

TD n° 12

Optimisation d'un Système Embarqué

Le but de ce TD est de mettre en œuvre quelques techniques pour l'optimisation d'un système. Le système que vous devrez optimiser sera basé sur une machine de type Pentium avec 64Mo de RAM.

Vous ferez un compte-rendu des choix et des actions effectuées qui vous ont conduits à un système plus performant.

1 Optimisation de la configuration du noyau

Pour ce TD, vous utiliserez l'image disque `sde-optim.vmdk` qui continent l'image d'un noyau 2.6.32-68 ainsi que l'image d'une machine virtuelle que nous utiliserons pour nos tests avec `qemu`.

<https://trolen.polytech.unice.fr/cours/sae/td12/sde-optim.7z>

1.1 Mettre en place les mesures sur le noyau

Pour pouvoir mesurer les gains que nous pourrions avoir sur le temps de chargement du noyau, il est nécessaire d'instrumenter celui-ci. Pour être sûr que tout le monde fasse les mesures sur les mêmes bases, vous partirez du fichier de configuration du noyau `config-2.6.32-68.base`.

Vous veillerez à bien faire le ménage complet des sources de votre noyau 2.6.32-68 si vous l'aviez déjà compilé. Pour utiliser cette configuration du noyau, vous copierez ce fichier dans les sources du noyau sous le nom `.config` et vous exécuterez la commande suivante afin de vérifier la validité du fichier de configuration :

```
cp config-2.6.32.68.base .config
make oldconfig
```

1.2 Evaluation du temps de démarrage d'un noyau

Vous commencerez donc par configurer votre noyau 2.6.32-68 afin d'avoir accès aux mesures temporelles lors du chargement de celui-ci. Vous n'oubliez pas de mettre toutes les fonctionnalités activées dans le noyau ou bien vous construirez une image `initrd` contenant les modules nécessaires.

Après compilation et génération du noyau, vous démarrerez un nouveau système avec votre nouveau noyau.

```
qemu -m 64 -kernel arch/x86/boot/bzImage -hda hda-bench.qcow2 -append
"root=/dev/hda1" -device ne2k_pci,netdev=mynet0 -netdev
user,id=mynet0,net=192.168.10.0/24, dhcpstart=192.168.10.10 -redir tcp:5555::22
```

Vous ferez une première sauvegarde des mesures réalisées du démarrage de ce système dans `output1.svg`. Pour cela, vous utiliserez le script `bootchart.pl` se trouvant dans `/work/scripts` (ce script se trouve normalement dans les sources du noyau). Vous veillerez à noter les conditions d'expérimentation, afin de les reproduire pour les futurs tests.

Vous utiliserez la commande `scp` depuis votre machine virtuelle de travail pour vous connecter par `ssh` à votre machine virtuelle `qemu` et récupérer le fichier `svg` via la commande:

```
scp -P 5555 root@localhost:/work/scripts/output1.svg .
```

Ce graphique vous donnera ainsi le temps de démarrage de référence (celui donc à optimiser) pour la configuration du noyau que nous avons. Pour visualisé celui-ci vous pourrez le copier sur votre machine physique via le partage de dossier `VirtualBox (/media/sf...)`.

1.3 Optimisation du temps de chargement du noyau

1.3.1 Optimiser les fonctionnalités les plus gourmandes en temps

Après la visualisation du résultat obtenu dans `output1.svg`, vous devez constater qu'il est important d'optimiser certaines fonctionnalités très gourmandes en tant d'exécution lors de l'initialisation du noyau. Commencez donc par chercher comment optimiser celles-ci.

TD n° 12

Optimisation d'un Système Embarqué

Vous pourrez consulter l'adresse http://elinux.org/Boot_Time pour y trouver les actions à entreprendre. Puis, vous confronterez les informations obtenues avec la documentation du noyau pour vous assurer que vous disposez des informations adaptées à votre version de noyau.

Appliquez un correctif pour gagner sur le temps de boot du noyau et stockez votre résultat dans un fichier `output2.svg`.

1.3.2 Optimisation générales du noyau Linux

Dans un deuxième temps, à l'aide du minimum d'essais, pour éviter de passer votre td à compiler, vous tenterez d'obtenir un gain important quant à la vitesse de démarrage de votre noyau (un minimum de 50%). Vous veillerez tout particulièrement aux options à activer, à désactiver ou bien encore à régler (changement de valeurs) pour optimiser votre système en terme de vitesse de démarrage et si possible en terme d'emprunte mémoire (critère moins prioritaire). Vous lirez la documentation des options afin de comprendre leur impact sur le temps de boot du noyau et son impact mémoire.

Vous pourrez trouver les options utiles dans les sections suivantes :

- General setup
- Processor type and feature

Vous comparerez ce nouveau noyau produit en termes de vitesse de boot à l'aide du script `bootchart.pl`. Vous veillerez bien à ce que les conditions de benchmarking soient identiques d'un test à l'autre afin de ne pas influencer sur les mesures effectuées.

En plus de la bonne configuration du noyau pour optimiser les temps de chargement, vous veillerez à utiliser les bonnes options pour le chargement du noyau pour optimiser au maximum votre temps de démarrage pour votre configuration. Vous stockerez les résultats correspondant à cette configuration dans un fichier `output3.svg`.

Quelles sont les fonctions les plus gourmandes en temps d'exécution ? Comment essayer d'optimiser ces délais ? Faites une conclusion sur les gains que vous avez obtenus.

2 Optimisation de la configuration du système

2.1 Optimisation du temps de démarrage des services

A l'aide de l'application `bootchart` présente sur l'image `hda-bench.qcow2`, vous analyserez le temps nécessaire à l'initialisation des services sur votre système.

Pour réaliser ces mesures, vous ajouterez l'option suivante pour le démarrage de votre noyau : `-append "...init=/sbin/bootchartd"`. Pour visualiser les graphiques, il vous suffit de lancer `bootchart` juste après vous être connectés au système.

Que pouvez-vous conclure à la visualisation des résultats obtenus par `bootchart` ? Est-ce que le système utilise déjà un lancement en parallèle des différents services ? Si c'est le cas, que pouvez proposer pour gagner du temps sur le démarrage du système ?

2.2 Optimisation de la consommation énergétique

Il est aussi possible de configurer votre système pour que celui-ci consomme moins d'énergie. Vous pouvez ainsi autoriser le changement de fréquence du cpu et aussi autoriser les modes de standby ou d'hivernation. Activez ces options dans le noyau.

TD n° 12

Optimisation d'un Système Embarqué

Vous utiliserez `powertop` sur votre système afin de suivre ses recommandations en termes de configuration de votre noyau et de votre système pour le rendre moins énergivore possible. Pour l'utiliser, il faudra activer :

- Kernel Hacking
 - Debug Kernel et Collect Kernel times statistics

Quel problème rencontrez-vous pour continuer à optimiser votre système ?