Traçage de série chronologiques de données avec Gnuplot

Gazette Linux n°126 — Mai 2006

Ron Peterson

Article paru dans le n°126 de la Gazette Linux de mai 2006.
Traduction française par Deny `<deny CHEZ monaco POINT net>`**.**
Relecture de la traduction française par Joëlle Cornavin `<jcornavi CHEZ club TIRET internet POINT fr>`**.**

Table des matières

1. Introduction 1
2. Préparation de quelques données 2
3. Traçage des données 2
4. Conclusion 3

1. Introduction

Les bons administrateurs système journalisent énormément. Beaucoup des informations que nous collectons se composent de séries chronologiques de données : un ensemble de valeurs numériques associées à une séquence de valeurs discrètes de temps.

Il existe une grande quantité d'outils pour aider l'administrateur système diligent à surveiller ces données visuellement lorsqu'elles sont collectées. Bon nombre d'entre elles sont construites à l'aide de l'excellent outil RRDTool (http://oss.oetiker.ch/rrdtool/) de Tobias Oetiker. Quelques exemples notables comprennent Cacti, Cricket et Smokeping. Il y en a beaucoup d'autres.

Tout cela convient tant que vous savez ce que vous voulez surveiller. Cependant, il se peut que vous ne souhaitiez parfois effectuer qu'une brève visualisation *ad hoc*. Comme vous pouvez le supposer, la plupart des systèmes Linux offrent une myriade d'outils de visualisation (par exemple, Grace (http://plasma-gate.weizmann.ac.il/Grace/) et Gri (http://gri.sourceforge.net/)). Dans cet article, je vous présenterai Gnuplot (http://www.gnuplot.info/)>, en me concentrant plus particulièrement sur la façon de tracer des séries chronologiques de données.

2. Préparation de quelques données

Gnuplot sans données n'est pas complet. Pour argumenter ou au moins dans le but de donner matière à discuter pour la suite de cet article, j'inclue la ligne suivante dans le fichier crontab de mon système :

Si comme moi, vous avez configuré le fichier `syslog.conf` comme ci-dessous :

...alors vous trouverez tous les messages de type `facility local3` dans leur propre fichier spécial. Du fait que nous demandons au programme logger de tracer toutes nos données de charge moyenne, il sera facile d'extraire ces informations du reste du désordre de notre fichier journal. Un simple **grep CRON-LOADAVG /var/log/cron.log > load.dat.1** devrait convenir. Voici à quoi devrait ressembler le fichier obtenu :

À présent extrayons seulement les données que nous voulons :

La commande de conversion **tr** remplace les espaces multiples par un seul espace, de sorte que nous pouvons nous attendre à un comportement plus cohérent de la commande **cut**. Dans ce cas, la commande de conversion **tr** est superflue, mais je pense néanmoins que c'est une bonne habitude à prendre. Avec un peu de chance, voici à quoi ressemblent nos données :

C'est presque parfait. Malheureusement, comme notre exemple tiré de gnuplot s'attendra à deux colonnes d'entrée délimitées par un espace, nous devons remplacer les espaces délimitant les composants d'horodatage par un autre caractère, comme un trait d'union.

Comme ce n'est pas un article sur Perl, je ne vous ennuierai pas avec les détails sur le rôle de cette commande. Cependant, pour l'intérêt pédagogique, je pense qu'il est utile d'expliquer parfois les rouages. Voici à quoi ressemblent nos données à présent :

3. Traçage des données

En premier lieu, je vous donnerai un aperçu, puis j'expliquerai en détail. Créez un fichier avec le contenu suivant, en excluant les numéros de lignes. Appelez-le `plot-load.conf`. Modifiez la plage de dates dans la ligne 6 pour inclure l'étendue de vos données.

Si vous exécutez la commande suivante, vous devriez obtenir un fichier appelé `load.png`. Utilisez votre afficheur d'images favori pour l'examiner. J'espère que rien de trop alarmant ne s'affiche.

La première ligne de notre commande gnuplot indique de créer un fichier PNG et donne ses dimensions. PNG n'est qu'un format parmi la myriade de formats de sortie possibles. La deuxième ligne indique que notre axe `X` représente des données temporelles. La troisième ligne utilise une spécification de format de date standard (reportez-vous à man date) pour indiquer à quoi ressemblent les données d'horodatage de notre fichier de données. Nous devons employer le même format dans la ligne 6, où nous indiquons l'heure de début et l'heure de fin de notre graphe. Vous pouvez omettre ceci, mais je pense qu'il est utile d'ancrer les points finaux, en particulier en traçant des sources de données multiples dans un seul graphe. La ligne 7 fixe les limites de traçage de notre axe `Y`.

La ligne 13 mérite un peu plus d'attention. Le nom de notre source de données s'affiche en premier lieu. L'option `using 1:2` signifie extraire des donnnées des colonnes 1 et 2 de notre source de données. L'option `index 0` signifie utiliser le premier ensemble de données dans le fichier. Les ensembles de données sont délimitées par des paires d'enregistrements vides. Notre fichier était simple. Il ne comportait que les `col1` et `col2` de l'ensemble de données zéro dans le pseudo fichier de données suivant :

En supposant que nous ayons des ensembles de données multiples dans un seul fichier (peut-être souhaitons-nous comparer les moyenns de charge d'hôtes multiples), un moyen de combiner ces données dans un seul graphe serait de développer la ligne 13 comme suit :

4. Conclusion

Il est utile de savoir comment déployer rapidement quelques traçages de séries chronologiques, mais Gnuplot offre des capacités bien plus étendues que ce à quoi j'ai fait allusion dans cet article. J'espère vous avoir donné envie d'en apprendre encore plus.

Ron Peterson est administrateur de réseau et systèmes au *Mount Holyoke College*, dans les collines riantes de l'ouest du Massachusetts. Il aime enseigner à ses trois petits enfants l'influence néfaste des codecs propriétaires des contenus multimédia pendant qu'ils regardent ensemble les dessins animés de *Homestar Runner*.