No Files Matching KEY by WinRM

Sommaire

Connaitre les consommation des checks afin de pouvoir dimensionner ses Pollers quand on rajoute des hôtes

Traditionnellement dans un projet de supervision, on commence par mettre en supervision un parc représentatif de l'ensemble de son infrastructure. Une fois cet échantillon représentatif en place, il est utile de savoir quelle va être la consommation CPU une fois que l'ensemble de son infrastructure sera supervisée.

Prenons comme exemple le cas où on a deux clients, partiellement déployés, avec chacun son royaume isolé:

- Royaume **customer-a**, déployé à **10%**
- Royaume **customer-b**, déployé à **20%**

On se pose la question du nombres de CPUs qui seront nécessaires une fois que 100% du parc sera supervisé.

Chaque royaume étant isolé, chacun a ses propres Pollers, et donc sa propre consommation CPU.

> Il est important de noter qu'ici que l'on extrapole en partant du principe qu'on rajoutera le même type d'éléments, dans les mêmes proportions.
>
> Si le déploiement préliminaire était sur un type d'élément (*exemple: windows*) et que la suite est composée d'éléments totalement différents (*exemple: équipement réseaux*), alors l'extrapolation ne sera pas concluante car les checks de ces éléments ne sont pas les mêmes.

Pour cela, on va extraire des informations issues de la commande **shinken-scheduler-export-data** afin d'avoir:

- **le nombre de CPUs utilisés par les checks pour chaque royaume**

Nous allons avoir besoin d'un dump de données avec en option une prévision de la charge sur une période représentative, disons par exemple 1 heure. Il suffit alors de lancer la commande comme ceci:

```
shinken-scheduler-export-data --export-type=sizing-pollers --duration-scheduling-simulation=3600
```

Avec ce lancement les noms (hôte, check, commandes et royaumes) seront présents dans l'export. Il est tout à fait possible de faire l'analyse sur un export en **--anonymous**, des hash des noms des royaumes seront utilisés au lieu des noms finaux.

L'importation du fichier .csv généré est décrit dans FOR SHINKEN TEAM - shinken-scheduler-export-data : export des données du Scheduler

Création du tableau récapitulatif sur la consommation totale des temps CPU par royaume

Consolidation des données: utilisation d'un tableau croisé dynamique

A partir des nombreuses données de checks que nous avons, nous allons devoir procéder à une consolidation afin d'avoir notre résultat facilement exploitable. On va utiliser un tableau croisé dynamique.

Un tableau croisé dynamique dans un tableur est un outil de analyse de données qui vous permet de créer une vue synthétique et facile à lire d'une grande quantité de données (ici nos exécutions de checks).

On y choisi les données à inclure, comment les organiser et comment les synthétiser, filtrer, classer et totaliser les données en fonction de nos besoins.

Création du tableau croisé dynamique

La création du tableau récapitulatif passe par la création d'un Tableau croisé dynamique. Depuis votre Feuille d'importation des données, il faut cliquer sur **Insertion Tableau croisé dynamique**, et valider:

? Unknown Attachment

Obtenir la consommation CPU totale par royaume

Sélection des royaumes en tant que lignes de notre tableau

Arrivé sur la nouvelle feuille, Excel nous propose de sélectionner les lignes de notre nouveau tableau, dans le bloc de droite.

Dans notre cas, ce sera nos royaumes. Il faut donc faire glisser le champ **realm** (ou bien **realm_anonymous_hash** si on a une version anonyme de l'export) vers le bloc "**Lignes**":

? Unknown Attachment

On obtient alors la base de notre tableau, avec des données qui seront désormais organisées par royaume.

Sélection des "Valeurs": le cpu_time, le temps consommé par les checks

A chaque ligne/royaume, il faut lui assigner une (ou plusieurs) "**Valeurs**". Pour cela, on fait glisser le champ "**cpu_time**" vers le bloc "**Valeurs**" afin d'avoir pour chaque royaume son champ **cpu_time** associé.

Par contre, par défaut, Excel prends par défaut le nombre d'occurrences du champ **cpu_time** comme "**Valeur**", ce qui n'est pas ce qui est souhaité:

? Unknown Attachment

Avec ce choix par défaut, Excel nous montre que nous avons prévu de lancer 5250 checks dans l'heure sur le royaume customer-a, et 15750 sur le customer-b.

Cependant, ceci ne nous apprends pas quelle est leur consommation CPU réelle, on doit donc passer d'un nombre d'occurrences à une "**Somme**"

Passer du nombres de lignes avec "cpu_time" à une vrai somme des temps CPU

Une modification des "**Paramètres des champs de valeurs**" est nécessaire sur "**Nombre de cpu_time**" :

- il faut changer le type de calcul de "**Nombre**" à "**Somme**"
- puis dans un second temps on en profite pour changer le titre de la colonne en "**Somme de temps CPUS utilisés par royaume**"

? Unknown Attachment

On obtient ainsi pour chaque royaume la somme de temps CPU utilisé pour l'ensemble des checks, sur la période voulue (*ici une heure*).

Passer du nombre total de temps CPU consommé au nombre de CPU nécessaires

Afin d'avoir le nombre de CPU nécessaires, il faut revenir à l'échelle d'une seconde, et donc rajouter une nouvelle colonne avec comme valeur la "**Somme de temps CPUS utilisés par royaume**" divisée par **3600 (** *secondes, soit une heure correspondant à notre extraction* **)**, et ce pour chaque ligne:

? Unknown Attachment

Le total général des CPUs utilisés est utile à titre informatif, mais c'est surtout les valeurs par royaumes qui sont intéressantes.

Tableau final avec le nombres de CPUs utilisés par royaume

Une fois le calcul effectué, voici le tableau final obtenu:

? Unknown Attachment

Il indique le nombre de CPUs utilisés par royaume.

Analyse des résultats

Le tableau final obtenu, on peut procéder à notre analyse.

Sur le tableau, on peut voir clairement notre consommation actuelle de CPUs pour chaque royaume:

- **0,36** CPUs pour le premier royaume (*customer-a*)
- **1,10** CPUs pour le second royaume (*customer-b*)

On était partis d'une hypothèse sur l'état d'avancement du remplissage des royaumes:

- Royaume **customer-a**, déployé à **10%**
- Royaume **customer-b**, déployé à **20%**

On peut alors extrapoler notre consommation finale une fois toute l'infrastructure supervisée.

> Pour rappel, l'extrapolation n'est possible que si on rajoute le même type d'éléments (*dans les mêmes proportions*) dans le futur.

En partant du principe que l'on rajoute le même type d'éléments que ce que nous avons actuellement (*afin d'avoir les mêmes checks, et donc la même consommation CPU*), on arrive sur une consommation finale suivante:

- Royaume **customer-a: 0,36*10 = 3.6 CPUs**
- Royaume **customer-b: 1.10*5 = 5.5 CPUs**

On aura donc besoin au final par royaume d'une allocation de:

- **4** CPUs pour le (ou les) poller(s) du royaume **customer-a**
- **6** CPUs pour le (ou les) poller(s) du royaume **customer-b**