Mainframes IBM et Haute Disponibilité

EPITA - SRS 2005

Thomas Joubert
Olivier Patole
Nicolas Vigier

Etudes techniques de la haute disponibilité des mainframes eservers.

Novembre 2004

Sommaire

0.1 Introduction

La societe IBM fut fondee le 15 juin 1911 a New York sous le nom Computing Tabulating Recording Co (CTR), fusion des societes Tabulating Machine, Computing Scale Corporation et International Time Recording Compagny. CTR fut ensuite rebaptisee International Business Machines en 1924. Societe a qui on doit de nombreuses innovations dans le monde informatique, par exemple SQL ou encore la norme PC-AT etc...

Il y a 40 ans, cette firme decide d'innover une fois de plus ; proposer des solutions aux petites comme aux grandes entreprises avec comme caracteristiques :

- Un meme OS pour differentes machines, permettant ainsi une portabilite des applications
- Integration directe des bases de donnees dans le systeme
- Une tolerance aux pannes avec garantie de la securite et de l integrite des donnees.

Sont nees les mainframes. Les mainframes sont des machines tres puissantes, qui sont mises en places pour des applications specialises et critiques (banque, assurances etc...). Sur ces mainframes tournent des systemes d'exploitation tels que Unix, linux, z/OS, OS/390, MVS ou encore VM.

Les quatres fonctionnalites importantes d'un mainframe sont :

1. la performance en mono-thread (ideal pour les bases de donnees)
2. grandes preformances en entrees/sorties sur les disques
3. bande passante efficace : pas de goulet d'etranglement
4. hardware redondant : pas ou tres peu de pannes materielles

On se propose aujourd hui d'etudier les solutions IBM et de montrer quelles sont fondamentalement orientees haut disponibilite.

Pour cela, il conviendra en un premier temps de montrer en quoi le materiel des mainframes garantit la haute disponibilite de ces super-ordinateurs. Ensuite on presentera les differents systemes pouvant etre executer sur ce materiel en mettant en avant leurs caracteristiques permettant la haute disponibilite et enfin on etudiera le fonctionnement de z/VM et sont principe de virtualisation qui permet une grande flexibilite et une forte redondance.

Chapitre 1

Le Materiel

Il y a 40 ans, IBM sortait son mainframe S/360. Depuis les mainframes IBM zSeries (dont le plus récent à l'heure actuelle est le zSerie 990) ont continué sur le meme chemin, et fournissent des serveurs extremement performants pouvant répondre à un besoin critique ayant de grosses contraines de disponibilité.

L'IBM zSerie 990 est permet de faire tourner plusieurs OS tels que z/OS ou Linux, ou plusieurs OS en parallel à l'aide de z/VM (voir chapitres suivants).

Afin de garantir une haute disponibilité ces mainframes disposent de fonctionnalités de RAS (Reliability, Availability, Serviceability) permettant de pallier à des problèmes materiels et garantir à l'OS un fonctionnent optimal en permanence.

1.1 Détécter les erreurs

Sur les zSeries, tout est fait afin d'éviter au maximum les erreurs dues au materiel. Cela passe par un bon design de la machine permettant d'éviter les SPOF (Single Point Of Failure) et le choix de composants haute disponibilté ayant subis des tests poussés à l'usine lors de sa fabrication.

Mais meme avec cela, une panne d'un composant reste possible. Afin de limiter les problèmes ces mainframes possèdent des fonctionnalités de détéction d'erreur, permettant de corriger ces erreurs grace à une redondance du materiel et alerter l'administrateur du problème afin qu'il vienne remplacer le materiel defaillant si nécéssaire.

Le mécanisme d'auto-détéction des erreurs utilise la redondance des composants afin de détécter les erreurs.

1.1.1 Processeurs

Chaque unitée de processeur nous avons en réallité une unité d'éxcution d'instructions, un cache L1, et des registres qui sont tous dédoublés. Le résultat

de tous les élements dédoublés permet de comparer les résultats. Lorsque les résultats sont differents, cela veut dire qu'une erreur à été détéctée.

Le z990 possède des unitées processeurs supplémentaires non utilisées permettant de prendre le relais en cas de problème. Lorsqu'une erreur est détéctée, l'instruction ayant causée problème est de nouveau exécuté depuis le meme état initial. Si la 2ème éxecution ne pose pas de problème, le programme en cours continue son éxécution normalement sur le meme processeur. Par contre si lors de cette 2ème tentative d'éxécution, une erreur est de nouveau détéctée, alors une nouvelle unitée de processeur va etre utilisée si il y en a de disponible. Le programme en cours d'exécution est alors transféré sur une unitée disponible sans aucune intervention de l'OS ou de l'application en cours.

1.1.2 Memoire

Comme nous l'avons vu plus au, le cache L1 est disponible est 2 exemplaire afin de permettre la détéction des erreurs. Pour le cache L2 et la RAM, les Error-Correcting Codes (ECC) sont utilisés afin de détécter les erreurs. De la mémoire supplementaire est disponible afin de prendre le relais en cas d'erreur détéctée.

1.2 Réparer les erreurs

Lorsqu'une erreur due à un element de materiel deffectueux, il peut etre nécéssaire de le remplacer. Certains des elements du z990 peuvent etre remplacés à chaud, sans interompre le fonctionnement de la machine :

- CoProcesseurs Cryptographiques
- Certains periphèriques tels que les disques dur
- Consoles de management
- Ventilateurs, batteries, alimentations ...

Pour d'autres elements tels que les modules de mémoire, le remplacement à chaud n'est pas possible, et demande un arret partiel de la machine, qui fonctionnera par exemple avec des performances un peu dégradées lors de l'intervention.

Les serveurs zSeries possèdent donc des fonctionnalitées leur permettant d'etre utilisés dans des environement de haute disponibilité.

Pour plus de details sur les fonctionnalités de haute disponibilité de ces mainframes, la lecture du redbook *IBM eServer zSeries 900 Technical Guide, SG24-5975* est conseillée.

Chapitre 2

z/OS

Le système z/OS a été conçu dès le départ en tenant compte des besoins de fiabilité, de sécurité, et de haute disponibilité qui sont désormais nécessaires à l'informatique moderne. Contrairement à d'autres systèmes qui nécessitent des logiciels extérieurs pour fournir ces services, z/OS intègre directement un certain nombre de fonctionnalités, que ce soit dans le domaine de la sécurité, de la récupération après incident, ou de la répartition de charge.

2.1 Sécurité

Dans le cadre d'une volonté de haute disponibilité, la sécurité du système a bien évidemment une grande importance. Même si de nombreux outils existent pour celà, z/OS se distingue en possédant, dans le système lui-même, d'un certain nombre d'outils dédiés à la sécurité.

2.1.1 Cryptographie et PKI

La sécurité passe souvent par l'utilisation de techniques cryptographiques pour protéger les données circulant sur les réseaux. La plupart du temps, ces méthodes nécessitent l'utilisation de certificats X.509 pour établir la communication. Ces certificats sont bien sur disponibles commercialement auprès de sociétés spécialisées, mais le coût peut être non négligeable, surtout si on doit disposer d'un nombre important de certificats. z/OS intègre directement une solution de PKI, ce qui permet d'éviter ce problème, et permet à l'administrateur d'utiliser des certificats sans craindre une facture non négligeable.

2.1.2 Détection d'intrusion

La détection d'intrusion, qui sur la majeure partie des autres systèmes est un module externe, est directement intégrée dans z/OS, au niveau du mo-

dule *Communication Server*. Elle est directement intégrée à la pile TCP/IP et permet de détecter et de protéger le système contre les attaques réseau.

De plus, à un plus haut niveau, z/OS dispose d'outils d'audit conçus pour détecter des activités suspectes, en loggant les accès refusés à des ressources. De plus, il est possible d'intercepter ces logs par des programmes, qui peuvent alors prendre des actions immédiates en fonction du problème, par exemple bloquer un compte si trop d'accès interdits ont été détectés, ceci pouvant être le signe d'une tentative d'attaque.

2.2 Capacité de récupération

On souhaite bien évidemment éviter les crashs. Mais celà n'est pas toujours possible, les applications n'étant pas infaillibles. Donc il est important de pouvoir à la fois relancer les applications qui ont connu un crash, mais également de disposer d'informations sur le crash pour pouvoir en diagnostiquer la cause.

2.2.1 Récupération des ressources

En cas de problème avec une application tournant sur le système, il est nécessaire de pouvoir la rédémarrer rapidement si on veut assurer la continuité du service. Mais il ne suffit parfois pas de simplement relancer l'application : selon l'endroit dans le code de l'application où le problème s'est produit, il faut restaurer l'état du système dans un état cohérent.

Dans z/OS, le module RTM (*Recovery Termination Manager* est chargé de toutes ces tâches de libération des ressources après une défaillance de l'application, et en particulier de libérer les verrous que l'application pouvait avoir au moment du problème. En effet, à quoi bon relancer une application si c'est pour qu'elle se bloque en attendant que l'ancienne instance (qui n'est plus là...) libère le verrou qu'elle avait sur une ressource...

2.2.2 First Failure Data Capture

Quand une application subit un crash, il est important de pouvoir obtenir un maximum d'informations sur ce qui se passait à ce moment là : en effet, autant il est utile de relancer l'application pour assurer la continuité du service, ce n'est pas suffisant à long terme. Il est important que les administrateurs soient à même de comprendre les raisons des crashs, afin de pouvoir les éviter par la suite en corrigeant le problème de fond.

Pour celà, z/OS fournit un moyen de récupérer des fichiers de dump, contenant l'état actuel de l'application au moment du crash, ainsi que les données systèmes associées. Bien évidemment, cette fonctionnalité est conçue de façon à ne pas perturber le fonctionnement du reste des tâches actives sur le système.

De plus z/OS dispose d'un système de tracing, qui permet d'obtenir des informations à différents niveaux, que ce soit au niveau du système lui-même ou d'un ou plusieurs de ses composants. L'intérêt de ces fonctionnalités est qu'elles sont activables au niveau du système lui-même, sans avoir besoin de relancer l'application que l'on veut surveiller.

2.3 Clustering avec Parallel Sysplex

2.3.1 Avantages de Parallel Sysplex

La technologie Parallel Sysplex est conçue de manière à permettre un accès concurrent à des données partagées entre les noeuds du cluster, tout en optimisant les performances et en garantissant l'intégrité des données partagées.

Chaque noeud peut conserver localement une partie des données partagées, avec un contrôle de cohérence global, afin de garantir que la copie locale des données est toujours en phase avec la version partagée. De cette façon on obtient des performances similaires à un stockage local.

De cette façon, on peut aisément distribuer des requêtes sur plusieurs noeuds du cluster en fonction de la capacité processeur disponible. De plus en cas de problème logiciel ou matériel, il est simple de redistribuer les tâches sur les machines restantes, garantissant la disponibilité de l'application.

En outre, la technologie Parallel Sysplex permet d'effectuer de la maintenance logicelle ou matérielle sans perturber le cluster : il est possible de retirer ou d'ajouter dynamiquement une machine du cluster, sans perturber la vision "externe" des applications qui y tournent.

2.3.2 Partage d'adresse IP et VIPA

De l'extérieur, un cluster Parallel Sysplex apparaît comme une seule machine vue du réseau. Grâce à la technologie VIPA (*Virtual IP Addressing*), une adresse IP est partagée entre les différents noeuds du cluster. En cas de défaillance de la machine gérant l'adresse "publique", une autre machine récupère cette adresse, et continue d'accepter de nouvelles connexions.

De plus, même si les anciennes connexions de clients utilisant l'ancienne machine ne peuvent être maintenues (car tout l'état applicatif n'est pas dupliqué), ces connexions sont terminées au niveau TCP, permettant au client de se reconnecter immédiatement sur le nouveau serveur possesseur de l'IP partagée, sans avoir à attendre de longs timeouts TCP pour se rendre compte que l'ancienne machine ne répond plus.

2.3.3 Automatic Recovery Manager (ARM)

Le module ARM, qui est directement intégré à z/OS, permet aux applications qui se sont enregistrées auprès de lui d'être survéillées, et, en cas de problème, d'être automatiquement relancées, soit sur la même machine, si le problème est purement applicatif, soit sur une autre machine, si la machine d'origine n'est plus en état de fournir le service.

Chapitre 3

z/VM

3.1 Qu'est ce z/VM ?

z/VM (Virtual Machine) est un systeme proprietaire IBM
- il fonctionne sur mainframe IBM ou compatible
- Premier systeme temps partage, cree en 1964
- concept de machine virtuelle
- supporte tous les OS IBM mainframe et linux

3.2 Pourquoi z/VM ?

Z/VM permet de virtualiser les ressources systemes. Cela permet d'installer plusieurs systemes sur des machines crees virtuellement. Avec z/VM on peut, au sein d'un seul mainframe creer des infrastructures systemes et reseau complexes pour chaque client heberge dessus. Ce qui par exemple offre un environnement de test et de production extremement souple pour les entreprises qui deploient des solutions e-business de pointe. Construit sur une base VM/ESA solide, z/VM offre aux entreprises les solutions serveur multi-systeme dont elles ont besoin, avec une prise en charge de nombreux systemes d'exploitation comme z/OS, OS/390, TPF, VSE/ESA, CMS, Linux pour OS/390 ou Linux pour zSeries.

Les avantages d'un tel systeme :
- l'infrastructure est consolidee
- benefice de la robustesse mainframe pour les serveurs Unix/linux
- pas de gachi de ressources : elles sont reparties sur toutes les machines virtuelles

par exemple prenons le cas d'une ferme de 100 serveurs x86 classiques hebergeant des sites web. Les serveurs ne seront jamais utilises au maximum de leur ressource en meme temps : d'ou un gachi des ressources systemes.

3.3 Les composants de z/VM

- le CP (Controle Program)
- le CMS (Conversional Monitor System)

L'hyperviseur CP s'occupe de la virtualisation, c'est a dire qu'il virtualise les ressources materielles, les partitionne ou encore les partage ce qui permet a chacune des machines virtuelles cree d'avoir sa propre memoire, ses propres peripheriques et son propre processeur ; qui peuvent etre virtuels ou pas. Ce concept de virtualisation permet d'executer simultanement plusieurs copies du meme systeme d'exploitation ou de fifferents systemes.

Le moniteur CMS lui, est l'outil utilise pour administrer z/VM, il permet par exemple la gestion des fichiers, le lancement autonome de systeme sur differente machines virtuelles etc...langage de commande permetant cela est REXX et l'editeur utilise est XEDIT. CMS est en fait un systeme d'exploitation qui tourne aussi sur une machine virtuelle, qui offre a l'utilisateur (le plus souvent l'admin), une interface utilisateur et de developpement ressemblant a celle d'un environement shell unix

3.4 Recouvrement des erreurs

Le microcode des zSeries et z/VM essaie tant que possible de palier a un certain nombre d'erreur de maniere autonome, c'est a dire sans intervention manuelle. Quand une erreur survient, le CP enregistre cette derniere et envoie un message a la console de supervision en specifiant une de ces trois possibilites :

- System operation can continue (Le systeme considere que l'erreur n'a pas d'incidence grave sur le fonctionnement globale)
- System operation can continue, but with fewer ressources (Le systeme considere que l'erreur affecte une partie des ressources, ou qu'il est plus prudent de limiter la charge afin que l'execution du programme se poursuit sans dommages)
- System restart and recovery is beginning (erreur ne peut etre toleree, le systeme redemarre et corrige l'erreur)

Dans le cas ou l'erreur n'affecte pas l'integrite du systeme, et qu'elle influe que sur un unique utilisateur ; z/VM arrete le programme fautif et permet a l'utilisateur de continuer a travailler. Ce qui montre que quelque soit la cause d'un crash d'un utilisateur, le processus CP ne sera pas atteint et cet incident sera sans consequence pour les autres utilisateurs.

Dans tous les cas un message est envoye a la console de supervision, message qui peut etre analyser et l'administrateur peut decider a tout moment d'entreprendre une action.

A cette gestion des erreurs logiciel, est associee une gestion materielle permettant par exemple de detecter a tout moment une erreur d'un processeur, ou sera effectue dans ce cas un echange de processeur de maniere transparente. Le processus en execution continuera a s'executer comme si aucune erreur ne s'etait produite. Notons bien que la gestion de ce genre d'erreur est purement materielle et que z/VM ou tout autre systemes tournant sur le zSeries (Linux ou z/OS)ne s'en rendra compte.

3.5 Communication entre plusieurs systemes z/VM

Le mechanisme de battement de coeur est tres important dans une infrastructure de haut disponibilite et pour assurer cela une connexion doit etre etablie entre differents z/VM. Effectivement, deux ou plusieurs zSeries peuvent etre relies avec une connexion "channel-to-channel" (CTC) via les deux architectures d'entree/sortie des zSeries, "Enterprise Systems Connection" (ESCON) ou "Fiber Connection" (FICON). La connexion CTC simule un pilote d'entree/sortie pour z/VM et fournit le chemin et assure la synchronisation pour les transfert de donnees entre deux canaux. La connexion CTC permet ainsi d'obtenir une architecture multiprocesseurs.

La communication entre plusieurs systemes z/VM requiere un programme de communication qui assure le tranfert de messages. le "Remote Communications Subsystem" (RSCS) est un programme reseau qui permet aux utilisateurs loggues sur un systeme d'envoyer des messages a d'autres utilisateurs sur d'autres systemes ou encore d'executer des commandes et des processus sur un autre systeme.

3.6 Recouvrement d'incidents

La solution de recouvrement d'incidents garantissant la haute disponibilite des infrastructure z/VM est "Geographically Dispersed Parallel Sysplex "(GDPS). GDPS est une application multi-site operationnelle permettant de maniere souple et efficace la replication des donnees, et la sauvegarde de sous-systemes. Permettant ainsi de conserver en toute integrite la totalite des donnees et de poursuivre les operations en cours sur d'autre z/VM.