Linux SMP HOWTO

David Mentré, David.Mentre@irisa.fr v1.9, 13 janvier 2000

Ce HOWTO passe en revue les principaux problèmes et leurs solutions liés la configuration et à l'emploi d'un système SMP sous Linux.

Contents

1 Introduction

Linux fonctionne sur les machines SMP (Symetric Multi-Processors). Le support SMP fut introduit dans la version 2.0 du noyau et a été largement amélioré depuis. La gestion est beaucoup plus fine dans la série 2.2.x que dans la 2.0.x, d'où de meilleures performances lorsque les processus font appel au noyau !

HOWTO maintenu par David Mentré (*David.Mentre@irisa.fr*). La dernière version de ce HOWTO peut être obtenue à

- *http://www.irisa.fr/prive/mentre/smp-howto/* (France)
- *http://www.phy.duke.edu/brahma/smp-faq/* (USA)

Si vous voulez contribuer à ce HOWTO, je préfère les diffs

SGML version `http://www.irisa.fr/prive/mentre/smp-howto/smp-howto.sgml` de ce document, mais toute remarque (en texte pur) sera grandement apprécié. Si vous m'envoyez un e-mail à propos de ce document, insérez s'il vous plaît un tag tel `[Linux SMP HOWTO]` dans le champ `Suject:` de votre e-mail. Ça m'aide à trier automatiquement mon courrier (et vous recevrez une réponse plus rapide `;)`).

Ce HOWTO reprend et enrichit *first draft* `http://www.ihoc.net/linux-smp-faq-draft.html` écrit par **Chris Pirih**.

Toutes les informations contenues dans ce HOWTO sont fournies "tel que". Toute garantie explicite, implicite ou légale, concernant l'exactitude de l'information, de la convenance à quelque usage particulier est par la présente spécifiquement déclinée. Alors que tous les efforts ont été faits pour assurer l'exactitude des informations contenues dans ce HOWTO, les auteurs n'assument aucune responsabilité pour les erreurs, omissions ou dommages résultant de l'utilisation des informations contenues dans ce document.

2 Questions indépendantes de l'architecture.

2.1 Côté noyau

1. **Linux supporte-t-il les threads multiples ? Si je lance deux ou plusieurs processus, seront-ils répartis entre les différents processeurs disponibles ?** Oui. Les processus et les threads du noyau sont répartis entre les processeurs. Ceux de l'espace utilisateur ne le sont pas.

2. **Quelles sont les architectures SMP supportées ?**

   D'après Alan Cox :

   Les versions 2.0 du noyau supportent les systèmes SMP de type hypersparc (SS20, etc...) et Intel 486, Pentium ou machines supérieurs compatible avec la norme Intel MP1.1/1.4. **Richard Jelinek** ajoute : jusqu'à présent, seul des systèmes ne comprenant pas plus de 4 processeurs ont été testés. La spécification MP (et donc Linux) autorise théoriquement jusqu'à 16 processeurs.

   Le support SMP pour les architectures UltraSparc, SparcServer, Alpha et PowerPC est disponible dans le 2.2.x.

   De Ralf Bächle :

   MIPS, m68k et ARM ne gèrent pas le SMP; les deux derniers ne le supporteront probablement jamais.

   Ceci étant, je ferai un hack pour MIPS-SMP dès que j'aurais une machine SMP...

3. **Comment construire un noyau Linux gérant le SMP ?** La plupart des distributions ne fournissent pas un noyau adapté au SMP. Vous devrez donc en faire un vous même. Si vous n'avez encore jamais compilé de noyau, voila une excellente occasion d'apprendre. Expliquer comment compiler un nouveau noyau dépasse le cadre de ce document; préférez-vous au Linux Kernel Howto pour de plus amples informations (**C. Polisher**). Dans la série 2.0 jusqu'à la version 2.1.132 exclue du noyau, décommentez la ligne `SMP=1` dans le Makefile principal (`/usr/src/linux/Makefile`).

   Dans la version 2.2, configurez le noyau et répondez "oui" à la question "Symetric multi-processing support" (**Michael Elizabeth Chastain**).

   Autorisez l'horloge temps réel en cochant l'option "RTC support" (**Robert G. Brown**). Notez qu'inclure le support RTC, en réalité, pour autant que je sache, n'empêche pas le problème connu de la dérive de l'horloge avec le SMP : activer cette fonctionnalité avertit en cas de lecture de l'horloge au démarrage. Une note de **Richard Jelinek** signale aussi qu'activer "Enhandced RTC" est nécessaire pour activer le deuxième processeur (identification) sur certaines cartes mère Intel exotiques.

   Enfin, sur plate-forme Intel, N'ACTIVEZ PAS l'option APM (Advanced Power Management)! APM et SMP sont incompatibles et votre système plantera certainement (ou du moins probablement ;)) au démarrage si APM est activé (**Jakob Oestergaard**). **Alan Cox** le confirme : désactivez APM pour les machines SMP en 2.1.x. En gros le comportement APM est indéfini en présence de systèmes SMP et tout peut arriver. Toujours sur plate-forme Intel, activez "MTRR (Memory Type Range Register) support". Certains BIOS sont défectueux et n'activent pas la mémoire cache du second processeur. Le support MTRR contient le code pour y remédier.

   Vous devez reconstruire votre noyau et vos modules quand vous passez en SMP et quand vous changez de mode SMP. N'oubliez pas d'effectuer un `make modules` et un `make modules_install` (**Alan Cox**).

   Si vous obtenez des erreurs au chargement des modules, vous n'avez probablement pas réinstallé vos modules. Néanmoins, avec quelques noyaux 2.2.x, certains ont rapporté des problèmes lors de la recompilation d'un noyau SMP en noyau UP (Uni-Processeur). Afin de résoudre cela, sauvegardez votre fichier .config, et faites un *make mrproper*, restaurez votre fichier *.config* et recompilez votre noyau (*make dep*, etc...) (**Wade Hampton**). N'oubliez pas de relancer lilo après avoir recopié votre nouveau noyau.

Récapitulation :

```
make config # ou menuconfig ou xconfig
make dep
make clean
make bzImage # ou ce que vous voulez
# copiez l'image du noyau manuellement puis RELANCER LILO
# ou make lilo
make modules
make modules_install
```

4. **Comment compile-t-on un noyau Linux non-SMP ?** Dans la série 2.0, **commentez** la ligne `SMP=1` dans le Makefile principal (/usr/src/linux/Makefile).

   Pour la série 2.2, configurez le noyau et répondez "no" à la question "Symmetric multi-processing support" (**Michael Elizabeth Chastain**).

   Vous devez absolument recompiler votre noyau et ses modules pour activer ou désactiver le mode SMP. N'oubliez pas de faire `make modules`, `make modules_install` et de relancer lilo. Voyez les notes plus haut sur les problèmes de configuration possibles.

5. **Savoir si ça marche**

   ```
   cat /proc/cpuinfo
   ```

   Sortie typique (bi-PentiumII):

```
processor       : 0
cpu             : 686
model           : 3
vendor_id       : GenuineIntel
[...]
bogomips        : 267.06

processor       : 1
cpu             : 686
model           : 3
vendor_id       : GenuineIntel
[...]
bogomips        : 267.06
```

6. **Statut de la migration du noyau vers un verrouillage fin et le multithreading ?** La version 2.2 du noyau possède une gestion des signaux, des interruptions et de quelque E/S à verrouillage fin (fine grain locking). Le reste est en cour de migration. En mode SMP, plusieurs processus peuvent être ordonnancés simultanément.

   Les noyaux 2.3 (futur 2.4) possèdent vraiment des verrous noyaux fins. Dans la série des noyaux 2.3 l'usage des gros blocages noyau a tout simplement disparu. Tous les sous-systèmes majeurs du noyau Linux sont complètement codés avec des threads : réseau, VFS, VM, ES, block/pages de cache, ordonnancement, interruptions, signaux, etc... (**Ingo Molnar**)

7. **Linux SMP supporte-t-il les affinités processeur ?**

Noyaux standard

Oui et non. Il n'est pas possible de forcer l'assignation d'un processus à un processeur spécifique mais l'ordonnanceur Linux possède un parti-pris pour chaque processus qui tend à conserver les processus attachés à un processeur spécifique.

Noyau patché

Oui. Voir *PSET - Processor Sets for the Linux kernel* `http://isunix.it.ilstu.edu/~thockin/pset/` :

> Ce projet a pour but d'offrir une version compatible au niveau sources et fonctionnalités de pset (tel que défini par SGI - partiellement retiré de leur noyau 6.4 IRIX) pour Linux. Cela autorise les utilisateurs à déterminer sur quel processeur ou ensemble de processeurs un processus peut tourner. Les utilisations possibles incluent l'assignement de thread à des processeurs distincts, la synchronisation, la sécurité (un processeur dédié à 'root') et sûrement davantage encore.

Nous nous sommes attachés à concentrer toutes les fonctionnalités autour de l'appel système sysmp(). Cette routine accepte un certain nombre de paramètres qui déterminent la fonctionnalité requise. Ces fonctions comprennent:

- affecter un processus/thread à un processeur spécifique;
- interdire un processeur d'exécuter certains processus;
- empêcher strictement l'utilisation d'un processeur;
- assigner à un processeur un _unique_ processus (et ses fils);
- information sur l'état du processeur;
- créer/supprimer un ensemble de processeurs, sur lesquels les processus peuvent être limités

8. **A qui rapporter les bogues SMP ?** Signalez s'il vous plaît les bogues à `linux-smp@vger.rutgers.edu`.

9. **A propos des performances SMP** Si vous voulez mesurer les performances de votre système SMP, vous pouvez essayer les tests de Cameron MacKinnon, disponibles à *http://www.phy.duke.edu/brahma/benchmarks.smp* .

2.2 Côté utilisateur

1. **Ai-je vraiment besoin de SMP ?**

   Si vous vous le demandez, vous n'en avez probablement pas besoin. :) En général les systèmes multiprocesseurs offrent de meilleurs performances que les systèmes monoprocesseurs, mais pour obtenir un gain quelconque vous devez considérer bien d'autres facteurs que le seul nombre de processeurs. Par exemple, sur un système donné, si le processeur est en général inactif, la plupart du temps à cause d'un disque dur lent, alors le système est bloqué au niveau des entrées/sorties ("input/output bound"); il ne bénéficiera probablement pas de la puissance d'un processeur supplémentaire. Si, d'un autre coté, un système doit exécuter beaucoup de processus simultanément et que l'utilisation processeur est très forte, alors vous êtes susceptible d'améliorer les performances de votre système. Les disques dur SCSI peuvent être très efficaces en utilisation avec plusieurs processeurs. Ils peuvent gérer plusieurs commandes simultanément sans immobiliser le processeur (**C. Polisher**).

2. **Obtient-on les mêmes performances avec un biprocesseur 300 MHz qu'avec un processeur 600 MHz ?** Tout dépend de l'application, mais généralement non. Le SMP implique quelques "frais de gestion" absents d'une machine monoprocesseur. (**Wade Hampton**). :)

3. **Comment afficher les performances de plusieurs processeurs ?** Grâce à **Samuel S. Chessman**, se ici trouvent quelques utilitaires pratiques :

Character based:

http://www.cs.inf.ethz.ch/˜rauch/procps.html

En gros, il s'agit de procps v1.12.2 (top, ps, et. al.) et de quelques patches pour le support SMP.

Pour les 2.2.x, **Gregory R. Warnes** a rendu disponible un patch à *http://queenbee.fhcrc.org/˜warnes/procps*

Graphique:

xosview-1.5.1 supporte le SMP, les noyaux supérieurs au 2.1.85 (inclus) et l'entrée cpuX dans le fichier /proc/stat.

Page d'accueil officielle pour xosview : *http://lore.ece.utexas.edu/˜bgrayson/xosview.html*

Vous ici trouverez une version patchée par **Kumsup Lee** pour les noyaux 2.2.p : *http://www.ima.umn.edu/˜klee/linux/xosview-1.6.1-5a1.tgz*

Les différents patches noyau de Forissier sont disponibles à : *http://www-isia.cma.fr/˜forissie/smp_kernel_patch/*

Néanmoins, vous ne pouvez pas contrôler l'ordonnancement de façon précise avec xosview car ce dernier le perturbe (**H. Peter Anvin**).

4. **Comment autoriser plus d'un processus lors de la compilation du noyau ?** Utiliser :

```
# make [modules|zImage|bzImages] MAKE="make -jX"
où X = nombre maximum de processus.
Notez que ça ne marche pas avec "make dep".
```

Avec un noyau 2.2, référez vous au fichier `/usr/src/linux/Documentation/smp.txt` pour des instructions précises.

Par exemple, comme lancer de multiples compilateurs autorise une machine avec suffisamment de mémoire à utiliser le temps processeur autrement perdu durant les délais causés par les E/S, `make MAKE="make -j 2" -j 2` aide réellement même sur les machines monoprocesseurs. (de **Ralf Bächle**).

5. **Pourquoi le temps donné par la commande `time` est-il erroné ?** (de **Joel Marchand**) Dans la série des 2.0, le résultat de la commande `time` est faux. La somme utilisateur+système est juste *mais* 'l'étendue' entre le temps utilisateur et le temps système est faux.

Plus précisément : "tout le temps passé sur un processeur autre que celui de démarrage est comptabilisé comme temps système. Si vous chronométrez un programme, ajoutez le temps utilisateur et le temps système. Votre mesure sera alors correcte, à ceci près qu'elle inclura aussi le temps système qui restera à décompter" (**Jakob Østergaard**).

Ce bogue est corrigé dans les versions 2.2.

2.3 Programmation SMP

Section par **Jakob Østergaard**.

Cette section a pour but de signaler ce qui fonctionne et ce qui ne fonctionne pas quand il s'agit de programmer des logiciels avec des threads pour Linux SMP.

2.3.1 Méthodes de parallélisation

1. threads POSIX (POSIX Threads)
2. PVM / MPI Message Passing Libraries

3. fork() – Processus multiples

Comme ni fork() ni les processus PVM/MPI ne partagent généralement la mémoire, mais communiquent au moyen d'IPC ou d'une API de messagerie, ils ne seront pas décrits davantage dans cette section. Ils ne sont pas vraiment spécifiques à SMP, puisqu'ils sont tout autant employés - sinon plus - avec des ordinateurs monoprocesseurs et des clusters.

Seuls les threads POSIX fournissent des threads multiples partageant certaines ressources telles la mémoire. Cette propriété des machines SMP autorise plusieurs processeurs à partager leur mémoire. Pour employer deux (ou plus ;)) processeurs avec un système SMP, utilisez une librairie de thread du noyau. Une bonne librairie *LinuxThreads, une librairie de thread écrite par Xavier Leroy* `http://pauillac.inria.fr/~xleroy/linuxthreads/`

est maintenant intégrée avec la glibc2 (aka libc6). Les distributions Linux récentes intègrent toutes cette librairie par défaut. Vous n'avez donc pas à obtenir un paquetage séparé pour utiliser les threads du noyau.

Il existe des mises en oeuvre des threads (et thread POSIX) de niveau application qui ne tirent pas avantage des threads du noyau. Ces paquetages gardent le thread dans un seul processus et, partant, ne profitent pas du SMP. Néanmoins, elles sont bonnes pour beaucoup d'applications et ont tendance à être plus rapides que les threads du noyau sur des systèmes monoprocesseurs.

Le multithreading n'a jamais été vraiment populaire dans le monde UN*X. Pour diverses raisons, les applications exigeant de multiples processus ou threads ont été pour la plupart écrites en utilisant fork(). Donc, avec une approche de type threads, on rencontre des problèmes d'incompatibilités et de non-adaptation aux thread des librairies, compilateurs et débogueurs. GNU/Linux n'y fait pas exception. Espérons que les sections qui suivent apporteront quelques lumières sur ce qui est possible et sur ce qui ne l'est pas.

2.3.2 La librairie C

Les vieilles librairies ne sont pas sûres au niveau des threads. Il est très important que vous utilisiez la GNU libc (**glibc**), aussi connue sous le nom de **libc6**. Vous pouvez évidemment utiliser des versions antérieurs, mais cela vous causera plus de problèmes que mettre à jour votre système. Enfin, probablement :)

Si vous voulez utiliser GDB pour déboguer vos programmes, voyez plus bas.

2.3.3 Langages, compilateurs et débogueurs

Il existe de nombreux langages de programmation disponibles pour GNU/Linux et beaucoup d'entre eux utilisent les threads d'une manière ou d'une autre. Certains langages comme Ada et Java incluent même les threads dans les primitives du langage.

Cette section, pour l'instant, ne décrira que le C et le C++. Si vous avez une expérience de programmation SMP avec d'autre langages, merci de nous en faire part.

Les compilateurs GNU C et C++, tout comme EGCS C et C++, fonctionnent avec le support thread de la librairie C standard (**glibc**). Il y a néanmoins quelques problèmes :

1. Quand vous compilez en C ou C++, incluez **-D_REENTRANT** dans la ligne de commande du compilateur. Il est nécessaire d'activer certaines fonctions de gestion des erreurs telles celles relatives à la variable errno.
2. Quand vous utilisez C++, si deux threads rencontrent des exceptions simultanément, le programme retournera une erreur de segmentation. Le compilateur génère un code d'exception inadapté aux threads. Une manière de contourner le problème consiste à mettre un pthread_mutex_lock(&global_exception_lock) dans le(s) constructeur(s) de chaque classe que vous

throw() et à insérer le pthread_mutex_unlock(...) correspondant dans le destructeur. Ce n'est pas très beau mais ça marche. Cette solution a été fournie par **Markus Ferch**.

Le débogueur GNU **GDB**, à partir de la version 4.18, devrait prendre en charge les threads correctement. La plupart des distributions Linux comprennent une version patchée de gdb qui gère les threads.

Il n'est pas nécessaire de patcher la **glibc** pour qu'elle fonctionne avec des threads. Si vous n'avez pas besoin de déboguer le logiciel (cela peut-être vrai pour toutes les machines qui ne sont pas dédiées au développement), il n'y a pas besoin de patcher la **glibc**.

Notez que les core-dumps ne sont d'aucune utilité quand vous utilisez des threads. D'une manière ou d'une autre, le core dump est attaché au thread courant et non au programme tout entier. Aussi, pour déboguer quoi que ce soit, faites le depuis le débogueur.

Astuce : si vous avez un thread qui perd la tête, se met à utiliser 100% du temps CPU et que vous ne voyez pas pourquoi, voici une méthode élégante de trouver ce qui se passe : lancez le programme depuis la ligne de commande, sans GDB. Faites dérailler votre thread. Utilisez **top** pour obtenir le PID du processus. Lancez GDB tel que **gdb program pid**. GDB s'attachera lui-même au processus dont vous avez spécifié le PID et arrêtera le thread. Vous disposez maintenant d'une session GDB avec le thread incriminé et vous pouvez utiliser **bt** ou d'autres commandes pour suivre ce qui se passe.

2.3.4 Autres librairies

ElectricFence : cette librairie n'est pas sûre du point de vue SMP. Il devrait néanmoins être possible de la faire fonctionner dans un environnement threadé en insérant des verrous dans son code source.

2.3.5 Autres points concernant la programmation SMP

1. **Où puis-je trouver plus d'informations sur la programmation parallèle ?** Voyez *Linux Parallel Processing HOWTO* `http://yara.ecn.purdue.edu/~pplinux/PPHOWTO/pphowto.html`

   Beaucoup d'informations utiles se trouvent sur *Parallel Processing using Linux* `http://yara.ecn.purdue.edu/~pplinux/`

   Voyez aussi *Linux Threads FAQ* `http://linas.org/linux/threads-faq.html`

2. **Existe-t-il des programmes ou des librairies utilisant les threads ?** Oui. Pour les programmes vous devriez regarder à

   Multithreaded programs on linux `http://www.informatik.uni-bremen.de/~hollow/mthread.html` (j'adore les liens hypertextes, le saviez vous ? ;))

   En ce qui concerne les librairies :

   OpenGL Mesa library

   Grâce à **David Buccarelli**, **andreas Schiffler** et **Emil Briggs**, il existe une version multithread (à l'heure actuelle [1998-05-11], une version fonctionne et permet d'obtenir un accroissement de 5-30% avec certaines suites de test OpenGL). La partie multithread est maintenant incluse dans la distribution Mesa officielle comme une option expérimentale. Pour plus d'information, voyez *Mesa library* `http://www.ssec.wisc.edu/~brianp/Mesa.html`

   BLAS

   BLAS et FFTs optimisés Pentium pro pour Intel Linux `http://www.cs.utk.edu/~ghenry/distrib/`
   Les routines multithread BLAS ne sont pas disponibles pour l'instant, mais une librairie multithread est prévue pour 1998-05-27, voir *Blas News* `http://www.cs.utk.edu/~ghenry/distrib/blasnews` pour plus de détails.

The GIMP

Emil Briggs, la même personne qui est impliquée dans la version multithread de MESA, est aussi en train de travailler sur la version multithread des plugins de The Gimp. Voyez *http://nemo.physics.ncsu.edu/˜briggs/gimp/index.html* pour plus d'info.

3 Questions spécifiques à l'architecture x86

3.1 Pourquoi cela ne marche-t-il pas avec ma machine ?

1. **Puis-je utiliser le mode SMP avec un CPU Cyrix/AMD/non-Intel ? Réponse courte:** non.

   Réponse longue Intel révendique la propriété sur les plan APIC SMP, et tant qu'une compagnie ne prend pas de licence d'Intel pour cela, ils ne peuvent pas l'utiliser. Aucune compagnie ne l'a fait pour l'instant. Cela peut évidement changer dans le futur. A titre anecdotique, Cyrix et AMD adhèrent au standard non-propriétaire OpenPIC SMP mais actuellement il n'existe pas de carte mère l'utilisant.

2. **Pourquoi mon vieux Compaq ne fonctionne-t-il pas ?** Mettez le en mode compatibilité MP1.1/1.4.

   Vérifiez "Configure Hardware" -> "View / Edit details" -> "Advanced mode" (F7 je pense) pour les options de configuration "APIC mode" et cochez "full Table mode". Il s'agit d'une recommandation officielle de Compaq (**Daniel Roesen**).

   Adrian Portelli :

   (a) Pressez F10 quand le serveur démarre afin d'entrer dans l'utilitaire de configuration système (System Configuration Utility)

   (b) Pressez Entrée pour effacer l'écran de démarrage

   (c) Pressez immédiatement CTRL+A

   (d) Un message apparaîtra vous informant que vous êtes maintenant en "Advanced Mode"

   (e) Sélectionnez ensuite "Configure Hardware" -> "View / Edit details"

   (f) Vous verrez alors les réglages avancés (mélangés avec les réglages ordinaires)

   (g) Descendez jusqu'au "APIC Mode" et sélectionnez alors "Fully Mapped"

   (h) Sauvegardez les changements et redémarrez

3. **Pourquoi mon ALR ne fonctionne-t-il pas ?** De **Robert Hyatt**: ALR Revolution quad-6 semble à peu près sûre, alors que quelques machines Revolution quad plus vieilles sans processeurs P6 ne semble pas "fiables"...

4. **Pourquoi ma machine SMP est-elle si lente ?** ou **Pourquoi un processeur montre-t-il une valeur bogomips basse et pas l'autre ?** De **Alan Cox**: si un de vos processeurs rapporte une valeur bogomips très basse, son cache n'est pas activé. Votre vendeur vous à probablement fournis un BIOS bogué. Obtenez un patch pour contourner cela ou mieux retournez la à votre vendeur et achetez une carte mère chez un fournisseur compétent.

   Un noyau 2.0 (> 2.0.36) contient un patch MTRR qui devrait résoudre ce problème (sélectionnez l'option "handle buggy SMP BIOSes with bad MTRR setup" dans le menu "General setup").

   Je pense que les BIOS SMP bogués sont pris en charge automatiquement dans les derniers noyaux 2.2.

5. **J'ai entendu dire que des machines IBM avaient des problèmes**

   Certaines machines IBM possèdent le bloc BIOS MP1.4 dans l'EBDA. C'est autorisé mais pas supporté en dessous des noyaux 2.2.

Il y a une vieille machine IBM SMP basée sur des 486SLC. Linux/SMP requiert un support FPU matériel.

6. **Les spécification MP 1.4 présentent-elles un quelconque avantage vis-à-vis des spécifications 1.1 ?** Non (selon Alan `:)`), 1.4 est juste une spécification plus stricte de 1.1.

7. **Pourquoi l'horloge dérive-t-elle si rapidement quand la machine fonctionne en mode SMP ?**

   Il s'agit d'un problème connu avec la gestion des IRQ et les blocages noyau longs dans la série 2.0 des noyaux. Pensez à mettre à jour votre système vers un 2.2 plus récent.

   De **Jakob Oestergaard**: ou pensez à utiliser xntpd. Cela devrait garder votre horloge à l'heure. Je pense avoir entendu qu'activer RTC dans le noyau corrigeait aussi le problème de dérive de l'horloge. Ça a marché pour moi, mais j'ignore si cela est général ou si j'ai juste été chanceux !

   Certaines corrections du noyau dans les derniers 2.2.x devraient résoudre ce problème.

8. **Pourquoi mes processeurs sont-ils numérotés 0 et 2 au lieu de 0 et 1 (ou autre numérotation bizarre) ?** Le numéro du processeur est fixé par le fabricant de la carte mère et ne veut absolument rien dire. Ignorez le.

9. **Mon système quadruple Xeon plante dès qu'il a décompressé le noyau** (**Doug Ledford**) Essayez de recompiler LILO avec le support LARGE_EBDA et faites attention à bien toujours utiliser bzImage quand vous compilez le noyau. Cela semble avoir résolu le problème de plantage au démarrage ici sur une carte mère Intel multi-Xeon. Notez cependant que cela semble aussi affecter LILO en ceci que l'option root= ne fonctionne plus. Faites donc bien attention d'avoir appliqué 'rdev' à votre noyau au moment où vous lancerez LILO afin d'être sur que votre noyau charge correctement le système de fichier racine au démarrage.

   (**Robert M. Hyatt**) Avec 3 processeurs, avez-vous un terminateur dans le 4ème emplacement ?

10. **Durant le démarrage la machine plante en signalant un problème IOAPIC** Essayez l'option de démarrage "noapic" (**John Aldrich**) et/ou "reboot=bios" (**Terry Shull**).

11. **Mon système se bloque lors de trafic NFS intense** Essayez le dernier noyau 2.2.x et le patch knfsd. Cela est en cours d'investigation. (**Wade Hampton**)

12. **Mon système bloque sans message oops** Si vous utilisez les noyaux 2.2.11 ou 2.2.12, récupérez le dernier noyau. Par exemple 2.2.13 possède de nombreuses corrections SMP. Plusieurs personnes ont rapporté ces noyaux comme instables pour le SMP. Ces mêmes noyaux peuvent avoir des problèmes NFS qui provoqueraient des blocages. Aussi, utilisez une console série pour capturer vos messages oops. (**Wade Hampton**)

    Si le problème persiste (et que les suggestions sur cette liste n'ont pas aidé davantage), vous devriez alors essayer les derniers noyaux 2.3. Ils ont un code SMP/APIC plus bavard (et plus robuste) et un code de prévention contre les blocages durs qui produit des oops plus significatifs au lieu de planter en silence (**Ingo Molnar**).

    (**Osamu Aoki**) Vous DEVEZ aussi *désactiver* toutes les fonctionnalités du BIOS liées à l'économie d'énergie. Exemple d'une bonne configuration (Dual Celeron 466 Abit BP6) :

```
POWER MANAGEMENT SETUP.
  ACPI:              Disabled
  POWER MANAGEMENT:  Disabled
  PM CONTROL by APM: No
```

Si les fonctions d'économie d'énergie sont activées, des plantages aléatoires peuvent se produire

13. **Déboguer des blocages** (item par **Wade Hampton**)

    Un bon moyen de déboguer les blocages consiste à se procurer le patch ikd de Andrea Arcangeli: *ftp://ftp.suse.com/pub/people/andrea/kernel-patches*

    Il y a plusieurs options de débogage. N'utilisez PAS l'option de blocage logicielle ! Pour des machines SMP récentes, activez l'option kernel debugging et ensuite l'option NMI oopser. Afin de vérifier que le NMI oopser fonctionne, après avoir démarré avec votre nouveau noyau, exécutez un `/cat /proc/interrupts` et vérifiez que vous obtenez des NMI. Quand la machine se bloque, vous devriez obtenir un oops.

    Vous pouvez aussi essayer l'option %eip. Elle autorise le noyau à écrire sur la console l'adresse %eip à chaque fois qu'une fonction du noyau est appelée. Quand la machine se bloque, écrivez sur un papier la première colonne ordonnée selon la seconde colonne et cherchez ensuite les adresses dans le fichier System.map. Ca ne marche qu'en mode console.

    Notez que l'utilisation d'une console série facilite grandement le débogage des blocages noyau, qu'ils soient SMP ou non !

14. **Messages "APIC error interrupt on CPU#n, should never happen" dans les logs** Un message comme:

```
APIC error interrupt on CPU#0, should never happen.
... APIC ESR0: 00000002
... APIC ESR1: 00000000
```

    indique la réception d'une erreur de calcul de code d'intégrité. Linux ne peut en être responsable car la partie calcul des messages APIC est complètement matérielle. Il peut s'agir d'un problème matériel marginal. Tant que vous ne percevez pas d'instabilité, ils ne sont pas problématiques. Les messages APIC sont renvoyés jusqu'à ce qu'il soient délivrés (**Ingo Molnar**).

3.2 Causes possibles de plantages

Dans cette section vous trouverez quelques information sur les causes **possibles** de plantage sur une machine SMP (merci à **Jakob Østergaard** pour cette partie). Autant que je sache (David), les problèmes évoqués ici sont spécifiques aux plate-formes Intel.

- **Problèmes de refroidissement** De **Ralf Bächle**: (concernant la taille des boîtiers et les ventilateurs) il est important que l'air circule. Bien sûr, ce n'est pas possible quand toutes sortes d'obstacles, tels des câbles, l'en empêchent dans des boîtiers par trop exigus. D'un autre côté, j'ai vu des boîtiers surdimensionnés provoquer de gros problèmes. Il existe des boîtiers au format tour sur le marché qui s'avèrent actuellement pire à rafraîchir que des boîtiers au format bureau. En bref, la meilleure chose à faire est de penser à l'aérodynamique dans le boîtier. Des boîtiers supplémentaires pour les périphériques dégageant de la chaleur sont également utiles.

  Bien sûr vous pouvez toujours aller chez Radio Shack (ou similaire) et acheter un ventilateur. Vous pouvez utiliser lm_sensor pour surveiller la température des processeurs PII et PIII. Cela peut vous aider à déterminer si la chaleur est un problème ou non (**Wade Hampton**).

- **Mauvaise barrette de mémoire** N'achetez pas de la RAM bon marché et ne mélangez pas des barrettes différente sur une même carte mère.

  Les cartes mères Tyan sont tout particulièrement connues pour leur susceptibilité sur la vitesse de la RAM (voir le paragraphe ci-dessous sur Tyan pour une solution éventuelle).

  Il y a eu des rapports sur des mémoire RAM PC 100 à 10ns vendues avec des cartes mères dont le processeur avait vraiment besoin de RAM à 8ns (**Wade Hampton**).

- **Mauvaise combinaison de processeurs de fréquences différentes** Vérifiez `/proc/cpuinfo` pour voir si vos processeurs fonctionnent à la même cadence.
- **Si votre système est instable, SURTOUT ne l'overclockez pas !** D'ailleurs, même s'il est stable, ne le surcadencez pas.

  De **Ralf Bächle**: le surcadencement pose des problèmes très subtils. J'ai un bel exemple: une de mes vieilles machines surcadencées commet des erreurs de calcul pour quelques pixels d'une fractale de 640 X 400. Le problème est seulement visible quand on les compare en utilisant des outils. Le mieux est donc de ne *jamais, never, nuncas, niemals* surcadencer.
- **Noyaux 2.0.x et Ethernet rapide** (de **Robert G. Brown**) Les noyaux 2.0.x sur des systèmes Ethernet rapide et hautes performances ont des problèmes significatifs (et connus) avec les conditions de course/inter-blocage (race/deadlock) dans la prise en charge des interruptions réseau.

  La solution consiste à obtenir la dernière version des pilotes 100BT en cours de développement à *CESDIS Linux Ethernet device drivers site* `http://cesdis.gsfc.nasa.gov/linux/drivers/` (ceux qui sont au point définissent SMPCHECK).
- **Un bogue dans le chipset 440FX** (de **Emil Briggs**) Si votre système utilise le chipset 440FX alors les problèmes de blocage sont peut-être dûs à une erreur (documentée) du chipset. En voici la référence:

  Intel 440FX PCIset 82441FX (PMC) et 82442FX (DBX) Specification Update. pg. 13

  http://www.intel.com/design/pcisets/specupdt/297654.htm

  Le problème peut se résoudre avec un contournement par le BIOS (ou un patch du noyau). David Wragg a écrit un patch qui est inclus dans le patch MTRR de Richard Gooch's. Pour plus d'informations ainsi qu'un descriptif de solution, voyez ici:

  http://nemo.physics.ncsu.edu/˜briggs/vfix.html
- **NE PAS lancer emm386.exe avant de démarrer Linux SMP** De **Mark Duguid**, Règle implicite #1 avec une carte mère W6LI. ;)
- **Si la machine redémarre/gèle au bout d'un moment, il peut y avoir deux bonne raisons liées à la mémoire et au BIOS** (**Jakob Østergaard**)
  - Si le BIOS est muni de réglages comme "memory hole at 16M" et/ou "OS/2 memory > 64MB", essayez de les désactiver tous les deux. Linux ne réagit pas toujours très bien à ces deux options.
  - Si vous avez plus de 64 MB de mémoire dans votre machine, et que vous spécifiez manuellement le chiffre exact dans la configuration de LILO, vous devriez spécifier 1 MB de moins que ce vous avez réellement dans votre machine. Si vous avez 128 MB, votre ligne dans votre lilo.conf ressemble à: append="mem=127M"
- **Soyez avertis des problèmes concernant les IRQ** Parfois, certaines cartes ne sont pas reconnues ou peuvent déclencher des conflits d'IRQ. Essayez de mettre les cartes sur des slots différents et si possible de les assigner à des IRQ différentes.

  Contribution de **hASCII** : enlever la ligne "append="hisax=9,2,3" dans lilo.conf autorisant à utiliser un noyau de la série 2.1.xx avec le support ISDN + Hisax activé. Les noyaux de la série 2.0.xx ne posent pas ce genre de problème.

  Essayez aussi de configurer les option de configuration du BIOS comme "MP 1.4 mode" ou "route PCI interrupts through IOAPIC" ou "OS Type" configuré ni pour DOS ni pour Novell (**Ingo Molnar**).
- **Utilisation simultanée du lecteur de disquettes de la sortie son** Si vous bloquez alors que vous essayez d'accéder au lecteur de disquettes (par exemple pendant que du son est joué) vous devriez peut-être éditer le fichier drivers/pci/quirks.c et positionner `/int isa_dma_bridge_buggy = 1;`. Le problème se manifeste avec mon Dell WS400 dual PII/300, 2.2.x, SMP (**Wade Hampton**).

3.3 Informations spécifiques aux cartes mères

Notez que des informations plus précises peuvent être trouvées avec la liste des *Cartes mère supposées fonctionner sous Linux SMP* `http://www.nlug.org/smp/`

3.3.1 Cartes mères avec des problèmes connus

- Aucune pour l'instant

3.4 Machine SMP Linux à bas prix (machine double Celeron)

(**Stéphane Écolivet**)

Les machines SMP Linux les moins chères avec des processeurs disponibles de nos jours sont les systèmes double Celeron. Un tel système n'est pas officiellement possible selon Intel. On a intérêt à vérifier qu'il s'agit bien de Celerons de seconde génération, ceux avec 128 Kb de cache L2.

3.4.1 Est-il possible de faire fonctionner une machine double Celeron ?

Réponse officielle d'Intel : non, le Celeron ne peut pas fonctionner en mode SMP.

Réponse pratique : c'est possible, mais cela demande une modification matérielle pour les processeurs Slot 1. La manipulation est décrite par Tomohiro Kawada sur sa page *Dual Celeron System* `http://kikumaru.w-w.ne.jp/pc/celeron/index_e.html` . Naturellement, de telles modifications annulent la garantie... Certaines versions du processeur Celeron sont aussi disponibles au format Socket 370. Dans ce cas, l'altération peut-être faite sur l'adaptateur Socket 370 à Slot 1 qui peut même être vendu pré-cablé pour une utilisation SMP (**Andy Poling**, **Hans - Erik Skyttberg**, **James Beard**).

Il existe aussi une carte mère (ABIT BP6) autorisant l'insertion de deux Celerons dans le format Socket 370 (**Martijn Kruithof**, **Ryan McCue**), l'ABIT Computer BP6 vérifiée, testée et supportée sous linux avec deux ppga socket 370 (**Andre Hedrick**).

3.4.2 Comment Linux se comporte-t-il sur les systèmes double Celeron ?

Bien, merci.

3.4.3 Les processeurs Celeron sont réputés pour être facilement surcadençable. Qu'en est-il des systèmes doubles Celeron ?

Cela **peut** marcher. Néanmoins, surcadencer un tel système n'est pas aussi facile que pour un monoprocesseur. Ce n'est franchement pas une bonne idée pour un système de production. Pour une utilisation personnelle, des systèmes double Celeron 300 A fonctionnant parfaitement à 450 MHz ont été signalés (**de nombreuses personnes**).

3.4.4 Et un système quadruple Celeron ?

C'est impossible. Les processeurs Celerons possèdent à peu près les mêmes fonctionnalités qu'un Pentium II basique. Si vous voulez plus de deux processeur dans votre système, vous devriez regarder du côté des machines à base de Pentium Pro, Pentium II Xeon ou Pentium III (?).

3.4.5 Pourquoi ne pas mélanger Celeron et Pentium II ?

Un système utilisant un Celeron "ré-autorisé" et un Pentium II à la même cadence **peut théoriquement** fonctionner.

Alexandre Charbey à fabriqué un tel système:

- Carte mère Asus P2B-D, proc 1: Celeron 366, proc 2: Pentium II 400@266
- Les fréquences de bus 66Mhz et 75Mhz furent fonctionnelles
- Le processeur le plus rapide (dans ce cas le Celeron) doit être placé sur le deuxième slot. Inverser les processeurs (le plus rapide en premier) conduit rapidement à un échec.

4 Questions spécifiques à l'architecture Sparc

4.1 Quellles sont les machines Sparc supportées ?

Citation de la page web *UltraLinux* `http://ultra.linux.cz/`

(systèmes SMP seulement):

- Workstation UltraSPARC à base de PCI: Ultra60, Ultra450
- Serveurs UltraSPARC à base de SBUS: Enterprise 1, 2, 150
- Serveurs large UltraSPARC à base de SBUS: Enterprise 3000, 4000, 5000, 6000, 10000
- Serveurs UltraSPARC à base de PCI : Enterprise 250, 450
- Machines SPARC sun4m SMP (**Anton Blanchard**)

UltraLinux a fonctionné sur une machine de 14 processeurs (voir la

sortie dmesg `http://lwn.net/1998/1210/a/dm-sparc.html`).

4.2 Problèmes spécifiques au support SMP Sparc

(**David Miller**) Il ne devrait pas y avoir d'inquiétudes.

Le seul problème connu et que nous n'avons pas l'intention de corriger, consiste en ce qu'un noyau SMP compilé pour des systèmes 32bits (ie. non-ultrasparc) ne fonctionnera pas sur les systèmes sun4c.

4.3 Limites SMP spécifiques au noyau courant (2.2)

David Miller: il y a un bug dans le fichier d'en-tête include/linux/tasks.h, cela nécessite de définir NR_CPUS à 64 sur UltraSparc puisqu'il s'agit de la limite supérieure pour le matériel que nous supportons :-)

5 Questions spécifiques à l'architecture PowerPC

5.1 Quellles sont les machines PPC supportées ?

- Cartes mères PowerSurge (incluant UMAX s900)
- PowerMac
- Motorola MTX : support en cours de développement. Les patches ne sont pas encore inclus dans le noyau principal (**Troy Benjegerdes**).

(**Cort Dougan**) Non supporté: Systèmes PPC RS/6000

5.2 Problèmes spécifiques concernant le support SMP PPC

Rien. Compilation SMP normale (voir plus haut). Comme d'habitude, soyez attentif. Les modules sont spécifiques pour UP ou pour SMP. Recompilez les (**Paul Mackerras**).

6 Questions spécifiques à l'architecture Alpha

6.1 Quelles sont les machines Alpha supportées ?

Geerten Kuiper : le SMP marche pour la plupart des serveurs AXP, sinon pour tous.

Jay A Estabrook : le SMP semble fonctionner sur la plupart de nos machines [Compaq] avec deux processeurs ou plus. La liste de celles-ci comprend :

- AS2000/2100 (SABLE)
- AS4000/4100 (RAWHIDE)
- DS20 (DP264)

En sont exclus :

- AS2100A (LYNX)
- TurboLaser bigboys (8200/8400)

6.2 Problèmes spécifiques au support SMP Alpha

Aucun (vraiment ? :-)).

7 Pointeurs utiles

7.1 Divers

- *Parallel Processing en utilisant Linux* `http://yara.ecn.purdue.edu/~pplinux/`
- *Linux Parallel Processing HOWTO* `http://yara.ecn.purdue.edu/~pplinux/PPHOWTO/pphowto.html`

- **(dépassé)** *Page d'acceuille SMP Linux* `http://www.uk.linux.org/SMP/title.html`
- linux-smp mailing list Pour **souscrire**, envoyez `subscribe linux-smp` dans le corps du message à *majordomo@vger.rutgers.edu*

  Pour se **désinscrire**, envoyez `unsubscribe linux-smp` dans le corps du message à *majordomo@vger.rutgers.edu*

  Archives Linux SMP `http://www.linuxhq.com/lnxlists/linux-smp/`

  Archives Linux SMP à progressive-comp.com `http://www.progressive-comp.com/Lists/?l=linux-smp&r=1&w=2#linux-smp`
- *La librairie pthread de Xavier Leroy* `http://pauillac.inria.fr/~xleroy/linuxthreads/`
- *Les Cartes mères qui paraît-il marche avec Linux SMP* `http://www.nlug.org/smp/`
- *procps* `http://www.cs.inf.ethz.ch/~rauch/procps.html`
- *patch pour procps pour 2.2.x* `http://queenbee.fhcrc.org/~warnes/procps`
- *xosview* `http://lore.ece.utexas.edu/~bgrayson/xosview.html`
- *xosview pour 2.2.x* `http://www.ima.umn.edu/~klee/linux/xosview-1.6.1-5a1.tgz`
- *Performance SMP de Linux* `http://www.phy.duke.edu/brahma/benchmarks.smp`
- *CESDIS Linux Ethernet device drivers site* `http://cesdis.gsfc.nasa.gov/linux/drivers/`
- *Systèmes Double Celeron* `http://kikumaru.w-w.ne.jp/pc/celeron/index_e.html`

7.2 Programmes et librairies multithread

- *Linux Threads FAQ* `http://linas.org/linux/threads-faq.html`
- *Programmes multithread sur linux* `http://www.informatik.uni-bremen.de/~hollow/mthread.html`
- *BLAS et FFTs optimisé Pentium pro pour Intel Linux* `http://www.cs.utk.edu/~ghenry/distrib/` (pas disponible tout de suite, mais une librairie double processeurs est prévue pour le 5/27/98. Pour plus de détails, voir *Blas News* `http://www.cs.utk.edu/~ghenry/distrib/blasnews`).
- *Librairie Mesa* `http://www.ssec.wisc.edu/~brianp/Mesa.html` (support multithread expérimental)
- *Plugins parallèles pour The GIMP* `http://nemo.physics.ncsu.edu/~briggs/gimp/index.html`

7.3 Patches spécifiques SMP

- *Patches noyau de Forissier* `http://www-isia.cma.fr/~forissie/smp_kernel_patch/`
- *Patch pour un bug dans le chipset 440FX* `http://nemo.physics.ncsu.edu/~briggs/vfix.html`
- *Patch MTRR (dernière version: 1.9)* `http://www.atnf.csiro.au/~rgooch/kernel-patches.html`
- *PSET - Processor Sets for the Linux kernel* `http://isunix.it.ilstu.edu/~thockin/pset/`
- *Patches SMP de Ingo Molnar* `http://www.redhat.com/~mingo/` (pour les esthètes seulement, s'il vous plaît lisez linux-smp@vger.rutgers.edu)

7.4 Compilateurs parallèliseurs/optimiseurs pour les machines 586/686 (Sumit Roy)

- *Pentium Compiler Group* `http://www.goof.com/pcg/` créateur de pgcc
- *Absoft* `http://www.absoft.com/` compilateur Fortran 90 et Fortran 77
- *The Portland Group, Inc.* `http://www.pgroup.com/` supporte le standard *OpenMP* `http://www.openmp.org` pour la parallèlisation Fortran sur Linux
- *Pacific-Sierra Research Corporation* `http://www.psrv.com/` contient un compilateur gratuit F90 pour Linux et aussi des compilateurs parallèlisant pour SMP Linux
- *Applied Parallel Research* `http://s006.infomall.org/index.html` inclut actuellement des compilateurs parallèlisant pour NT
- *KAI* `http://www.kai.com` offre un compilateur C++ pour Linux qui inclut OpenMPI. Il s'appelle Guide_OpenMP. Information à *http://www.kai.com/parallel/kappro/guide*

  (**Gero Wedemann**).

8 Glossary

- **SMP** Multi-Processeur Symmétrique
- **APIC** Contrôleur d'Interruptions Programmable Avancé
- **thread** Un thread est l'activité processeur dans un processus. Un même processus peut avoir de multiples thread. Ces threads partagent l'espace adresse du processus et peuvent donc par-là partager des données.
- **pthread** Posix thread, threads définie par le standard Posix.
- **APM** Gestion avancée de l'énergie

9 Quoi de neuf ?

v1.9, 13 janvier 2000

- Rappel, désactivation de toutes les fonctions de gestion d'énergie du BIOS (**Osamu Aoki**)
- Explication sur la manière d'accéder au mode de configuration avancé du serveur Compaq (**Adrian Portelli**)

v1.8, 8 novembre 1999

- La carte mère quadruple celeron était un canular, restauration de l'ancien paragraphe (**Simen Timian Thoresen**)

v1.7, 6 novembre 1999

- Nouvelle introduction (**C. Polisher** aka cp)
- De nombreuses corrections typographiques et grammaticales (cp)
- Paragraphe d'introduction sur la compilation du noyau (cp)
- Paragraphe d'introduction sur les besoins SMP (cp)

- Référence sur KAI un compilateur optimisé (**Gero Wedemann**)
- Les cartes mères quadruple celeron existent (**Jeffrey H. Ingber**)

v1.6, 21 octobre 1999

- Ajout d'information sur la perturbation horaire de xosview
- Ajout du message d'information "Erreur d'interruption APIC sur le CPU#n"
- Ajout d'information sur les blocages matériels
- Suppression de la section "Comment obtenir un maximum de performance" (obsolète)
- Ajout d'information sur les systèmes double processeurs avec différents processeurs x86 (un Celeron et un PII)

v1.5, 4 octobre 1999

- Plus de précision dans la description de PSET

v1.4, 30 septembre 1999

- Précision sur l'activation du support MTRR pour les noyaux SMP x86 (moi)

v1.3, 29 septembre 1999

- Beaucoup beaucoup de corrections grammaticales et typographique (**Wade Hampton** aka hww)
- Ajout d'information dans la courte introduction à propos des différences entre 2.2/2.4/2.0 (hww)
- Ajout des choses à faire pas à pas pour recompiler le noyau (hww et moi)
- Ajout d'information concernant les problèmes liés aux modules SMP/UP (hww)
- Ajout de précision dans la section Threads Posix concernant les threads utilisateurs vs. les threads du noyau (hww)
- Nouvel item à propos de NFS et des blocages du noyau (hww)
- Nouvel item à propos des blocage noyau sans message d'alerte (hww)
- Nouvel item à propos du débogage des problèmes de blocage (hww)
- Ajout d'information à propos des problèmes de dégagement de chaleur (hww)
- Divers mise à jour que j'ai oublié (hww)
- Nouvel item à propos des accès disquette et du son (hww)

v1.2, 27 septembre 1999

- Changement de nom: ce document est maintenant un Howto. TWD, et rapide! (**Guylhem Aznar**)

v1.1, 26 septembre 1999

- Ajout d'un lien vers le premier brouillon de la FAQ de Chris Pirish
- Extension des problèmes liés aux IRQ

v1.00, 25 septembre 1999

- Première mise à jour depuis bien longtemps !
- Retraitement de toute la FAQ: le 2.2 est là et le 2.4 arrive
- Ajout des informations sur le verrouillage noyau de Ingo Molnar

- Suppression de l'item "Quelle seront les performance de mes applications sous SMP" : dépassé
- Suppression de l'item "Mon système SMP se verrouille tout le temps." : dépassé
- Suppression de l'item "Vous utilisez le 2.0.35, n'est-ce pas ?" : dépassé
- Suppression de l'item "Certains matériels sont aussi connu pour poser des problèmes." : dépassé
- Effacement de la section "Cartes mère avec des problèmes connus". Nous devrions recommencer du début.
- Suppression de la section "Carte mère sans problèmes connus" : dépassée
- Mise à jour de la section celeron (de nombreuses personnes)
- Ajout de "Les machines SPARC sun4m SMP" dans les machines Sparc supportées (**Anton Blanchard**)
- Ajout de l'item "Durant le démarrage la machine se bloque en signalant un problème IOAPIC" dans la section "Pourquoi cela ne marche-t-il pas sur ma machine ?"
- Ajout de l'item "A propos des performances SMP ?"
- Mise à jour de l'item "Pourquoi mon vieux Compaq ne marche-t-il pas ?"
- Réparation d'un lien dépassé
- Ajout d'un pointeur vers les patches de test SMP d'Ingo

v0.54, 13 mars 1999

- Ajout de la section à propos des systèmes SMP Alpha

v0.53, 08 mars 1999

- Ajout de la section sur les systèmes PowerPC SMP

v0.52, 07 mars 1999

- Ajout de la section sur les systèmes Sparc SMP

v0.51, 06 mars 1999

- Ajout de la section dual-celeron
- Suppression de la section Adaptec
- Mise à jour du lien procps
- Mise à jour du lien xosview
- Ajout d'une réponse pour le plantage du quadri Xeon
- Mise à jour de l'item à propos du patch de la glibc pour gd : devrait être inclus dans la RH 5.2

v0.50, 03 février 1999

- Mise à jour du lien "Programmes Multithread sous linux"

v0.49, 13 janvier 1999

- Mise à jour à propos de CONFIG_SMP. Ajout du .txt dans Documentation/smp. (**Michael Elizabeth Chastain**)

v0.48, 10 décembre 1998

- Fautes d'orthographes corrigée. Adresses email corrigée.

v0.47, 20 novembre 1998

- Ajout de la mention du patch MTRR est inclus 2.0.36 (lié à des problème de BogoMips)

v0.46, 10 novembre 1998

- Mise à jour à propos des cartes mère Epox KP6-LS

v0.45, 25 octobre 1998

- Correction d'une erreur concernant le fichier /proc/stat
- Ajout d'un pointeur vers le site CESDIS Ethernet Linux Drivers

v0.44, 14 octobre 1998

- Mise à jour du lien vers la page web : *Cartes mère supposées fonctionner sous Linux SMP*
- Ajout de l'explication de Jakob : comment chronométrer un système SMP avec les noyaux 2.0

v0.43, 9 septembre 1998

- Mise à jour de la première question dans la section 3.1
- Mise à jour du lien mt-Mesa : multithread est maintenant inclus comme expérimental dans la distribution Mesa

v0.42, 2 septembre 1998

- Mise à jour cosmétique dans la section 3.3
- Deux liens sont marquer comme obsolètes (Multithreaded Mesa et performance SMP)
- Mise à jour de l'item à propos des threads et des exceptions en C++ (sect 3.3)

v0.41, 1 septembre 1998

- Ajout d'une section majeur: "3.3 Programmation SMP" écrite par Jakob Østergaard
- Déplacement de la section "3.2 Coté utilisateur" vers la section 3.3

v0.40, 27 août 1998

- Mise à jour: section 3.1, item 7: processor affinity

v0.39, 27 août 1998

- Mise à jour nécessaire du BOIS Award pour les cartes mères Tyan (**hASCII**)
- Ajout d'un item sur les IRQ dans la section plantage (moi et **hASCII**)
- Ajout du bon support de l'Asus P2B-DS (**Ulf Rompe**)
- Ajout d'une autre archive smp-list dans la section pointeur (**Hank Leininger**)

v0.38, 8 août 1998

- Ajout d'un pointeur vers la FAQ Linux Threads

v0.37, 30 Juillet 1998

- **Emil Briggs** est en train de travailler sur des plugins parallèles pour Gimp (voir "Existe-t-il des programmes ou des library utilisant les threads ?", section "Coté utilisateur")

v0.36, 26 Juillet 1998

- Merci à **Jakob Østergaard**, deux changement dans "Possible causes of Crash"
  - Changé le 2.0.33 pour le 2.0.35 (dernier noyau stable)
  - Ajout de la section "Les plantages liés au BIOS"

v0.35, 14 Juillet 1998

- Ajout des N440BX Server Board dans carte-mère-sans-aucun-problème
- Ajout d'une success story pour la carte mère GigaByte avec une mise à jour du BIOS
- Ajout de la section "Comment obtenir les performances maximum ?" (attend vos suggestions ;)

v0.34, 10 juin 1998

- Ajout de la section "Parallelizing/Optimizing Compilers for 586/686 i machines" dans la section "Useful Pointers", merci à **Sumit Roy**
- Correction, "Asus P/I-UP5" est en fait "Asus P/I-P65UP5"

v0.33, 3 juin 1998

- Encore une success story avec une carte mère GigaByte DLX.
- Une astuce pour les cartes mère Tyan, désactiver l'option "DRAM Fast Leadoff" du BIOS

v0.32, 27 mai 1998

- Asus P/I-UP5 ajouter à la section carte-mère-sans-aucun-problème

v0.31, 18 mai 1998

- Elitegroup P6LX2-A marche avec le 2.1.100 et le 101
- Les bugs doivent être rapportés à`linux-smp@vger.rutgers.edu`

v0.30, 12 mai 1998

- SuperMicro est maintenant une carte mère dans la section carte-mère-sans-aucun-problème

v0.29, 11 mai 1998

- La success story d'une carte mère GigaByte 686 avec le 2.1.101
- Ajout d'un nouvel item dans la section "Coté utilisateur" : "Existe-t-il des programmes ou des library utilisant les threads ?"
- La library OpenGL Mesa library est en train de passer au multithread. Cool! Voir la nouvelle section pour plus de détails.

v0.28, 09 mai 1998

- Un miroir US de cette FAQ est maintenant disponible (voir Introduction)
- Fusion de deux entrées confuses, Gigabyte 686

v0.27, 05 mai 1998

- Nouvelles informations pour les pilotes Adaptec et TekRam
- Les cartes mères Micronics W6-LI marche avec SMP

10 Liste des contributeurs

Un grand merci à ceux qui m'ont aidé à maintenir ce HOWTO:

1. Tigran A. Aivazian
2. John Aldrich
3. Niels Ammerlaan
4. H. Peter Anvin
5. Osamu Aoki
6. Guylhem Aznar
7. Ralf Bächle
8. James Beard
9. Troy Benjegerdes
10. Anton Blanchard
11. Emil Briggs
12. Robert G. Brown
13. Alexandre Charbey
14. Michael Elizabeth Chastain
15. Samuel S. Chessman
16. Alan Cox
17. Andrew Crane
18. Cort Dougan
19. Mark Duguid
20. Stéphane Écolivet
21. Jocelyne Erhel
22. Jay A Estabrook
23. Byron Faber
24. Mark Garlanger
25. hASCII
26. Wade Hampton
27. Andre Hedrick
28. Claus-Justus Heine
29. Benedikt Heinen
30. Florian Hinzmann

31. Moni Hollmann
32. Robert M. Hyatt
33. Jeffrey H. Ingber
34. Richard Jelinek
35. Tony Kocurko
36. Geerten Kuiper
37. Martijn Kruithof
38. Doug Ledford
39. Kumsup Lee
40. Hank Leininger
41. Ryan McCue
42. Paul Mackerras
43. Cameron MacKinnon
44. Joel Marchand
45. David Maslen
46. Chris Mauritz
47. Jean-Francois Micouleau
48. David Miller
49. Ingo Molnar
50. Ulf Nielsen
51. Jakob Oestergaard
52. C Polisher
53. Adrian Portelli
54. Matt Ranney
55. Daniel Roesen
56. Ulf Rompe
57. Jean-Michel Rouet
58. Volker Reichelt
59. Sean Reifschneider
60. Sumit Roy
61. Thomas Schenk
62. Terry Shull

63. Chris K. Skinner
64. Hans - Erik Skyttberg
65. Szakacsits Szabolcs
66. Jukka Tainio
67. Simen Timian Thoresen
68. El Warren
69. Gregory R. Warnes
70. Gero Wedemann
71. Christopher Allen Wing
72. Leonard N. Zubkoff