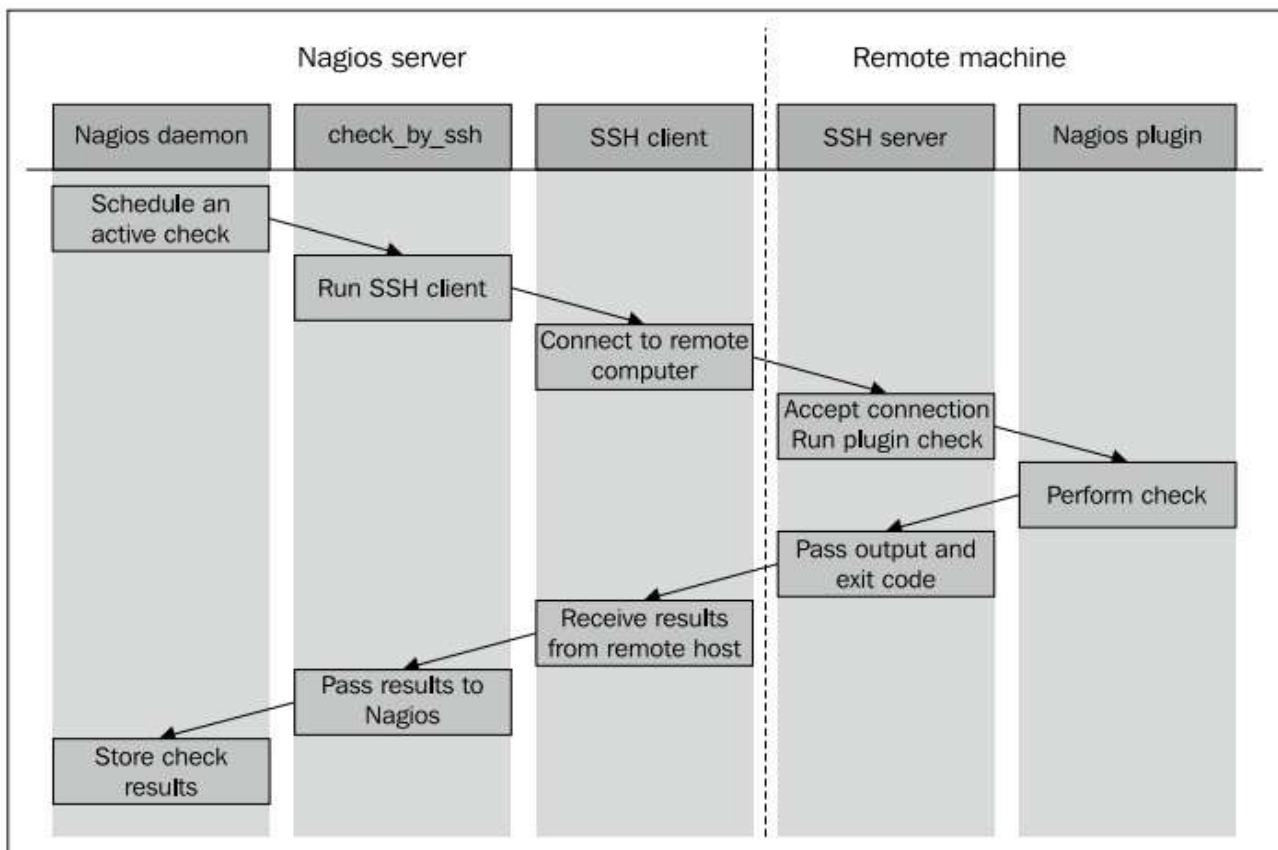
# Interrogation des systèmes distants

## ➤ Interrogation via SNMP (matériel réseau)

- Ex sous Nagios : plugin **check\_snmp** :

```
check_snmp -H <ip_address> -o <OID> [-w warn_range] [-c  
crit_range] [-C community] [-s string] [-r regex] [-R  
regexi] ...
```

## ➤ Interrogation via SSH + Plugin (si pas d'agent pour le système à superviser).



# La métrologie

- Métrologie : science de la mesure au sens le plus large. Mesure : opération qui consiste à donner une valeur à une observation.
- Métrologie réseau : mesure des performances du réseau :
  - Instrumenter le réseau et mesurer des caractéristiques du réseau
  - Analyser les mesures collectées
  - Repérer des comportements normaux/anormaux
- Supervision « classique/historique » (monitoring) informe en temps réel de l'état des équipements, par contre ne permet pas de savoir si le réseau assure parfaitement le service pour lequel il a été conçu.
- Métrologie : notion de qualité de service / Supervision : avant tout disponibilité.
- Monitoring : lancer un nombre élevé de vérifications sur les hôtes distants. La métrologie doit conserver un nombre plus restreints d'éléments mais sur une durée assez longue.