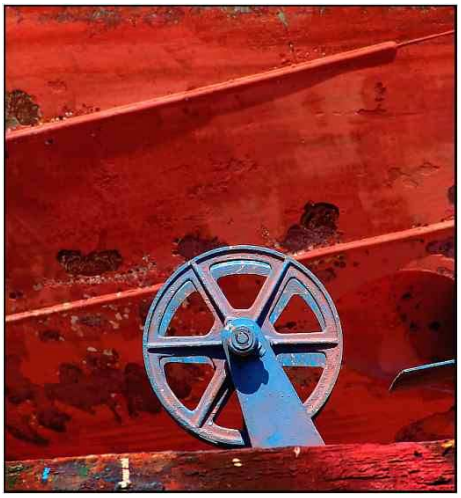
Enezis



Virtualisation des ressources serveur

Exemple : Systèmes partitionnés sous HP-UX et Oracle

Enezis

Sommaire

1 Principes de la virtualisation des serveurs

Les grandes entreprises ont entrepris de consolider les serveurs de bases de données après avoir migré les données sur des réseaux de stockage. Bien qu'actuellement peu exploité, l'un des avantages de la consolidation réside dans la virtualisation des ressources des serveurs, en particulier de leurs CPUs. On recherche essentiellement ici à atteindre deux objectifs :

- Meilleure adaptation des ressources à la demande : cette dernière varie en effet de façon significative au cours du temps (fin de mois, traitements de nuits, arrêtés comptables, etc.)
- Meilleure utilisation des ressources par l'augmentation des taux d'utilisation (ici des CPU) : les ressources inutilisées sont déplacées sur les partitions où elles seront mises à profit.

Dans le domaine des serveurs, la virtualisation peut prendre deux formes :

- Utilisation de serveurs banalisés, typiquement des blades sous Windows ou Linux. Les procédures de déploiement permettent de déployer ou de remplacer rapidement des serveurs pour répondre rapidement aux demandes des métiers.
- Utilisation de serveurs « virtualisés », pourvus de ressources virtuelles. On trouve dans cette catégorie les serveurs dotés de VMWare, qui partagent des ressources entre plusieurs machines virtuelles sans pour autant contrôler la répartition de la puissance, et les systèmes partitionnés comme les systèmes PA-RISC et Integrity de Hewlett Packard.

L'exemple présenté dans ce document est basé sur le VSE (Virtual Server Environment), qui est un ensemble d'outils techniques mais aussi financiers qui permettent de mieux aligner les ressources sur la demande de puissance.

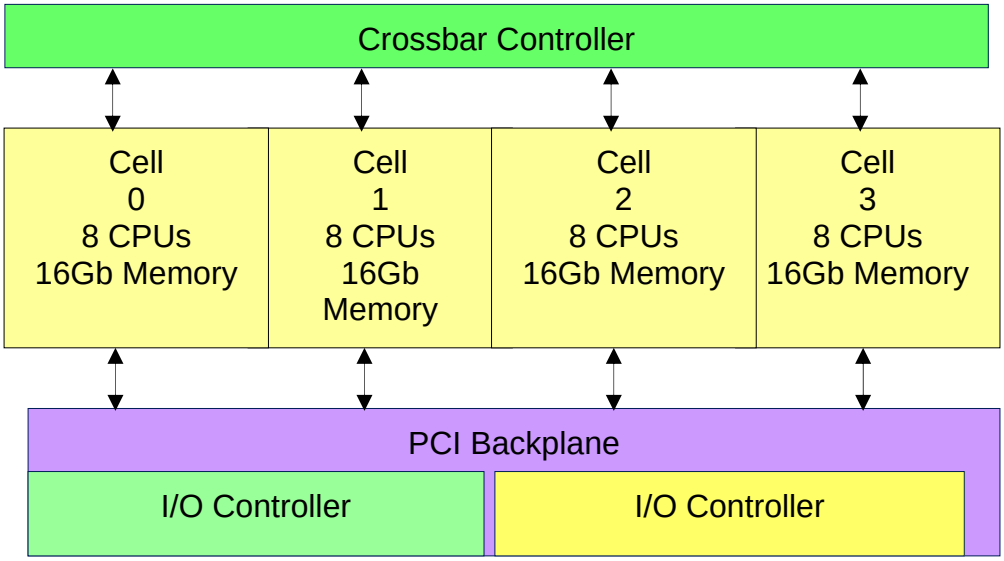
2 Principes de la virtualisation des serveurs partitionnés sous HP-UX

2.1 ARCHITECTURE MATERIELLE

Les serveurs partitionnés sont dotés de cellules, qui contiennent quatre emplacements pour les CPU. Ces derniers peuvent êtres des processeurs mono-core, dual-core ou encore un mx2 (un double processeur sur la même carte). Une cellule peut donc contenir 4 ou 8 processeurs. Les processeurs appartiennent soit à la gamme « historique » PA-RISC, dont le PA-8900 est le dernier représentant, soit à la gamme Itanium. Les processeurs PA-RISC PA-8800 et PA-8900 sont des « dual-core », tandis que les processeurs Itaniums peuvent être mono core ou décliné en version mx2. Le véritable « dual core » Itanium verra le jour en 2006.

Le PA-RISC comme l'Itanium acceptent le système d'exploitation HP-UX. Windows, OpenVMS et Linux sont quand à eux disponible uniquement sur les Itaniums.

Le schéma de principe ci-dessous montre par exemple un serveur rp8400 doté de quatre cellules. Les serveurs de haut de gamme sont les Superdome, qui existent en version Integrity (Itanium) ou HP9000 (PA-RISC). Ils permettent de disposer de 16 cellules au maximum, donc de 128 CPU.



2.2 OUTILS DE LA VIRTUALISATION

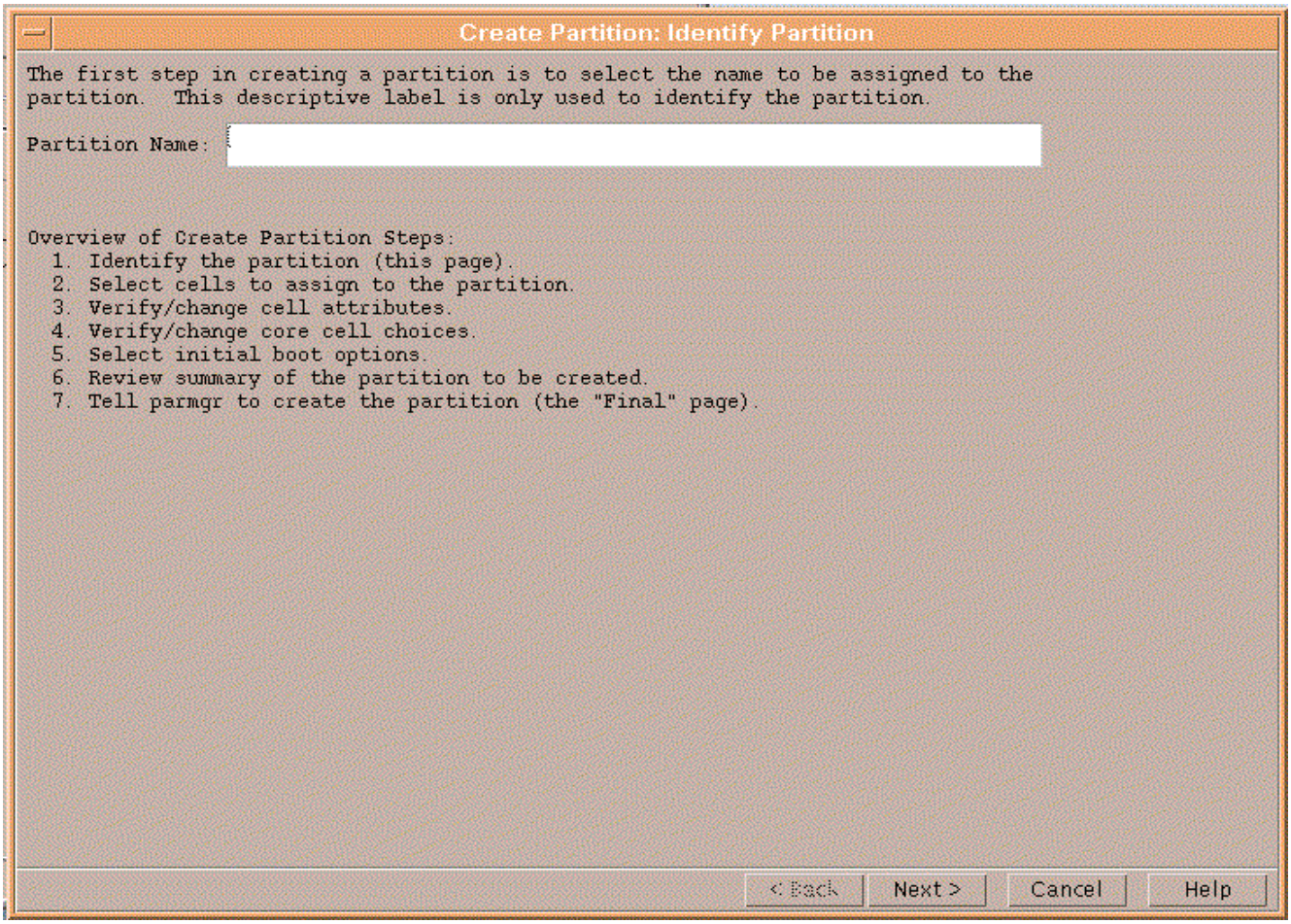
2.2.1 Partitions matérielles (npar)

Les npar sont des groupes de cellules. Sur un système doté de 4 cellules équipées de processeurs Itanium mono-core, une npar peut donc par exemple être constituée d'une cellule de 4 Itaniums, tandis qu'une autre npar sera constituée de trois cellules, dotée de 12 CPU. Chaque npar peut contenir des systèmes d'exploitation différents : HP-UX 11iv2, Windows 2003 (Enterprise et Datacenter editions), Linux ou OpenVMS. Il n'y a pas de mutualisation de ressources physiques CPU et Entrées/ Sorties : ceci permet une relative isolation jusqu'au niveau électrique des différentes applications déployées dans les partitions physiques. Les partitions physiques peuvent elles mêmes être découpées en partitions logiques (vpar sous HP-UX), mais on peut également doter les partitions d'un système d'exploitation sans vpar. Windows et Linux ne proposent d'ailleurs pas encore de système complémentaires de partitionnement (bien que l'Integrity VM sur systèmes EPIC – Itanium – doive combler ce manque).

Les partitions peuvent être créées par ligne de commande :

parcreate -P testing -c 4:base:y:ri -b 4/0/1/0/0.9 -B

ou par un utilitaire graphique (parmgr):



2.2.2 Partitions logiques (vpar – virtual partitions)

Les partitions virtuelles sont créées à l'intérieur des partitions. Contrairement à ces dernières qui sont physiquement isolées les unes des autres, les vpar partagent logiquement un espace physique. HP-UX est le seul des quatre systèmes d'exploitation à proposer les vpar. Les vpar imposent quelques restrictions, la principale étant la dépendance des versions des systèmes d'exploitation des vpar d'une même npar. Mais les vpar offrent beaucoup de souplesses : elles permettent

- l'activation de nouveaux CPU (icap – Instant Capacity On Demand)
- les mécanismes de paiement à la consommation (PPU – Pay Per Use)
- le déplacement de cpu de vpars en vpars. Les cpu candidats doivent être déclarés « flottants ». Ils ne pourront pas manipuler d'interruptions (en particulier les E/S).

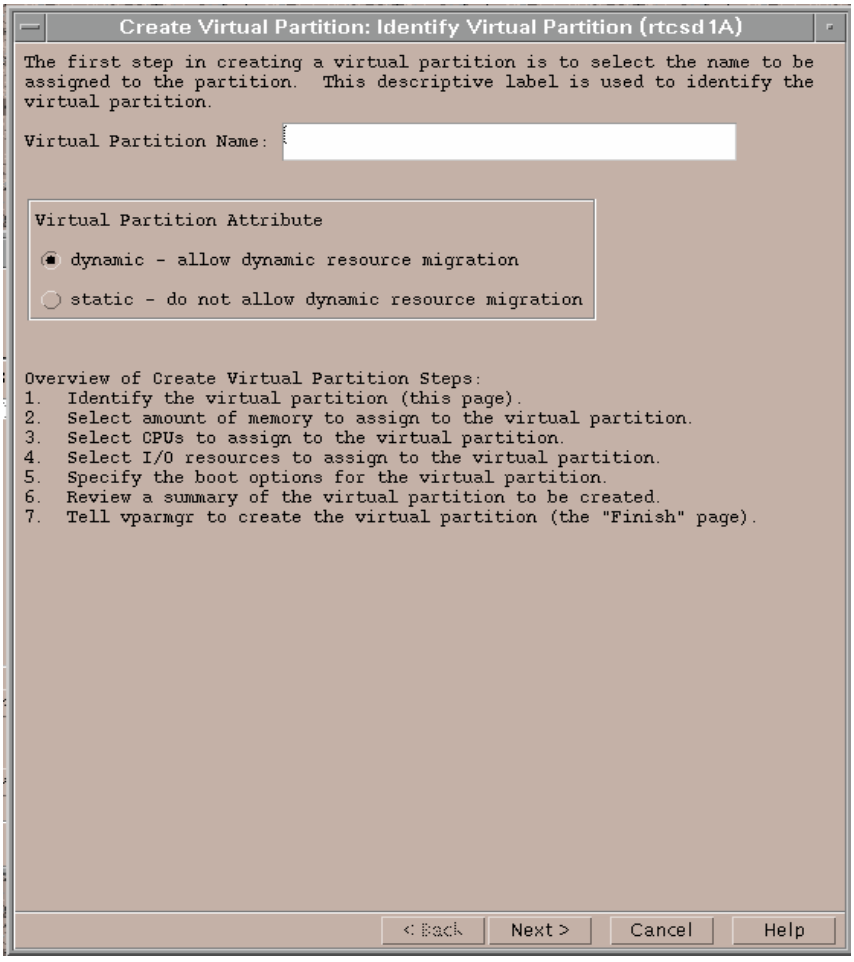
Les partitions virtuelles peuvent elles aussi être crées à partir de la commande vparcreate :

```
vparcreate -p vpar2 -a cpu::2 -a cpu:::2:8 -a mem::16384 \
                    -a io:4.0.0.8.0.0 -a io:0.0.6.0.1 -a io:0.0.6.0.1.5.0:BOOT
```

Création de la vpar nommée vpar2, dotée d'un maximum de 8 CPU, dont 2 permanents, ainsi que de 16Go de mémoire.

L'utilitaire graphique vparmgr permet également d'effectuer une configuration :

vparmgr :



2.2.3 Mécanismes technico-financiers

Les mécanismes de paiements Icap (Instant Capacity on Demand), Ticap (Temporary Icap) et Pay per use sont des mécanismes qui participent à la virtualisation : il s'agit ici de proposer aux sociétés utilisatrices des ressources facturées en fonction des besoins de consommation.

Icap

Activation de processeurs, de mémoire et de disques déjà présents dans les systèmes en fonction de la croissance de l'activité. L'Icap permet en fait de différer les achats de matériels, en ne les facturant qu'au moment ou ils seront utilisés : l'activation de ces matériels supplémentaires se faisant par simple modification de licence, aucune intervention sur des ordinateurs en production ne sera nécessaire, l'ajout peut être effectué à tout moment, sans à coup.

Extrait du « man » d'icod_modify :

 # /usr/sbin/icod_modify

- System contact: **-c** *UserName:EmailAddr:Phone#*
- Activate processors: **-a** *#ofProcessorsToActivate desc:UserName:AuthorizingManager*
- Set active processors: **-s** *#ofActiveProcessors desc:UserName:AuthorizingManager*
- Deactivate processors: **-d** *#ofProcessorsToDectivate desc:UserName:AuthorizingManager*

- Defer activate/deactivate: **-D -[a|d]** *#ofProcessors desc:UserName:AuthorizingManager*

Ticap (Temporary Instant Capacity On Demand)

Si l'Icap à pour objet de différer l'activation de ressources pré installées, le Ticap permet quand à lui la désactivation de CPU, de zones de mémoires et de ressources de stockage. Ce mécanisme est particulièrement intéressant lorsque l'activité varie à la hausse comme à la baisse sur une période de temps donnée : fin de mois, clôtures comptables, consolidation en fin de journée dans les salles de marché, etc.

PPU (Pay Per Use)

Le PPU est un système original de facturation à la consommation. Il est lui-même proposé sous deux formes :

- A la consommation de ressource : facturation =f(pourcentage des CPU utilisés)
- Facturation à l'utilisation de CPU : facturation =f(nombre moyen des CPU utilisés)

2.2.4 Gestion des ressources : WLM (Workload Manager)

2.2.5 Service Level Objectives

La richesse principale du gestionnaire de ressources WLM est de permettre la définition de SLO (Service Level Objectives).

Un SLO est défini par une combinaison d'objectifs, de contraintes et de conditions. Un SLO est basé sur les éléments suivants (certains sont optionnels) :

- « Une charge de travail » (workload), généralement définis par une définition générique de process (exemple : ora_*)
- Des contraintes d'occupation des ressources (par exemple pourcentage minimum et maximum d'utilisation des CPU)
- Un ou plusieurs objectif (cf. ci-dessous)
- Une priorité, afin d'arbitrer entre des demandes contradictoires, comme lorsque deux charges de travail doivent se partager des ressources saturées.
- Des conditions (horaires, évènements, etc.)

Les SLO peuvent combiner des objectifs de natures différentes :

- Utilisation de ressources, par exemple un pourcentage de CPU
- Performance de la charge de travail, par exemple un temps de réponse ou un débit
- Mesure spécifiques aux applications : nombre d'utilisateurs ou de process, files d'attentes, etc.

Exemples d'objectifs :

- Groupe d'ordres SQL pour lesquels les temps d'exécution doivent être inférieurs à 2s
- Chaque process de la charge de travail ora_* dispose d'un vingtième à un dixième de CPU.

2.2.6 Outils utilisables par le Workload Manager :

Le WLM peut répartir des ressources au sein d'une machine physique ou d'une partition. Il pilote alors le PRM (Process Ressource Manager), qui permet de fixer des quotas d'utilisation des ressources matérielles pour les charges de travail.

Mais la richesse du WLM réside surtout dans sa capacité à utiliser les fonctionnalités de utility computing évoqués au début de ce document :

- Migrations de CPU à travers les vPars
- Migration Icap de CPU à travers les npars
- Activation et désactivation Icap et PPU
- Réallocation dynamique de la mémoire (lors du démarrage de batchs par exemple)

Exemple d'un extrait d'un fichier de configuration :

```
slo Ora_1_slo {
        pri = 1;
        mincpu = 15;
        maxcpu = 75
        goal = metric O1_transaction01_time < 300;
        entity = PRM group Ora_grp_1;
}

tune O1_transaction01_time {
    coll_argv = wlmrcvdc
                  wlmoradc
                    --configfile times.oradc
                    --instance instance1
    ;
}
```

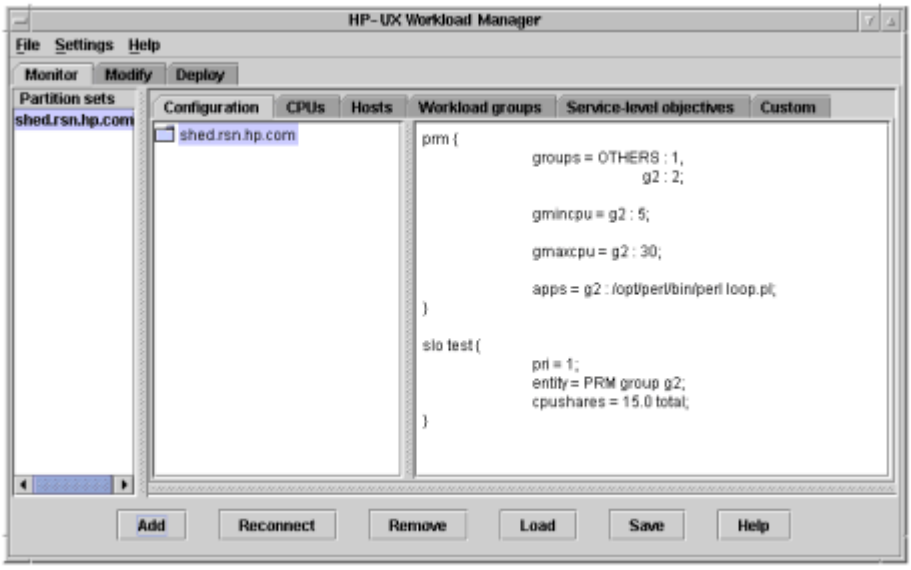
Il faut comprendre la méthode de collecte (champs tune) de la façon suivante :

Recevoir le métrique `o1_transaction01_time` en attendant l'envoi par la ligne de commande `wlmoradc`.

Le fichier `times.oradc` contient :

`Select numero_compte from transactions where numero_compte=123456`

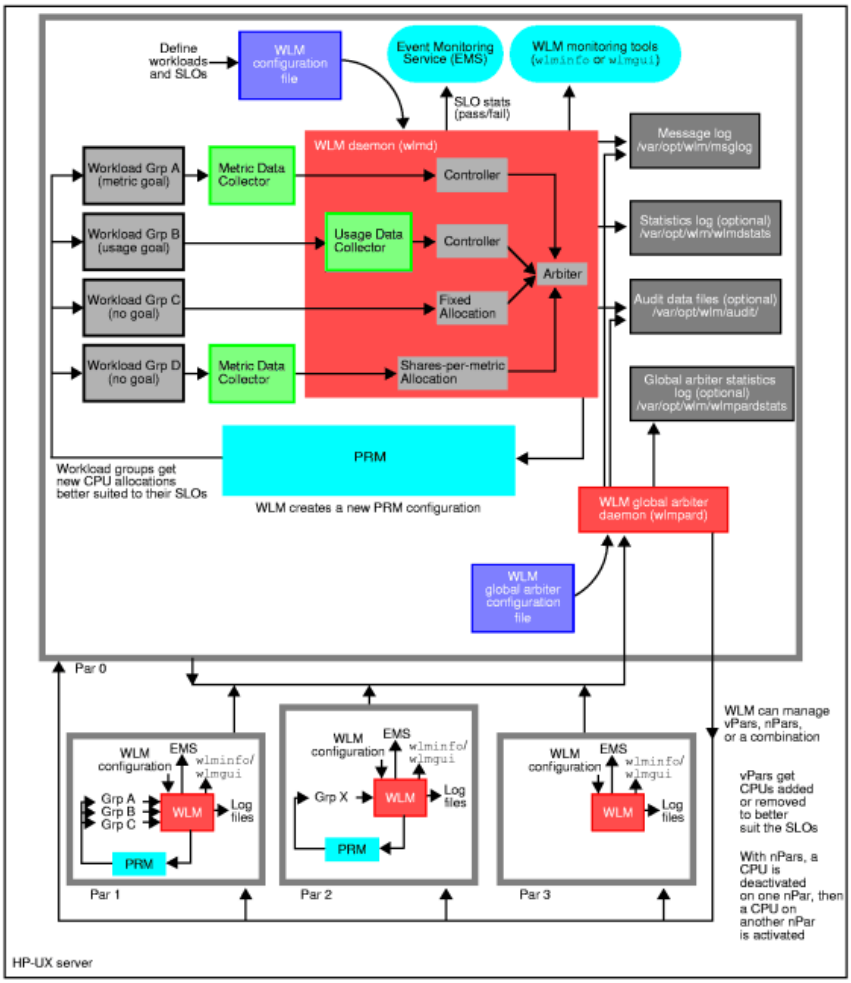
Le fichier de configuration peut être alimenté par une interface graphique:



Il ne reste plus qu'à programmer le lancement à intervalles réguliers de `wlmoradc` dans la crontab :

```
# wlmoradc --configfile times.oradc
```

2.2.7 Schéma de principe du Workload Manager:



3 Conclusions

Hewlett Packard dispose à travers la combinaison des nPars et des vPars, de l'Icap, du Ticap et du PPU, et enfin du WLM d'un environnement virtualisé qui a permit de constituer une offre nommée Virtual Server Environment (VSE). La présentation simplifiée à l'extrême de l'exemple de la configuration du WLM était basée sur un seul ordre SQL sous Oracle. Il existe des toolkits pour Oracle, BEA Welogic ou encore SAS qui permettent de disposer d'un framework qui peuvent être adaptés aux environnements des utilisateurs.