le cnam

Les systèmes à haute disponibilité

NSY 104

Introduction

- Suite à une défaillance, un système à haute disponibilité doit être en mesure de maintenir le service qu'il est censé rendre
  - Un système à haute disponibilité masque ses défaillances aux utilisateurs.
  - Le système maintient en permanence plusieurs versions de l'état courant du système
    - En cas de défaillance, l'exécution se poursuit à partir de l'un des états courants disponibles.
  - Le système maintient des états à partir desquels le traitement peut être repris.
    - C'est la technique des points de reprise.
- Les hypothèses dans lesquelles le traitement en cours serait abandonné et où le système serait réinitialisé ne sont pas envisageables.
- Le maintien de plusieurs contextes implique une redondance coûteuse au niveau matériel.

Sureté de fonctionnement

On appelle critique une fonction d'un système pour laquelle la propriété de sûreté de fonctionnement est une contrainte stricte. Un défaut de fonctionnement peut entraîner la perte de la mission ou des dommages inadmissibles sur le système ou sur son environnement.

Sureté de fonctionnement

- **Fiabilité** *(reliability).*
  - Cela correspond à la continuité de service rendu.

- **Disponibilité** *(availability)*
  - Aptitude du système à rendre le service pour lequel il a été conçu. Elle dépend des caractéristiques du système et de sa maintenance et se mesure par le rapport entre le temps pendant lequel le service est disponible et une période de temps considérée

- **Maintenabilité** *(maintenability)*
  - Cela correspond à la possibilité de maintenir un système en condition opérationnelle, mais aussi à recevoir des réparations, des modifications éventuellement en phase opérationnelle.

Sureté de fonctionnement

- **Innocuité** *(safety)*
  - C'est la sécurité vis-à-vis de l'environnement. Cela correspond à l'absence d'événements ayant des conséquences non désirées sur l'environnement du système.

- **Immunité** *(immunity)*
  - C'est la résistance du système aux agressions extérieures. Les critères essentiels sont l'intégrité et la confidentialité.
    - L'intégrité assure que les données ne peuvent pas prendre des valeurs inadéquates.
    - La confidentialité assure l'absence d'accès non autorisés à ces données.

Sureté de fonctionnement

Impact de l'indisponibilité d'un système suivant le secteur d'activité :

Application	Secteur d'activité	Coût de l'indisponibilité d'une heure
Courtage	Finance	$6.45 millions
Ventes par carte	Finance	$2.6 millions
Films à la demande	Loisirs	$150 000
Téléachat	Distribution	$113 000
Ventes sur catalogue	Distribution	$90 000
Réservation aérienne	Transport	$89 500

Sureté de fonctionnement

Classification des systèmes en fonction de leur disponibilité :

Type de système	Indisponibilité (en minutes par an)	Disponibilité	Classe de disponibilité
Non géré (*unmanaged*)	50 000	90	1
Géré (*managed*)	5 000	99	2
Bien géré (*well managed*)	500	99.9	3
Tolérant les fautes (*fault tolerant*)	50	99.99	4
Haute disponibilité (*high availability*)	5	99.999	5
Très haute disponibilité (*very high availability*)	0.5	99.9999	6
Ultra haute disponibilité (*ultra availability*)	0.05	99.99999	7

Cinq minutes par an, soit : 60 * 24 * 365 – 5 / (60 * 24 * 365) (525600),
= rapport de 99.999
Ce rapport mesure la disponibilité d'un système et détermine sa classe.

Concepts

- **Service**
  - Ensemble des résultats et conditions de délivrance que le système doit fournir à l'utilisateur.
- **Défaillance** (*Failure*)
  - Discordance entre le service fourni à l'utilisateur et le service attendu.
- **Erreur** (*Error*)
  - Discordance entre une valeur ou condition calculée et la valeur ou condition théorique correspondante.
- **Défaut** (*Latent Error*)
  - Partie du système inadaptée ou manquante
- **Faute** (*Fault*)
  - Dysfonctionnement du système ou des personnes en charge de la production.

Faute ⟶ Défaut ⟶ Erreur ⟶ Défaillance

Etats du système

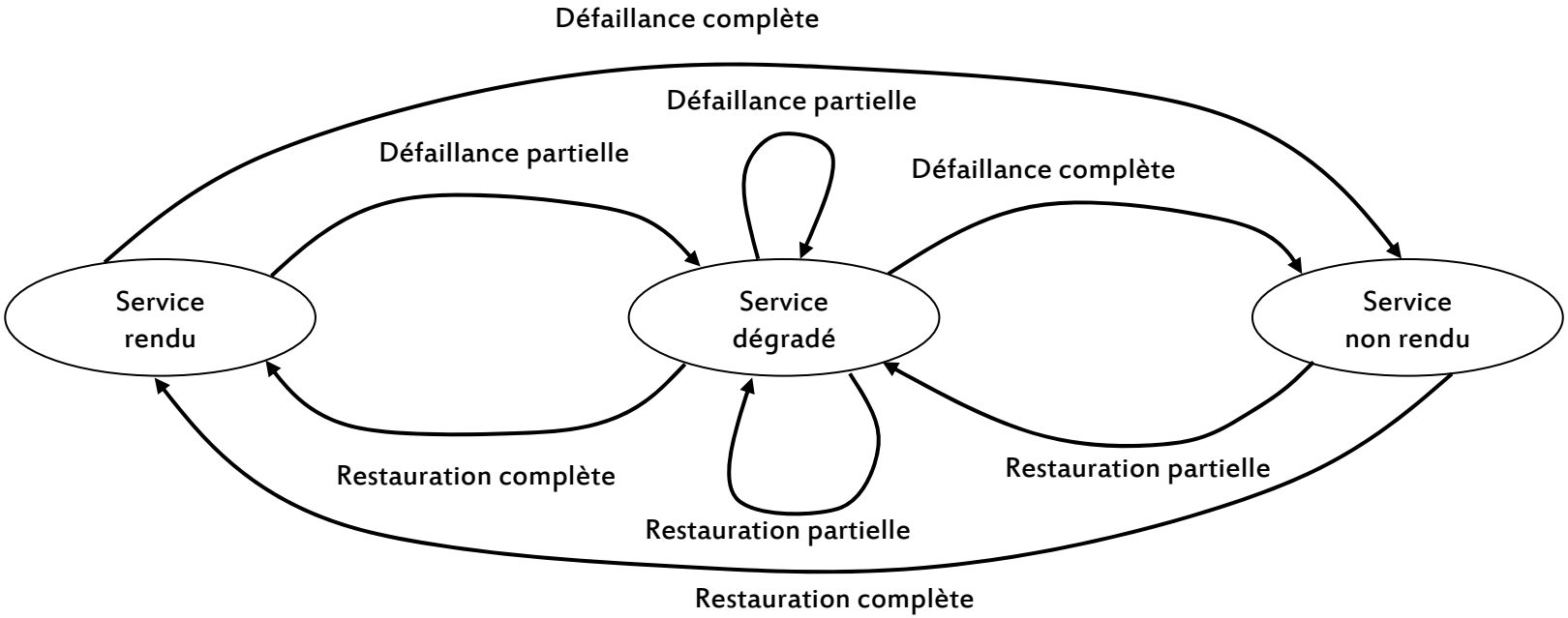
Vie opérationnelle d'un système :

- **Service rendu** (*proper service*)
  - lorsque les résultats fournis et leurs conditions de délivrance sont conformes à ceux du service attendu.
- **Service dégradé** (*degraded service*)
  - lorsque les résultats sont conformes à ceux du service rendus, mais que leurs conditions de délivrance ne sont pas conformes à celles du service attendu.
- **Service non rendu** (*improper service*)
  - lorsque les résultat fournis ne sont pas conformes à ceux attendus.

Une défaillance (*failure*) est une discordance observée entre le service fourni à l'utilisateur et le service attendu.

Une défaillance peut être détectée par l'utilisateur (humain ou autre système) ou par le système lui – même.

Etats du système



- **Défaillance complète** (*complete failure*)
  - C'est une discordance sur les résultats du service, par exemple blocage ou crash du système d'exploitation.
- **Défaillance partielle** (*partial failure*)
  - C'est une discordance sur les conditions de la délivrance des résultats, par exemple, perte de performance ou support d'un nombre moindre d'utilisateurs.

Contraintes du système

- La réponse est fonction de l'état initial du système et des données en entrée.
- La réponse doit être fournie à l'intérieur d'un intervalle de temps.
- Le système doit se trouver dans un état résultant spécifié.
- Les valeurs des données fournies par le système sont spécifiées.

Défaillances du système

- **Défaillance par omission** (*omission failure*)
  - Cette classe de défaillance se caractérise par une absence totale de réponse du système à une sollicitation.

- **Défaillance temporelle** (*timing failure*)
  - Cette classe de défaillance se caractérise par le fait que le système ne répond pas à une sollicitation dans l'intervalle de temps spécifié. On peut distinguer une défaillance hâtive (early timing failure) et une défaillance tardive (late timing failure) suivant que le système réponde trop tôt ou trop tard.

- **Défaillance de réponse** (*response failure*)
  - Cette classe de défaillance se caractérise par le fait que le système fournit une valeur incorrecte (défaillance de valeur ou value failure) ou par le fait que la transition d'état du système qui se produit est incorrecte (défaillance de transition d'état ou state transition failure).

- **Défaillance de type « plantage »** (*crash failure*)
  - Si après la première omission à produire une réponse, un système ne répond plus aux sollicitations suivantes avant qu'il fasse l'objet d'une réinitialisation, on considère qu'il subit un défaillance de type « plantage ».

Indicateurs

- **Service :**
  - *MUT, Temps moyen de service rendu (Mean Up Time).*
    - durée pendant laquelle le système assure le service attendu
  - *MDT, Temps moyen de service non rendu (Mean Down Time).*
    - durée pendant laquelle le système n'assure pas le service attendu.

- **Disponibilité** (technique ou utilisateur)
  - *At, Disponibilité technique*
    - At = MTTF/MTBF ou MTTF/(MTTF+MTTRes)
  - *Au, Disponibilité utilisateur*
    - Au = MUT/(MUT+MDT)

- Le MTTRep caractérise le temps nécessaire à la réparation des éléments alors que le MTTRes caractérise les temps nécessaires à la remise en état du bon fonctionnement du système.

- D'autres mesures de ce type existent pour l'évaluation de la prise en charge des défaillances du système.

Indicateurs

- **Défaillance :**
  - *MTTF, Temps moyen jusqu'à la défaillance (Mean Time to Failure).*
    - durée de fonctionnement jusqu'à la première défaillance.
  - *MTBF, Temps moyen entre défaillance (Mean Time Between Failure).*
    - durée entre deux défaillances. Ce temps comprend la durée de remise en état du système.

- **Maintenabilité :**
  - *MTTRes, Temps moyen jusqu'à restauration (Mean Time To Restore (recover)).*
    - durée jusqu'à restauration.
  - *MTTRep, Temps moyen jusqu'à réparation (Mean Time To Repair element).*
    - durée jusqu'à réparation.

- Le MTTRep caractérise le temps nécessaire à la réparation des éléments alors que le MTTRes caractérise les temps nécessaires à la remise en état du bon fonctionnement du système.

- D'autres mesures de ce type existent pour l'évaluation de la prise en charge des défaillances du système.

Principes de conception

La conception des systèmes à haute disponibilité repose sur :

- La modularité.
- Fail fast.
- Indépendance des modes de défaillance.
- Redondance et réparation.
- Elimination des points de défaillance unique.

Principes de conception

La modularité :

• Un système est constitué par un ensemble de modules.
• La décomposition d'un système de modules est le résultat de la conception fonctionnelle du système.
• Un module est une unité de service, de cloisonnement des fautes.

Fail fast :

• Tout module fonctionne de façon correcte ou arrête immédiatement son exécution dès la détection d'une erreur.

Indépendance des modes de défaillance :

• Les différents modules constituants le système et leurs interconnexions doivent être conçus de façon que la défaillance d'un module n'affecte pas les autres modules.

Principes de conception

Redondance et réparation :

La redondance est l'une des techniques de base pour la construction de système à haute disponibilité. L'installation de modules de rechange (niveau physique) et leur prise en compte dans la configuration du système doit pouvoir se faire pendant le régime de fonctionnement normal du système.

La redondance et la réparation sont des éléments clés de la disponibilité des systèmes.

Elimination des points de défaillance uniques :

Un point de défaillance unique est un élément dont la défaillance entraîne la défaillance complète du système. L'un des principes de la conception des systèmes à haute disponibilité consiste à identifier et à éliminer les points de défaillance unique.

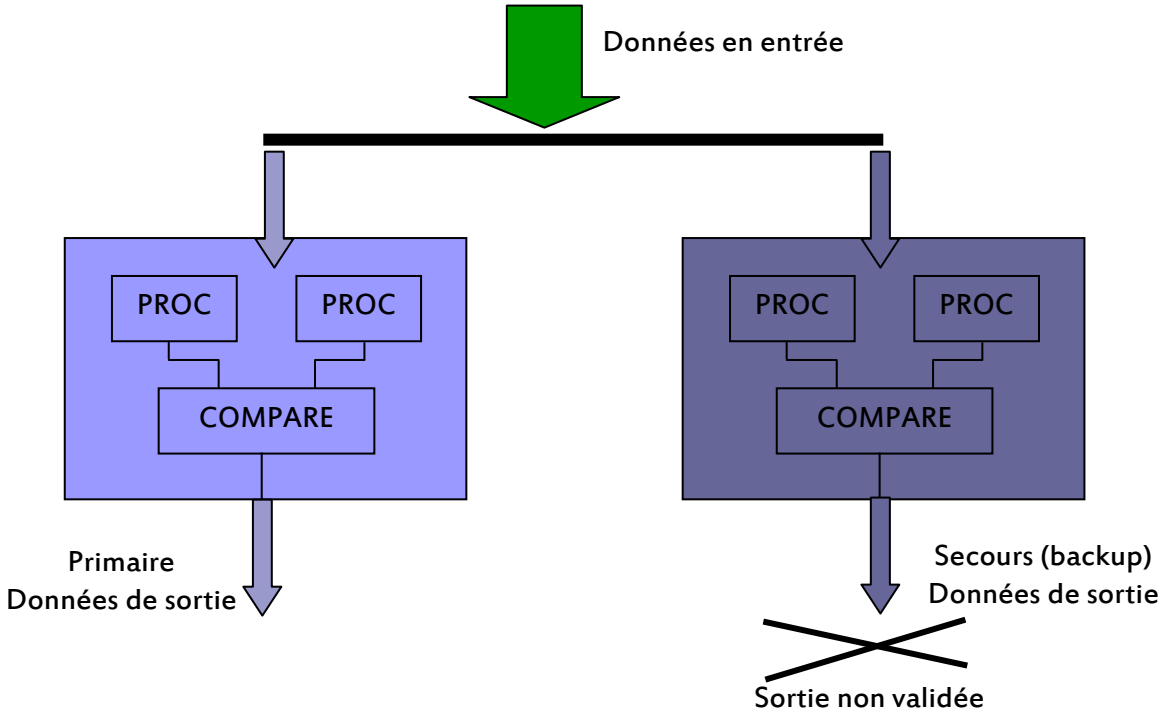
Exemple de points de défaillance unique :

- L'alimentation en énergie électrique pour un système ne disposant pas d'une alimentation secourue.
- Le bus système pour un SMP.
- Le système d'exploitation dans le cas d'un système monoprocesseur ou d'un système multiprocesseurs symétrique : la défaillance du système provoque l'arrêt du système et le retour à l'état opérationnel nécessite une réinitialisation du système.

Solution matérielle: Stratus

Approche *Pair and Spare* (doublement et rechange) :

- Appariement des organes actifs, *Pair*
  - consiste à faire exécuter une fonction en parallèle sur deux processeurs au sein d'un même organe et de comparer pas à pas les résultats.
- Doublement des organes actifs, *Spare*.
  - On double l'unité de traitement par exemple.



Solution matérielle: Stratus

Une unité de traitement logique est composé de deux cartes processeurs physiquement indépendantes. Ces cartes exécutent le même processus de façon synchronisée, les mêmes instructions et les mêmes données étant présentées aux deux cartes processeur.

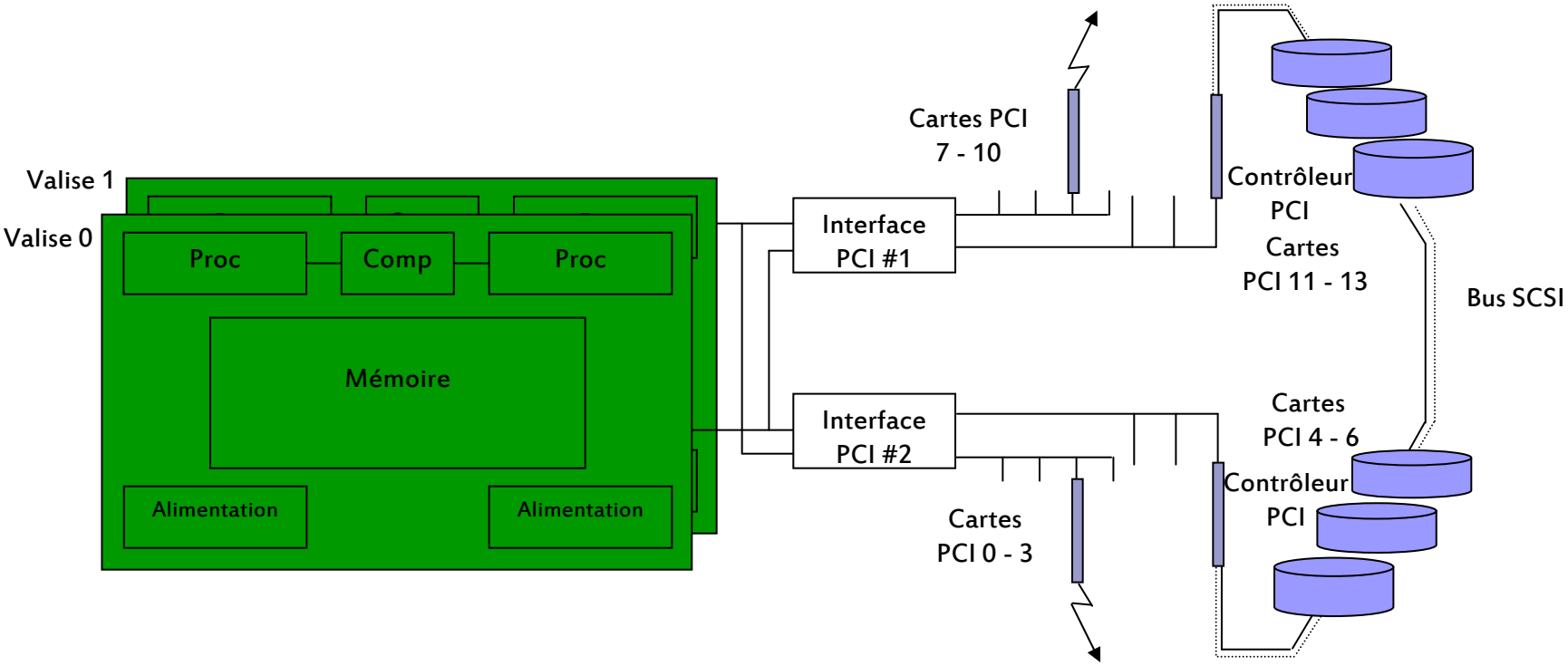
L'une des cartes est dite carte primaire, ses sorties sont validées, l'autre carte est dite carte de secours (backup) et ses sorties ne sont pas validées.

Chaque carte est munie de deux processeurs qui travaillent de façon synchronisée. Les résultats fournis par les deux processeurs sont comparés. En cas de divergence, la carte processeur est déclarée en défaut et est écartée de la configuration. Aucun des résultats produits par les processeurs de la carte primaire n'est pris en considération si cette carte est en défaut. Comme le traitement se produit aussi sur la carte de secours, cette carte devient la carte primaire de façon transparente et le traitement continue.

Suite à une déclaration de carte en défaut, des tests sont exécutés de façon à déterminer la nature de la défaillance et à décider de déclencher une demande de remplacement de la carte. Une nouvelle carte peut alors être introduite dans le système « à chaud ».

Cette nouvelle carte est alors configurée comme carte de secours et se synchronise avec la carte primaire. Tous les éléments du système peuvent être remplacés dynamiquement sans qu'il soit nécessaire d'interrompre le fonctionnement du système.

Solution matérielle: Stratus



Solution matérielle: Stratus

Ce système utilise des microprocesseurs PA7100 ou PA800, PA pour Precision Architecture, de type RISC (reduced instruction-set computer) d'origine HP.

L'unité de traitement est conditionné sous forme de valise (*case*) et comprend deux microprocesseurs, la logique de comparaison, la mémoire (jusqu'à 2 Go), l'alimentation électrique et le ventilateur.

Chacune des valises peut accéder au deux bus d'entrées – sorties au standard PCI. Les disques sont en double accès entre les contrôleurs SCSI, liés chacun à des bus PCI. Les disques sont donc accessibles en cas de défaillance d'un bus PCI ou d'un des contrôleurs.

Le système fonctionne sous le contrôle d'un système d'exploitation propriétaire ou de HP-UX. Stratus prévoit de faire évoluer son système sur les microprocesseurs IA-64 de façon à faire tourner X-Window.

Solution matérielle: Netra FT 1800

Sun microsystems. Ce système est composé d'éléments redondants :

- Modules de traitements
- Modules d'entrée – sortie
- Mémoire

- Le système est composé de deux cartes principales (les cartes mères).
- Chaque module peut être inséré ou extrait sans interrompre le fonctionnement du système (à chaud).

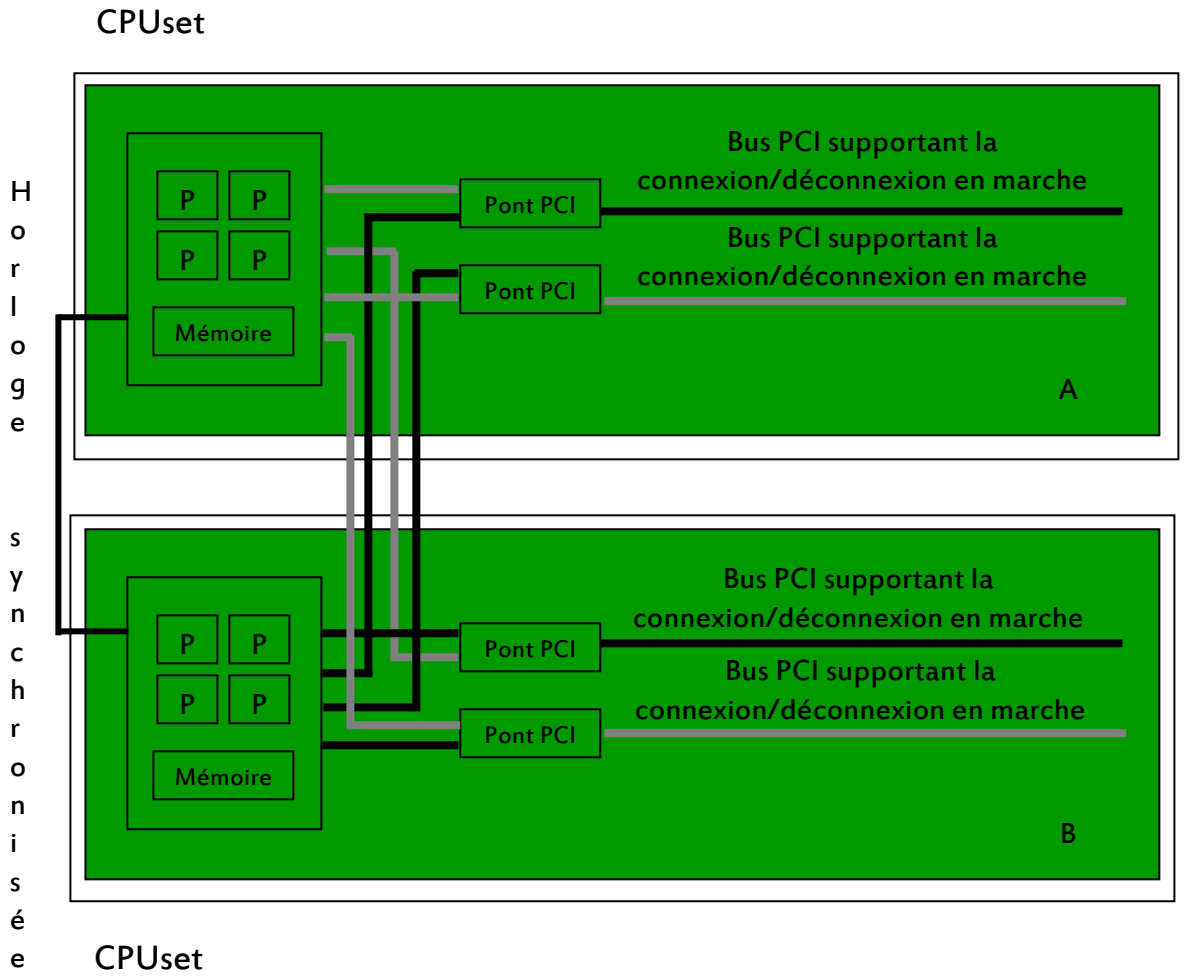
Chaque carte comporte

- un ensemble {processeurs, mémoire} appelé *CPUset*
- deux bus d'entrée-sortie PCI.
  - Ces bus supportant la connexion ou la déconnexion des contrôleurs (sans arrêt du système).
  - Un CPUset comporte de 1 à 4 processeur UltraSparc et une mémoire allant de 256 Mo à 4 Go.

Les *CPUset* peuvent être changés sans arrêter le système, de même pour les modules d'alimentation électrique.

Le principe de fonctionnement de cette architecture est de type *Spare* (rechange).

Solution matérielle: Netra FT 1800



Solution matérielle: Netra FT 1800

- Les processeurs des deux cartes CPUsets sont synchronisés par une horloge commune
  - mais ils ne fonctionnent pas avec une logique de comparaison à chaque étape.

- Chacun des CPUsets exécute le même ensemble de processus de façon telle que les deux processeurs correspondant d'un CPUset exécutent le même processus au même moment.
  - Il n'y a pas de comparaison, à chaque cycle des résultats produits par chacun des processeurs.

- La comparaison a lieu lorsque le processus fait une demande d'entrée-sortie.
  - En cas de discordance des demandes émises par les processeurs (qui exécutent le même processus) une erreur est dénoncée.

- Dans ce cas, le système entre dans une phase de test et de diagnostic afin de déterminer lequel des deux CPUset est défaillant.

- Le CPUset défaillant est alors écarté et le système continue à fonctionner avec un seul CPUset.

- Le temps de recouvrement annoncé pour une défaillance processeur est de l'ordre de 200 ms.

- Pour la réintégration d'un nouveau CPUset dans le système, la resynchronisation s'opère en tâche de fond en parallèle avec le fonctionnement du système.

Solution matérielle: Netra FT 1800

Le sous-système d'entrée-sortie est fondé sur le bus PCI et les contrôleurs peuvent être échangés lorsque le système est en fonctionnement.

Un contrôle de validité des écritures en mémoire est réalisé par le matériel.

Le matériel fonctionne sous le contrôle du système d'exploitation Solaris.

Solutions logicielles

Les solutions logicielles reposent sur le principe du point de reprise :

- Mémorisation de états du système à un instant donné.
- Journal permettant de refaire toutes les modifications du système depuis cet instant.

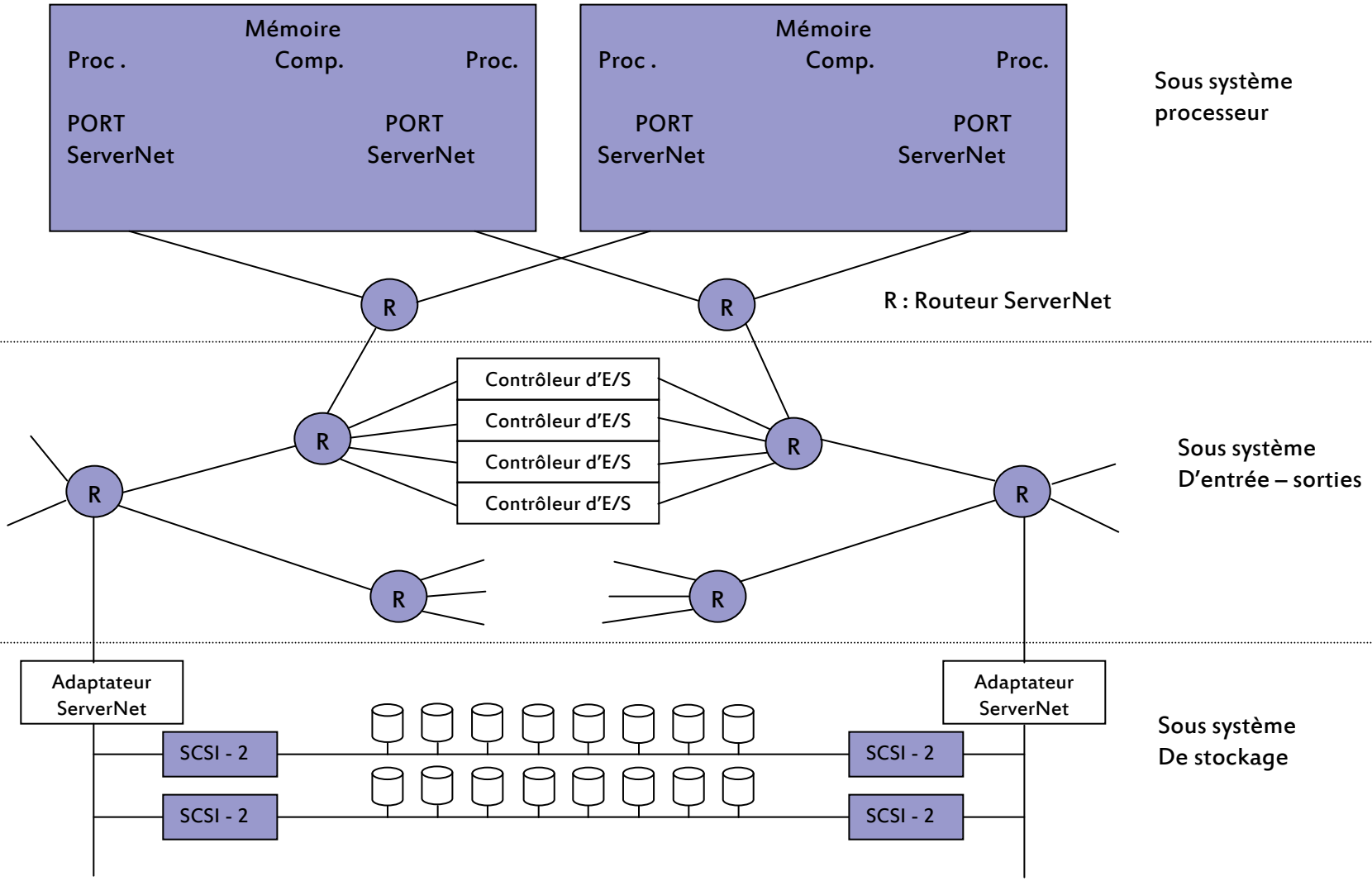
La définition de tels points est naturelle dans les systèmes transactionnels, ces systèmes sont fondés sur :

- Un système de gestion de la transaction.
- Un gestionnaire de données.

En cas de défaillance, le système transactionnel se sert de journaux pour annuler les effets des transactions qui n'ont pas encore été validées au moment de la défaillance.

Solutions logicielles: NSK

Le système NonStop Himalaya S7000 de Tandem



Solutions logicielles: NSK

• Ce système est basé sur le réseau d'interconnexion appelé SAN autrefois (System Area Network (ne pas confondre avec les *Storage Area Network*) et aujourd'hui ServerNet.

• Chaque carte dispose de deux processeurs qui exécutent le même processus de manière synchrone et le résultat est comparé.

• L'architecture du système est de type *Share Nothing*, le système étant composé de nœuds indépendants.

- Chaque nœud possède ses deux processeurs, sa mémoire, sa propre copie du système d'exploitation et un ensemble de propres périphériques.
- Les nœuds communiquent à travers le réseau ServerNet.

Solutions logicielles: NSK

- Le système d'exploitation NonStopKernel (NSK) est propre à Tandem.
  - Ce système possède des interfaces de type Unix.
  - Cette architecture fournit une image unique du système d'exploitation.

- Le système supporte un gestionnaire de base de données spécifique NonStop SQL/MX qui donne aux utilisateurs une vision unique des bases de données.
  - Celles-ci sont réparties sur les différents nœuds.
  - Les instances de NonStop SQL/MX assurent la synchronisation de leurs opérations.

Solutions logicielles: NSK

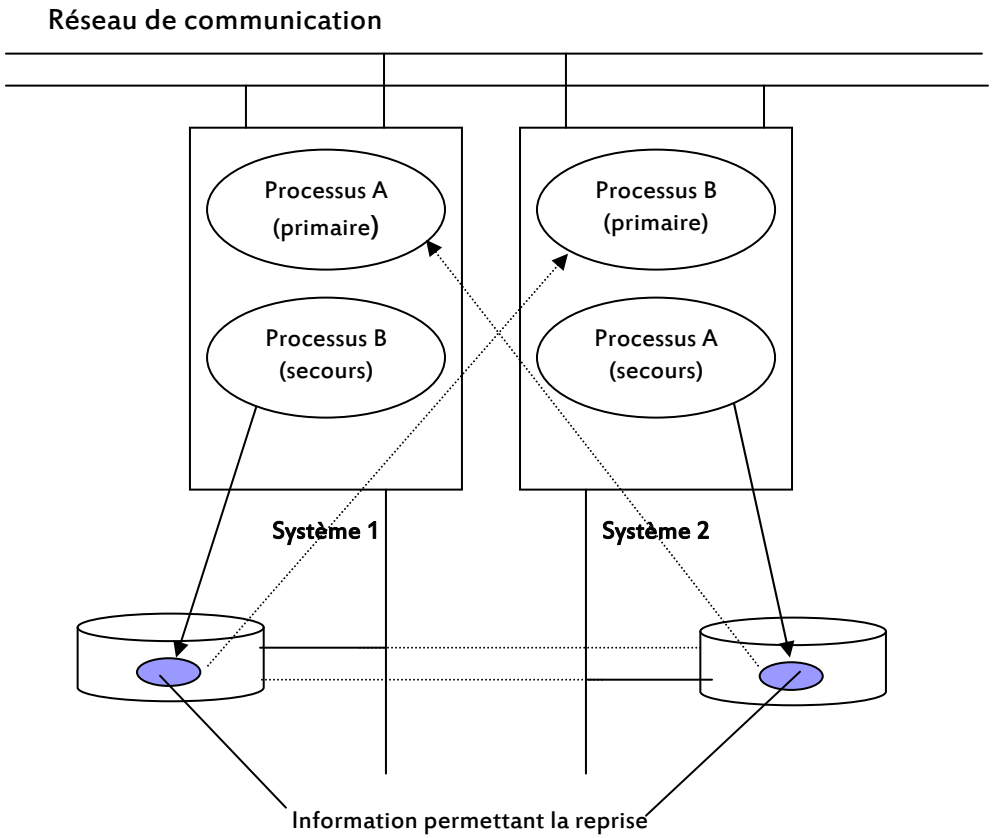
- Les utilisateurs disposent donc d'un système à croissance modulaire
  - l'ajout d'un nœud augmente a capacité de traitement ainsi que la connectivité vers les entrée-sorties.
  - NonStop SQL/MX met aussi à profit le parallélisme pour le traitement de requêtes complexes en décomposant une requête en sous requête, dont l'exécution est répartie sur les nœuds qui sont en charge de la gestion des données concernées.

- La carte unité de traitement possède deux microprocesseurs fonctionnant de façon synchrone liés par une logique de comparaison.
  - Lorsque la logique de comparaison détecte une divergence entre les résultats produits par ces deux microprocesseurs, le système d'exploitation déclare la carte défaillante et écarte immédiatement cette carte de la configuration.
  - Cet événement est enregistré dans un journal et des tests sont exécutés sur la carte de façon à déterminer la nature de la défaillance pour décider de son remplacement éventuel.
  - Les différents éléments du système (cartes, périphériques, modules d'alimentation, ventilateurs etc) peuvent être changés sans devoir interrompre le système.

- La continuité de services est fondé sur les concepts de processus primaire et de processus de secours.

Solutions logicielles: NSK

Concepts de processus primaire et de processus de secours dans le Tandem NSK :



Solutions logicielles: NSK

Les informations permettant la reprise du processus primaire (par exemple au processus A sur le système 1 et au processus B sur le système 2) sont enregistrées sur une mémoire stable (disque avec technologie RAID) via des messages envoyées sur le réseau de communication.

Lorsqu'une défaillance se produit, cela provoque l'arrêt du processus primaire, le processus de secours est activé et reprend l'exécution à partir des informations enregistrées. Le processus de secours devient alors processus primaire et il est possible de lancer un nouveau processus de secours sur un autre nœud du système.
Les bornes de transaction (begin work et commit work) sont des points auxquels les informations nécessaires à la reprise peuvent être sauvegardées.

Les différents nœuds se surveillent au moyen de la technique du *heartbeat*.

Une défaillance d'un serveur au niveau matériel ou logiciel provoque un arrêt des applications qui s'exécutaient sur ce serveur. Les applications peuvent être automatiquement réactivées sur le serveur de secours, la reprise, en ce qui concerne l'état des fichiers manipulés, étant à la charge de l'application (sauf pour les applications transactionnelles).

IBM HACMP/Bull Power Cluster

Les composants sont :

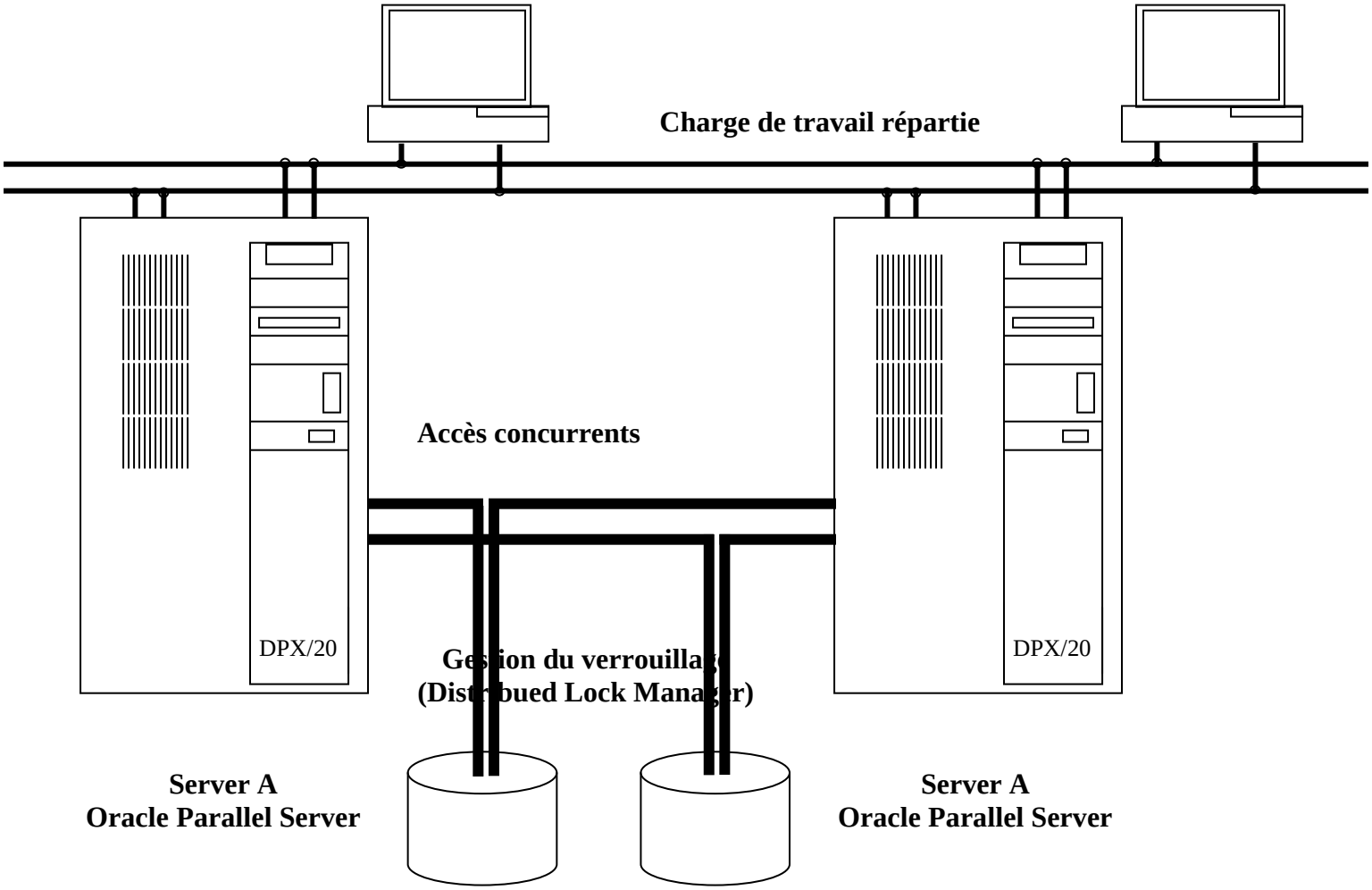
⇨ Le gestionnaire du cluster (Cluster Management) actif sur tous les nœuds du cluster. Il se charge de maintenir à jour la configuration du cluster, l'état du réseau d'interconnexion et il reflète l'état de fonctionnement du cluster vis-à-vis des autres systèmes.

⇨ CLINFO (Cluster Information Services) est optionnel sur le cluster et sur les clients. Il informe les clients sur l'état du cluster au moyen d'un API et il communique avec les agents SNMP.

⇨ L'agent SNMP (Simple Network Management Protocol) est un standard de fait pour l'administration des systèmes distribués. Il reçoit de l'information de la part du gestionnaire du cluster et la rend accessible à travers SNMP.

⇨ Le gestionnaire de verrous (Cluster Lock Manager) fournit un service de synchronisation distribué aux différents services et applications sur le cluster (concept du DLMn Distributed Lock Manager).

IBM HACMP/Bull Power Cluster

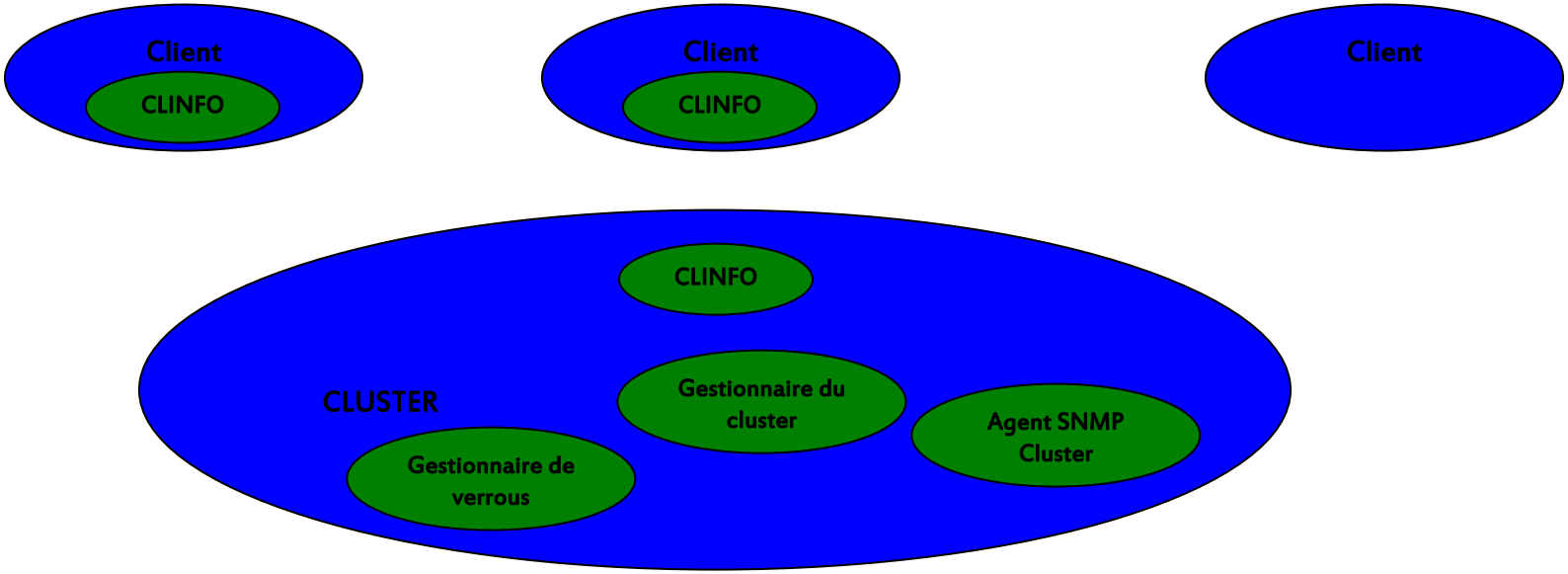


IBM HACMP/Bull Power Cluster

Ces clusters ne sont pas limités à deux nœuds.

Les ressources partagées entre les nœuds sont les unités de disque, les connexions réseau (WAN, LAN) ainsi que le gestionnaire de verrous distribué DLM.

L'interconnexion du cluster est fondé sur Ethernet ou FDDI, cette interconnexion pouvant être doublée pour augmenter la disponibilité du système, la charge de travail est répartie entre les deux serveurs .



IBM HACMP/Bull Power Cluster

Eléments fonctionnels du gestionnaire du cluster :

- **Le gestionnaire du cluster** est composé d'un contrôleur de ressources système (*System Resource Controller)* et fournit un ensemble de services facilitant la création et le contrôle des sous systèmes.

- **Le contrôleur du cluster** (*Cluster Controller*) est le composant principal du gestionnaire du cluster, il reçoit l'information sur le cluster par l'intermédiaire de la couche d'interface réseau (NL) et des modules d'interface réseau (NIM).

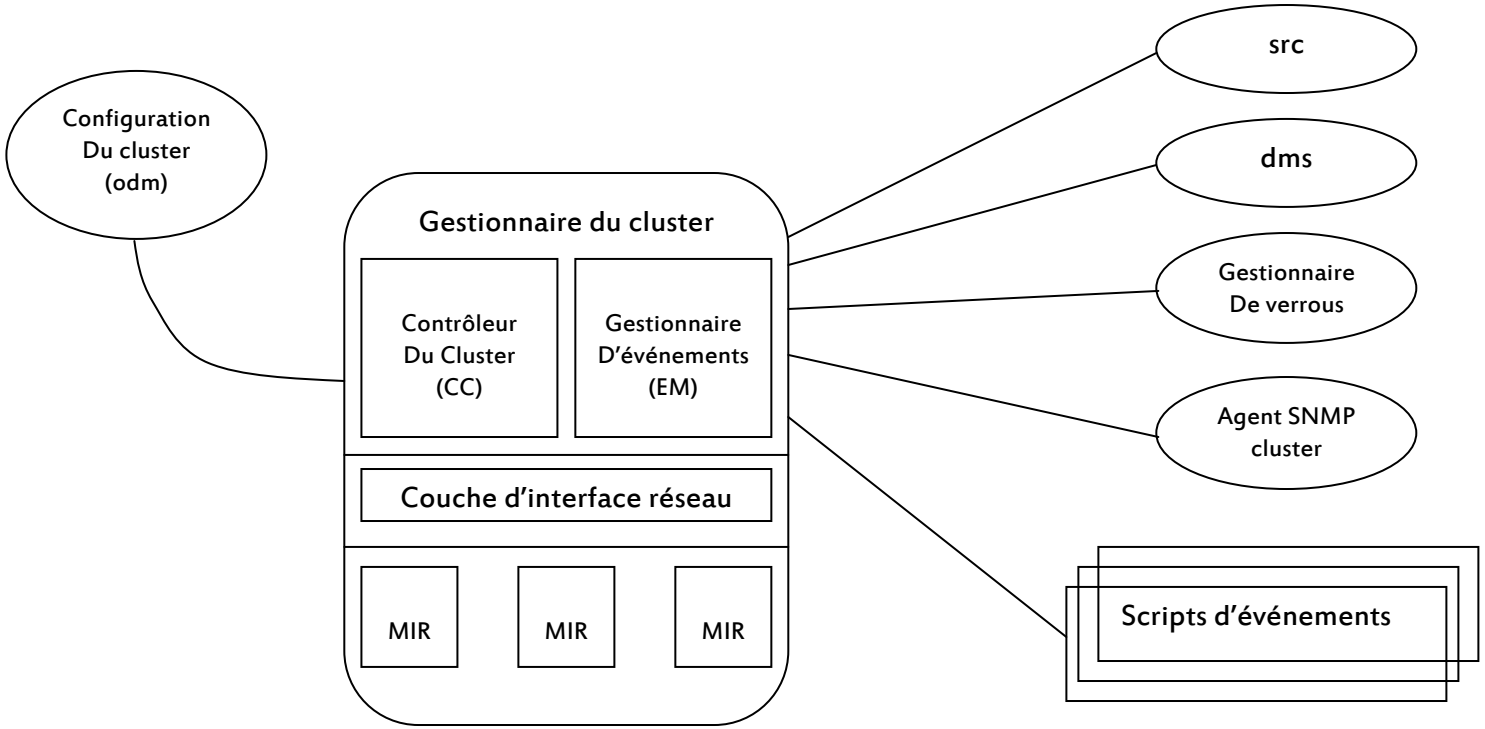
- **Le gestionnaire d'événements** (*Event Manager*) déclenche l'exécution des scripts associés aux événements et se charge de la notification des événements vers l'agent SNMP du cluster ou vers le gestionnaire de verrous.

- **La couche d'interface réseau** (*Network Interface Layer*) contrôle le fonctionnement des modules d'interface réseau, détermine la fréquence des battements de cœur et gère l'envoi des messages à des destinataires multiples.

- **Les modules d'interface réseau** (*Network Interface Modules*) émettent ou réceptionnent les messages et détectent les défaillances réseau.

IBM HACMP/Bull Power Cluster

Architecture du gestionnaire de cluster :



IBM HACMP/Bull Power Cluster

Un sous système est un ensemble de programmes ou de processus accomplissant une fonction donnée.

Le gestionnaire de cluster possède un service permettant d'éviter les étreintes fatales :

- Le service dms (Dead Man Switch ou technique de l'homme mort).

Un compteur est initialisé et mis à jour régulièrement, si sa mise à jour est arrêtée cela provoque un arrêt du système, la mise à jour étant considérée comme une preuve du bon fonctionnement.

Le contrôleur du cluster maintient à jour l'appartenance des systèmes au cluster et l'état du réseau. Il maintient aussi la configuration du cluster au moyen d'un gestionnaire de base de données objets :

- Object Databse Manager (odm).

Odm contient toutes les informations de configuration et de ressources gérées par le cluster. Il détermine aussi quels systèmes doivent échanger des signaux de surveillance (battement de cœur ou Keep Alive, KA).

IBM HACMP/Bull Power Cluster

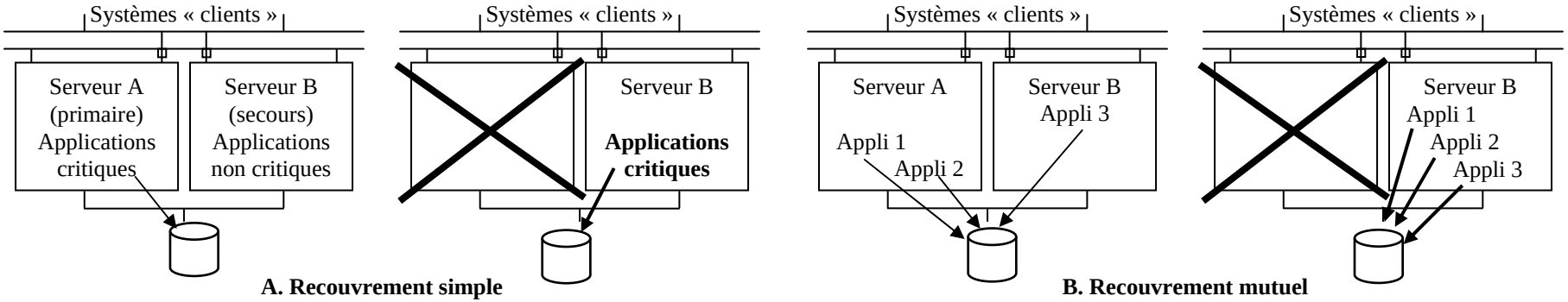
Deux types de trafic au sein d'un cluster :

- *Les échanges entre les modules d'interface réseau* (NIM). il s'agit des battements de cœur. Cela permet aux différents modules de s'assurer du bon fonctionnement des autres modules ainsi que des systèmes. La fréquence d'émission est paramétrable, ainsi que le nombre de battements qui peuvent rester sans réponse avant que le système concerné soit déclaré défaillant.

- *Les messages* nécessités par les applications.

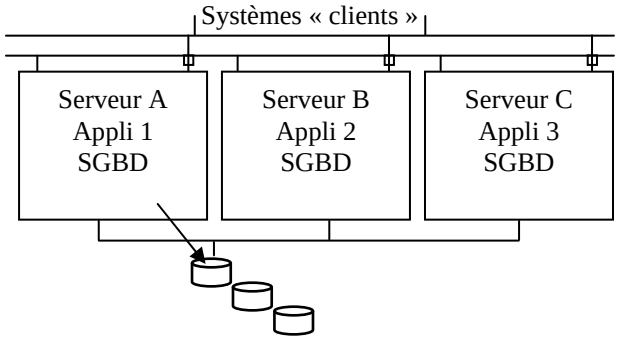
IBM HACMP/Bull Power Cluster

Les différents modes de fonctionnement du cluster :



A. Recouvrement simple

B. Recouvrement mutuel



C. Partage de charge

IBM HACMP/Bull Power Cluster

Les différents modes de fonctionnement du cluster :

Recouvrement simple et mutuel :

L'application fonctionne sur l'un des systèmes (primaire). Un autre système dit de secours est soit inactif (standby), soit exécute des tâches indépendants non critiques (backup). En cas de défaillance du système sur lequel l'application fonctionne, le système de secours va prendre le relais en suspendant éventuellement l'exécution des tâches non critiques. Lorsque le système primaire redevient opérationnel, il reprend le contrôle des ressources et l'exécution des applications.

Une variante consiste à faire passer le système des applications de secours en primaire et le défaillant une fois opérationnel devient alors celui de secours (Rotating standby).

L'ensemble des systèmes participent à l'exécution des applications, l'ensemble des applications doit être partitionné de telle sorte que les systèmes (Serveur A et Serveur B) ne partagent pas d'informations (comme par exemple des fichiers). Chaque système a donc ses propres ressources.

En cas de défaillance de l'un des systèmes, les applications qui s'exécutaient sur ce système, ainsi que les ressources (partagées) concernées sont reprises par l'autre système. Il peut y avoir plus de deux nœuds.

IBM HACMP/Bull Power Cluster

Les différents modes de fonctionnement du cluster :

Partage de charge :

Les applications sur les différents systèmes accèdent à la même base de données. En cas de défaillance de l'un des systèmes, ses applications sont reprises par les autres systèmes composant le cluster.

Chaque système composant le cluster possède une instance active du gestionnaire de la base de données. Chacun de ces gestionnaires accède en parallèle à la base de données pour le compte des applications qui s'exécutent sur le système sur lequel il réside. Les gestionnaires des bases de données se synchronisent au moyen du gestionnaire de verrous. Il faut bien sur que le gestionnaire de base de données soit conçu pour supporter ce type d'architecture comme par exemple Oracle Parallel Server (OPS).

Cette architecture présente l'avantage de la haute disponibilité, mais aussi celle de la répartition de charge et la croissance modulaire puisque chaque système participe à l'exécution de la charge globale.

Pour les utilisateurs d'un système défaillant, la défaillance n'est pas visible puisque les transactions sont reprises par un autre nœud, si ce n'est par un temps de réponse accru.

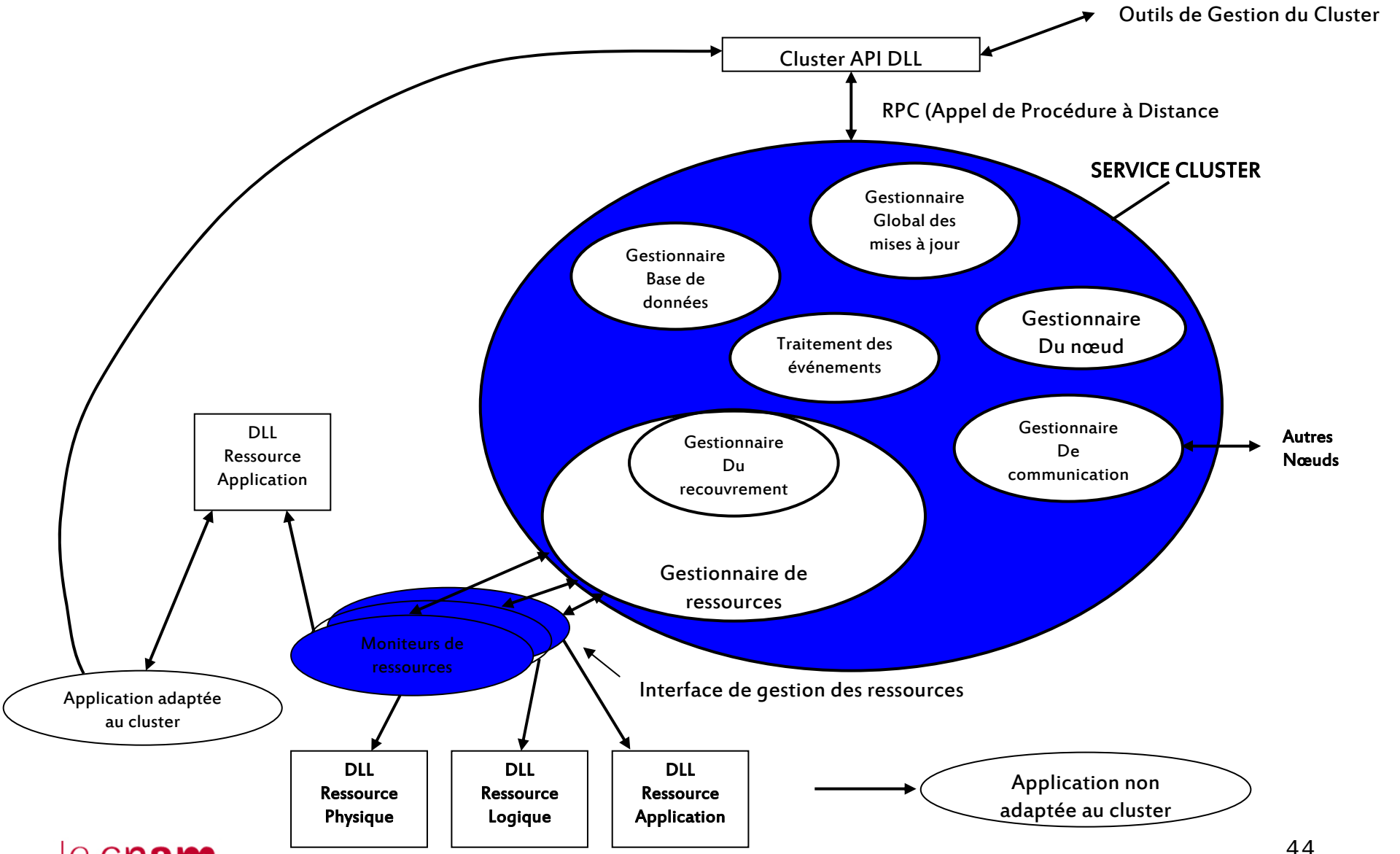
Microsoft Cluster Server

C'est une version particulière de Windows NT ou 2K appelé Microsoft Cluster Server. Ce cluster est de type Share Nothing.

Les concepts de ce cluster sont les suivants :

- **Service Cluster** (Cluster Service) est un ensemble de logiciels implémentés sur chacun des nœuds du cluster et gérant les activités spécifiques du cluster.
- **Ressource** (Resource) est une entité gérée par le service cluster. Le service cluster considère toutes les ressources comme des objets opaques (il ne s'intéresse pas aux opérations qu'il est possible de faire dessus) : disques, cartes contrôleurs de communication, adresses TCP/IP, les applications et les bases de données.
- **Ressource en ligne** est une ressource lorsqu'un nœud fournit son service sur ce nœud.
- **Groupe** (Group) est une collection de ressources gérées comme une entité unique. Habituellement, un groupe contient tous les éléments nécessaires à une application.

Microsoft Cluster Server (MCS)



Microsoft Cluster Server (MCS)

• **Le gestionnaire du nœud** (Node Manager) gère l'ensemble du nœud au cluster et surveille l'état de bon fonctionnement des autres nœuds du cluster. Cette surveillance s'opère au moyen de la technique des battements de cœur. En cas de défaillance de l'un des membres du cluster, chacun de nœuds actifs du cluster procède à une vérification du cluster de façon à le reconfigurer.

• **Le gestionnaire de la base de données** (Database Manager) maintient la base de données représentant la configuration du cluster. Les entités représentés dans cette base de données sont le cluster lui-même, les nœuds, les types de ressources, les groupes et les ressources proprement dites. Les gestionnaires des différents nœuds coopèrent au moyen d'un protocole pour maintenir un état cohérent de ces bases de données répliquées sur chacun des nœuds.

•**Le gestionnaire de ressources / gestionnaire de recouvrement** (Resource Manager / Failover Manager) gère les ressources, les groupes et lancent les actions appropriées telles que démarrage, redémarrage et recouvrement.

•**Le traitement d'événements** (Event Processor) assure la connexion entre les composants du Service Cluster et prend en charge les opérations communes.

•**Le gestionnaire de communication** (Communication Manager) prend en charge la communication avec les autres nœuds du cluster.

•**Les gestionnaire global des mises à jour** (Global Update Manager) prend en charge les fonctions de mises à jour pour les composants du Service Cluster.

Microsoft Cluster Server (MCS)

• **Les moniteurs de ressources** (Resource Monitor) sont implémentés sous forme de processus et communiquent avec le service cluster au moyen d'un appel de procédure à distance (RPC). Ils s'assurent de l'état du bon fonctionnement des ressources et assurent une transformation entre les appels génériques et les appels spécifiques de chacune des ressources.
• **Le service temps** (Time Service) maintient un temps homogène à l'intérieur du cluster. Il est implémenté sous forme de ressource.

• Ces deux composants ne font pas réellement partie du Service Cluster.

• Les ressources sont implémentées sous forme de bibliothèques liées dynamiquement (DLL), chargées dans l'espace d'adressage des gestionnaires de ressources. A chaque ressource est associée une politique locale de redémarrage définissant les actions à exécuter sur la ressource ne peut plus continuer à s'exécuter sur le nœud courant.

• Cluster Server fournit en standard des DLL pour les ressources telles que les volumes physiques, les volumes logiques (un ou plusieurs disques physiques), les fichiers, les imprimantes partagées, les noms et adresses réseau, les applications ou services génériques et le service internet.

Microsoft Cluster Server (MCS)

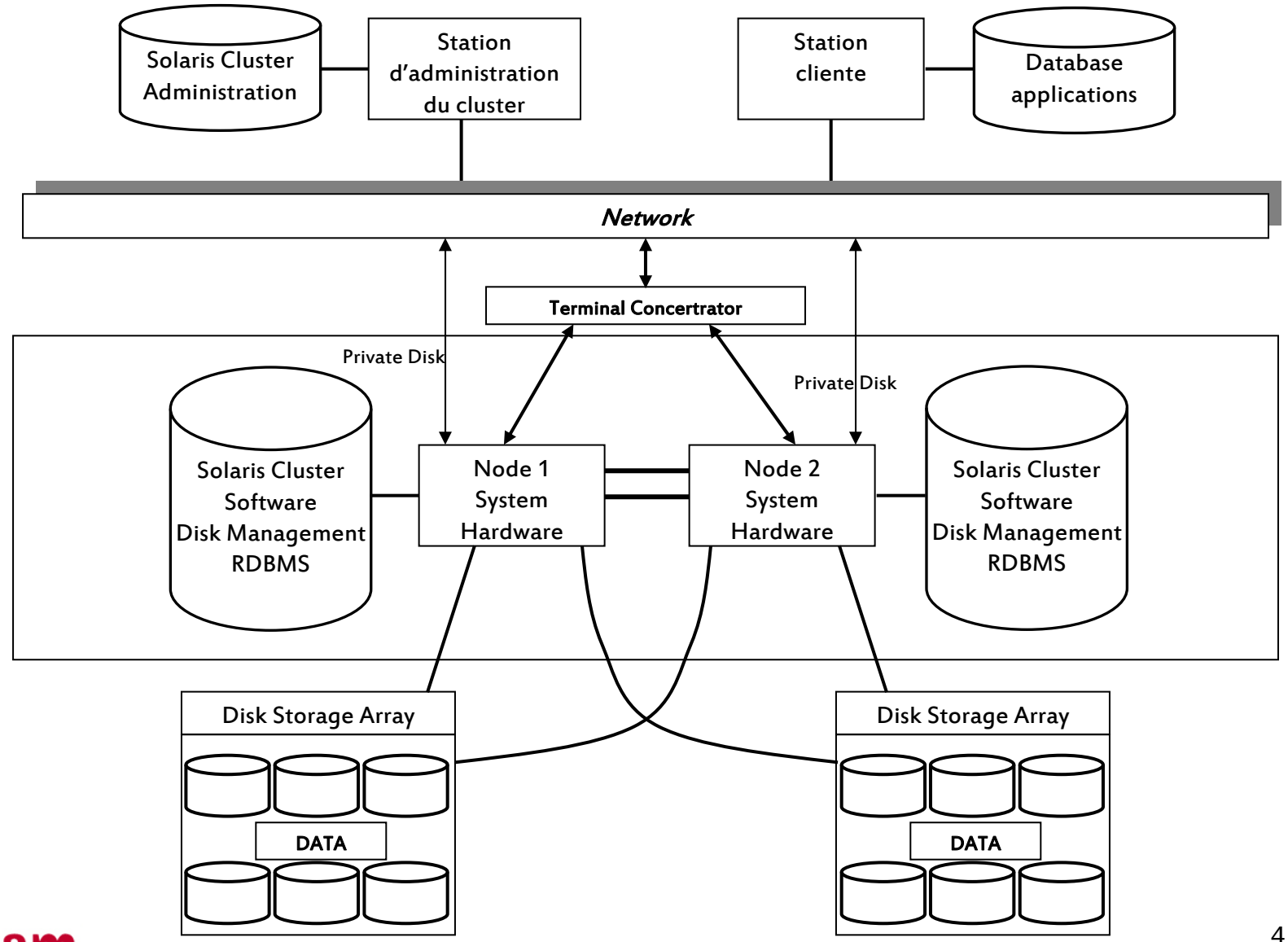
Suite à la défaillance d'une ressource, le gestionnaire de la ressource peut décider de redémarrer la ressource ou de la mettre hors ligne, de même pour les ressources qui en dépendent. Dans ce cas, il indique au gestionnaire de recouvrement que la ressource doit être redémarrée sur un autre système du cluster. Cette action peut être déclenchée par l'administrateur.

Lorsqu'un système défaille complètement, les groupes de ressources qu'il supportait doivent être retirés de ce système et installés sur un ou plusieurs autres systèmes. Les systèmes restant négocie entre eux pour déterminer lequel d'entre eux supportera le groupes de ressources du système défaillant. Une fois le destinataire du groupe de ressources déterminé, les bases de configuration sont mises à jour.

Lorsqu'un système redevient actif, le gestionnaire de recouvrement peut décider de déplacer des groupes de ressources vers ce système suite à une déclaration de préférence associée au groupe de ressources.

La création d'un cluster débute avec l'exécution d'un logiciel d'initialisation sur un système qui devient alors le premier membre du cluster. Cette procédure d'initialisation doit être exécutée sur tous les autres membres du cluster, mais ils doivent préciser le nom du cluster qui veulent rejoindre. Le temps (théorique) de basculement est de l'ordre de la minute.

Sun Entreprise Cluster



Sun Entreprise Cluster

Cette solution de cluster repose :

- Un terminal concentrateur, il offre aux clients du cluster la vision unique d'un système connecté au réseau par une adresse IP unique. Cette adresse est mappée sur le nœud actif.
- Chaque nœud peut aussi accéder individuellement au réseau par l'adresse IP propre au système.
- Un réseau interne au cluster permet au software gérant le cluster de surveiller les autres nœuds.
- En cas de défaillance, le cluster bascule sur le nœud de secours.
- Les applications et les données manipulées par le cluster se trouve sur des disques partagés entre les nœuds. De ce point de vue, il est nécessaire que la baie de disques puisse être partagée entre plusieurs systèmes. Le système propriétaire d'une baie est le système actif dans le cluster. En cas de basculement, le système qui devient actif prend la propriété de la baie de disques partagée.
- Chaque système est un système « classique » de SUN, dispose de son propre disque système sur lequel a été ajouté le software gérant le cluster : Cluster Software Distribution.