Systèmes à haute disponibilité : Définitions et solutions

Octobre 2001

René J. Chevance

Contenu

L'objectif de cette présentation est d'introduire les définitions propres aux systèmes à haute disponibilité et de présenter ensuite des solutions utilisées dans le cadre de serveurs pour des systèmes d'information

- Définitions
  - Sûreté de fonctionnement
  - Coût de l'indisponibilité
  - Classification des systèmes
  - Concepts et terminologie
  - Modes de défaillance
  - Grandeurs caractéristiques
  - Analyse des causes des défaillances
  - Principes de conception
- Solutions
  - Solutions au niveau du matériel
  - Solutions au niveau du logiciel
- Recouvrement après catastrophe
- Estimation de la disponibilité des systèmes
- Comparaison des solutions
- Perspectives

Définitions

Sûreté de fonctionnement

- **Notion de sûreté de fonctionnement**
  - Propriété qui permet à un utilisateur de placer une confiance justifiée dans le service que le système délivre
  - C'est la propriété et les caractéristiques d'une entité ayant rapport au temps qui lui confèrent l'aptitude à satisfaire les besoins exprimés ou implicites pour un intervalle de temps donné et des conditions d'emploi fixées (X50-125, norme de management de la qualité et de l'assurance de la qualité, standard ISO 8402).
  - C'est l'ensemble des aptitudes d'un produit lui permettant de disposer des performances fonctionnelles spécifiées, au moment voulu, pendant la durée prévue, sans dommage pour lui-même et pour son environnement

Définitions(2)

- **Propriétés liées à la sûreté de fonctionnement**
  - **Fiabilité (*reliability*). Correspond à la continuité du service rendu.**
  - **Disponibilité (*availability*). Correspond à l'aptitude du système à être prêt à rendre le service pour lequel il a été conçu.**
  - **Maintenabilité (*maintenability*). C'est l'aptitude d'un système à être maintenu en condition opérationnelle.**
  - **Innocuité (*safety*), ou encore sécurité vis-à-vis de l'environnement.**
  - **Immunité (*immunity*). Correspond à la résistance d'un système aux agressions externes.**
    Pour les systèmes informatiques, l'immunité correspond aux 3 critères suivants : *disponibilité* du système, *intégrité* et *confidentialité* des données.

Coût de l'indisponibilité

- **Coût moyen d'une heure d'indisponibilité du système (Source Contingency Planning Research)**

Application	Secteur d'activité	Coût de l'indisponibilité
Courtage	Finance	$6,45 millions
Ventes par carte de paiement	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

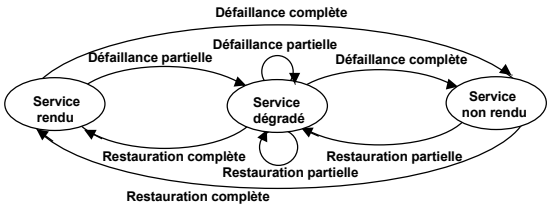
Classification des systèmes

- D'après [GRA91], classification des systèmes en fonction de leur disponibilité
  - Hypothèse 7 jours sur 7, 24 heures sur 24 (7x7, 24x24)
  - Attention, la terminologie peut varier, il convient de s'en tenir aux chiffres

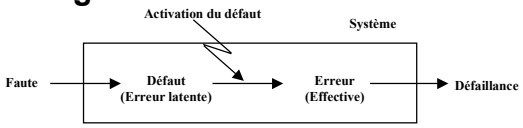
Type de système	Indisponibilité (minutes par an)	Disponibilité (%)	Classe de disponibilité
Non géré (Unmanaged)	50000	90	1
Géré (Managed)	5000	99	2
Bien géré (Well Managed)	500	99,9	3
Tolérant les fautes (Fault-tolerant)	50	99,99	4
Haute disponibilité (High Availability)	5	99,999	5
Très haute disponibilité (Very High Availability)	0,5	99,9999	6
Ultra haute disponibilité (Ultra High Availability)	0,05	99,99999	7

Concepts et terminologie

- Concept de service (Source Bull)



- Terminologie



  - Types de défauts
    - Bhorbug : un ré-essai conduit à une erreur
    - Heisenbug : un ré-essai ne conduit pas systématiquement à une erreur

Modes de défaillance

- **Défaillance = non conformité de la réponse d'un système par rapport à ses spécifications. La réponse d 'un système : est fonction de l'état du système, doit être fournie dans un intervalle de temps, le système doit se trouver dans un état spécifié, les valeurs fournies doivent correspondre aux spécifications.**
- **Modes de défaillance**
  - **Défaillance par omission (Omission Failure)**
  - **Défaillance temporelle (Timing Failure)**
  - **Défaillance de réponse (Response Failure)**
    - **Valeur fournie (Value Failure)**
    - **État résultant (StateTransition Failure)**
  - **Défaillance de type «plantage» (Crash Failure)**

Page 9

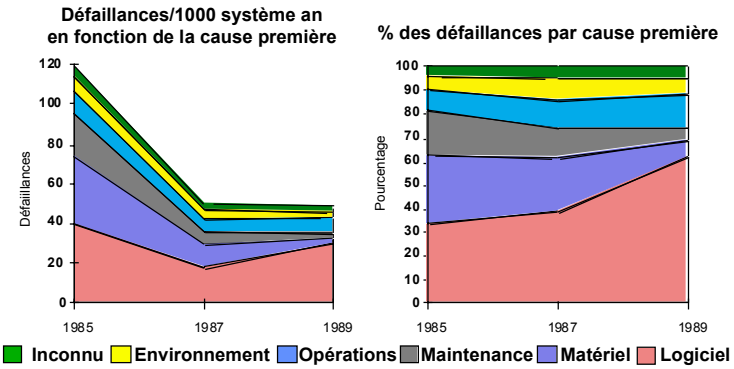
Grandeurs caractéristiques

- **Il s'agit de grandeurs accessibles et mesurables (comme toute mesure, elles doivent vérifier les propriétés suivantes : représentativité, interprétables, fidèles, précises et utiles).**
- **Mesures liées au concept de défaillance**
  - **MTTF, ou Temps moyen jusqu'à défaillance** (*Mean Time To Failure*)
  - **MTBF, ou Temps moyen entre défaillances** (*Mean Time Between Failures*)
  - **Note : si pas de redondance MTBF = MTTF + MTTRes**
- **Mesures liées au concept de maintenabilité**
  - **MTTRes, ou Temps moyen jusqu'à restauration (*Mean Time To Restore* ou *Mean Time to Recover*)**
  - **MTTRep, ou Temps moyen jusqu'à réparation (élément) (*Mean Time To Repair element*)**
- **Mesure liée au concept de disponibilité**
  - **At = MTTF/(MTTF+MTTRes)**

Page 10

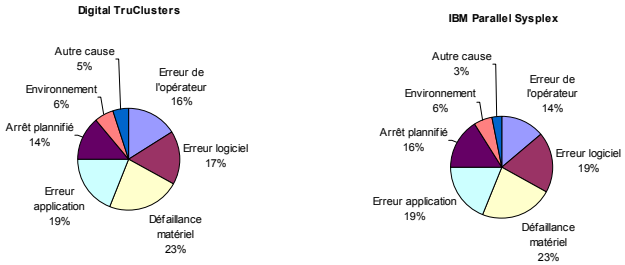
Analyse des causes des défaillances

- Peu de données publiées. Tandem a publié, à deux reprises (1985 et 1989) [GRA90], des données observées sur ses systèmes en clientèle.

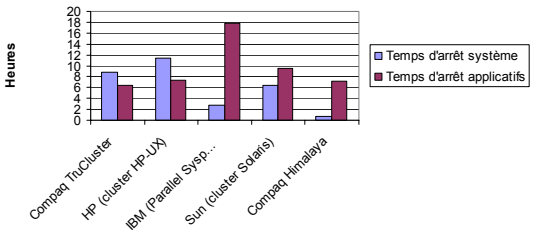


Analyse des défaillances(2)

- Analyse des causes de défaillances pour des systèmes d'entreprise (Source : Standish Group 2000)

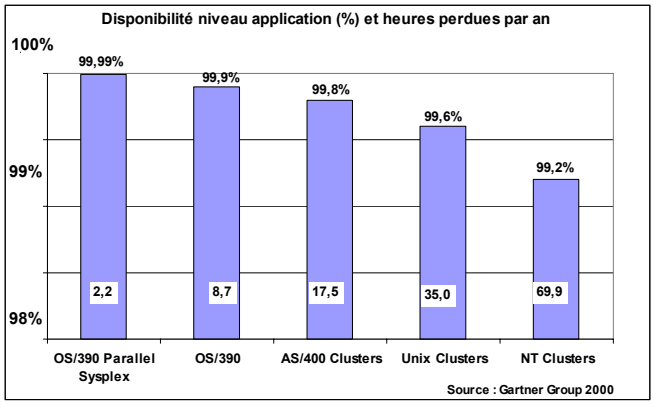


Temps d'arrêts annuels d'exploitation



Analyse des défaillances(3)

- **Disponibilité au niveau application et heures perdues par an (Source : Gartner Group 2000)**



Remarque : Ces chiffres ne sont pas cohérents avec ceux du Standish Group

Principes de conception

- **Concept d'état. Un système à haute disponibilité doit masquer les défaillances auxquelles il est sujet à son environnement. Deux possibilités :**
  - **Le système maintient en permanence plusieurs versions de l'état courant**
  - **Le système maintient des états à partir desquels le traitement peut être repris**
- **Principes de conception**
  - **Modularité**
  - **Fail Fast : fonctionnement correct ou arrêt immédiat**
    - **Fail Safe : les conséquences des défaillances sont bénignes**
    - **Fail Silent : les défaillances sont du type «plantage»**
    - **Fail Stop : les défaillances se traduisent par un arrêt**
  - **Indépendance des modes de défaillance**
  - **Redondance et réparation**
  - **Élimination des points de défaillance uniques**

Principes de conception(2)

- **Mise en œuvre des principes de conception**

Concept	Matériel	Logiciel
Modularité	• Classique	• Classique
Fail Fast	• Autovérification • Comparaison	• Autovérification • Programmation en *N* versions
Indépendance des modes de défaillance	• Mode de couplage entre les modules	• Mode de couplage entre les modules
Redondance et réparation	• Modules de rechange approche *N*+1 • Échange de modules en fonctionnement	• Redondance de certains modules logiciel • Remplacement du logiciel en marche
Élimination des point de défaillance uniques	• Redondance	• Redondance

Solutions

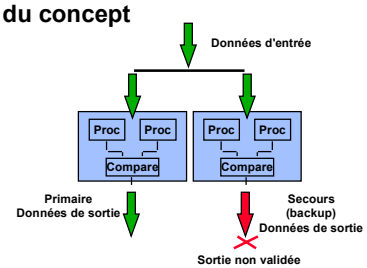
Notes :

- ***Les solutions proposées sont conçues pour résister à une défaillance.***
- ***Pour la commodité de l 'exposé, on a séparé les solutions en deux familles : solutions au niveau du matériel et solutions au niveau du logiciel. De fait, une solution est une combinaison des deux approches.***

Solutions au niveau du matériel

Stratus

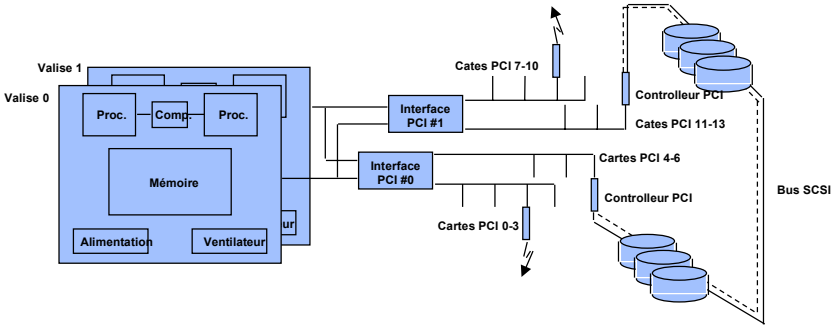
- «Pair and Spare» de Stratus
  - Illustration du concept



  - Principe de fonctionnement
    - Appariement (Pair) des organes actifs : le même processus est exécuté en parallèle par deux processeurs au sein d'un même organe
    - Doublement (Spare) des organes actifs : les deux organes exécutent le même processus, l'un des deux est dit organe primaire. Seules les sorties de l'organe primaire sont validées. En cas de défaillance de l'organe primaire, l'organe de secours prend le relais et ses sorties sont validées.
    - Les différents organes peuvent être échangés sans interrompre le fonctionnement du système.

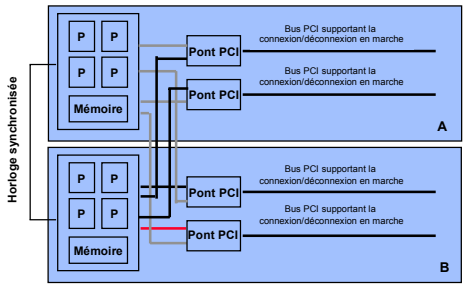
Stratus(2)

- Architecture Continuum 4000 de Stratus
  - Fondée sur processeurs HP PA 8000
  - Deux systèmes d 'exploitation supportés :
    - VOS Virtual Operating System (Stratus)
    - HP-UX (Unix de HP adapté à PA)
  - Prochaine génération à base d'IA-64



Sun NETRAft

- NETRAft 1800 de Sun Microsystems
  - Architecture du système



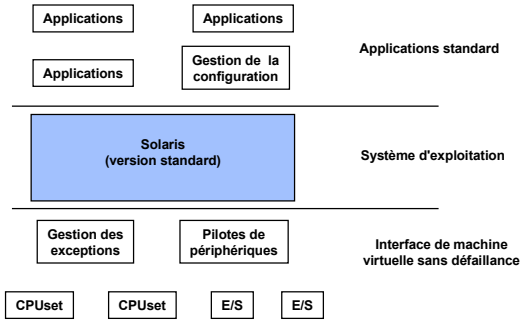
CPUset (pouvant être changé pendant la marche du système)

CPUset (pouvant être changé pendant la marche du système)

  - Principe de fonctionnement de type «Spare» (rechange)
    - Structure multiprocesseur (jusqu'à 4)
    - Processeurs fonctionnant de façon synchrone (horloge commune) mais sans vérification à chaque pas
    - Les CPUsets exécutent le même ensemble de processus
    - Comparaison de l'état lorsqu'un processus fait une demande d 'entrée-sortie. Si divergence, le système entame une phase de diagnostique pour déterminer quel est le processeur défaillant
    - Temps de recouvrement annoncé : 200 ms

Sun NETRAft(2)

- NETRAft 1800 de Sun Microsystems
  - Architecture du logiciel (Solaris)



Page 21

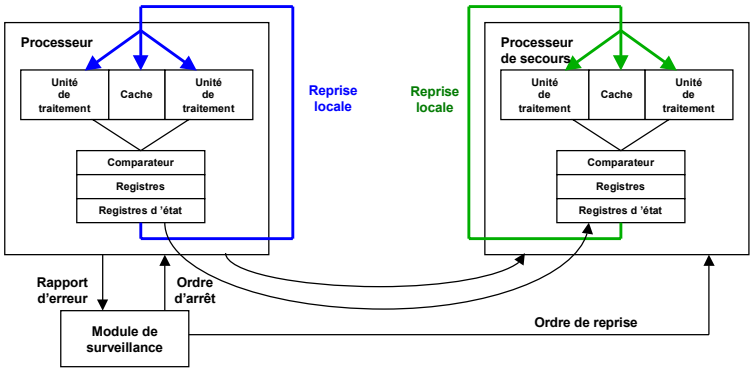
IBM - Haute disponibilité

- Haute disponibilité (hors clustering intra ou inter-serveur par Sysplex) :
  - En l 'absence de données précises sur le z900, cet exposé est fondé sur les caractéristiques de la génération précédente G5
  - Concept de processeur de secours susceptible de venir remplacer un processeur défaillant
  - Échange, en fonctionnement, des composants défaillants :
    - Matériel
    - Microcode
    - Logiciel?
  - Stratégie N+1 pour les modules d 'alimentation électrique et les ventilateurs
  - Reprise au niveau instruction en cas de défaillance du processeur (voir ci-après)

Page 22

IBM - Haute disponibilité(2)

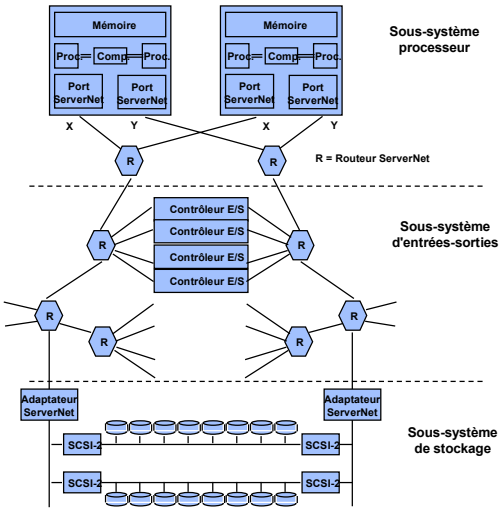
- Modèle d'exécution (S/390 G5 et G6) fondé sur :
  - La notion de point de reprise au niveau instruction
  - Le doublement, au sein même de l'unité de traitement, de la fonction d'exécution avec comparaison pas à pas. En cas de divergence possibilité de reprise :
    - Sur le même processeur (ré-essai)
    - Sur un processeur de secours (reprise)



Solutions au niveau du logiciel

Tandem

- NonStop Himalaya de Tandem
  - Architecture du matériel

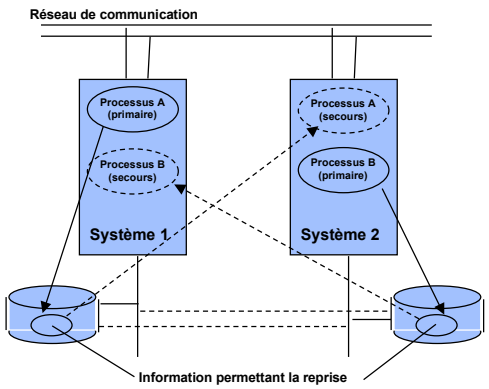


Principe de fonctionnement :

- Architecture matérielle spécifique
- Logique de type «pair», comparaison, cycle à cycle, des sorties des processeurs;
- En cas de défaillance, pas de fonction «spare», c'est un mécanisme de reprise logiciel fondé sur la technique des points de reprise qui assure la poursuite des activités (voir structure du logiciel sur la page suivante)
- Les points de reprise sont déterminés par les bornes des transactions (cette solution n'est applicables qu'aux applications transactionnelles)
- Les différents sous-ensembles composant le matériel peuvent être échangés sans interrompre le fonctionnement du système.

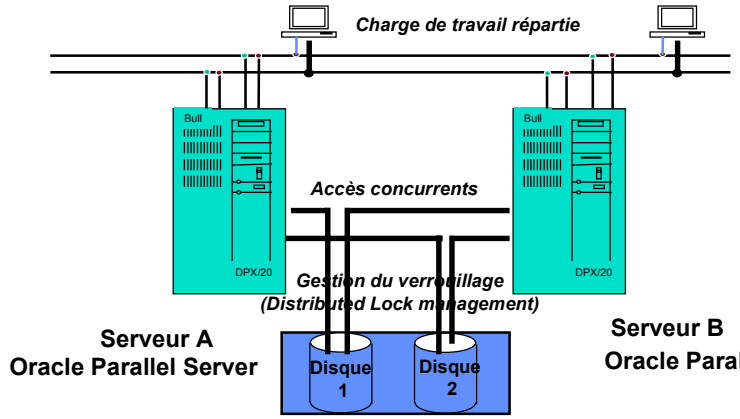
Tandem(2)

- NonStop Himalaya de Tandem
  - Architecture du logiciel - Principe du processus de secours (backup)



Solutions de type cluster (Unix et NT)

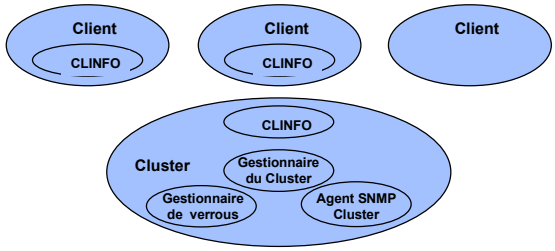
- Cluster IBM RS/6000 AIX HACMP (High Availability Cluster MultiProcessing)
  - Exemple de configuration



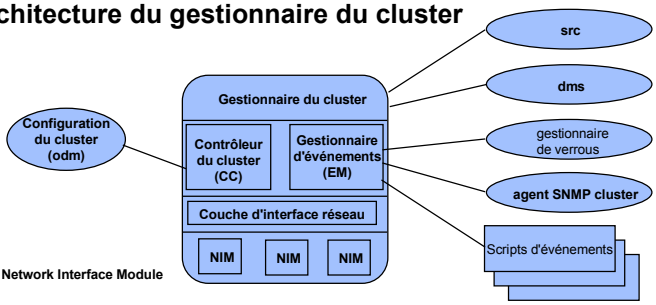
Page 27

Clusters - HACMP

- Architecture générale HACMP



- Architecture du gestionnaire du cluster



Page 28

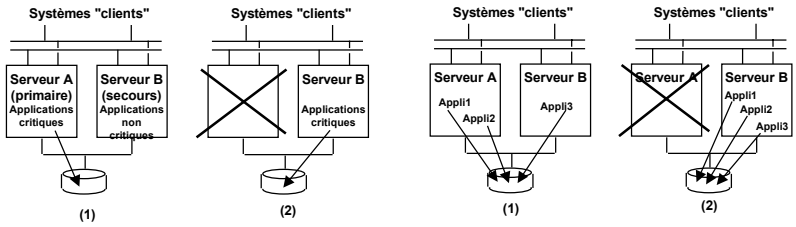
NIM : Network Interface Module

Clusters - HACMP(2)

- **Composants :**
  - **CLINFO (Cluster Information Services) informe sur l 'état du cluster (au moyen d 'une API)**
  - **Agent SNMP rend l 'information fournie par le gestionnaire du cluster accessible via SNMP**
  - **Gestionnaire de verrous (Cluster Lock Manager) : service de synchronisation distribué**
  - **Gestionnaire du cluster : ensemble de services facilitant le contrôle et la création des sous-systèmes**
    - **Contrôleur du cluster (Cluster Controller) reçoit l 'information sur le cluster (via NIL et NIM) et est en relation avec src (System Resource Controller). Il gère la base de données représentant les objets du système (odm Object Data base Manager)**
    - **Gestionnaire d'événements (Event Manager)**
    - **Couche d'interface réseau (Network Interface Layer - NIL)**
    - **Modules d'interface réseau (Network Interface Modules - NIM)**
  - **Surveillance mutuelle par la technique du «Dead Man Switch» ou dms et des battements de cœur (Keep Alive)**

Clusters - HACMP(3)

Modes de fonctionnement



A) - Recouvrement simple (Hot Standby ou Simple Failover)

B) - Recouvrement mutuel (Mutual Takeover)

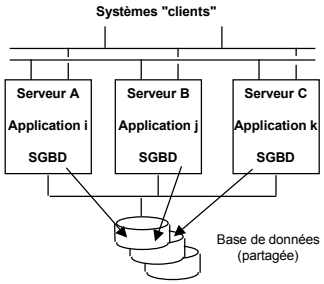
Hot Standby ou Simple Failover: l'application fonctionne sur l'un des systèmes (primaire), un autre système est inactif (standby) ou bien exécute des tâches indépendantes non-critiques (backup). En cas de défaillance du système sur lequel l'application fonctionne, le système de secours "prend" le relais 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 critiques.
Rotating Standby: le rôle du système primaire et du système de secours ne sont pas figés, lorsque que le système primaire redevient opérationnel, il se comporte comme système de secours.

•**Mutual Takeover ou Partionned Workload** Configuration: l'ensemble des systèmes participent à l'exécution des applications. Chacun des systèmes opère sur des ressources différentes (c'est-à-dire que les applications fonctionnant sur chacun des systèmes sont indépendantes et que les ensembles de fichiers qu'elles manipulent sont disjoints). En cas de défaillance de l'un des systèmes, les applications qui s'exécutaient sur ce système sont reprises par l 'autre système. La configuration présentée ici peut s'étendre à plusieurs noeuds.

Remarque Importante : Tous ces modes de fonctionnement n'apportent aucune garantie quant à l'état des fichiers suite à une défaillance. Un système de fichiers journalisé (tel JFS d 'AIX) permet de maintenir l'intégrité du conteneur mais pas celle du contenu. Cette intégrité peut être assurée par des gestionnaires de données dans le cadre de transactions.

Clusters - HACMP(4)

Modes de fonctionnement(2)



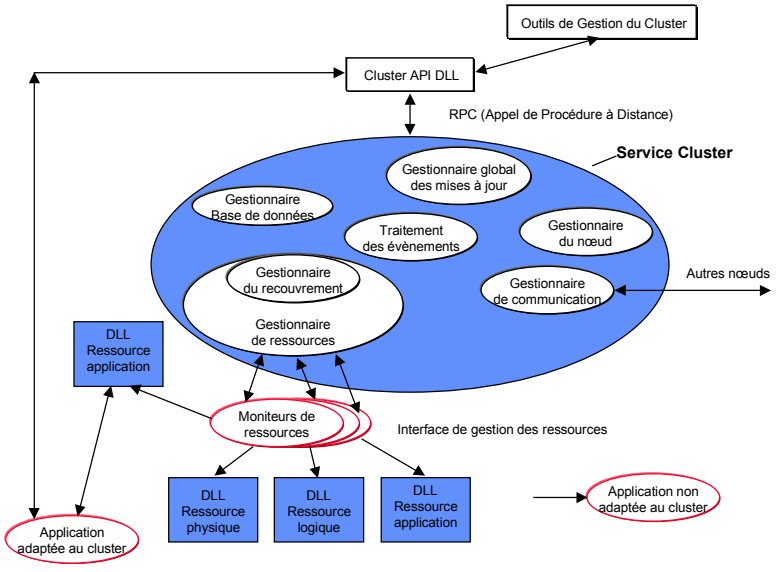
C) - Partage de charge
(Cluster MultiProcessing)

•**Cluster MultiProcessing (CMP)**: les applications fonctionnant 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.
• Les applications se synchronisent au moyen du Distributed Lock Manager.
• Outre la fonctionnalité de haute disponibilité, ce type de configuration autorise la croissance modulaire: chaque système composant le cluster participe à l'exécution de la charge globale (redondance participative).
• Si les applications sont transactionnelles (propriétés ACID), la défaillance est équivalente à un abandon de transaction: l'intégrité des données est assurée. En d'autres termes, il s'agit d'un fonctionnement en mode checkpoint avec redondance participative.
• Ce mode implique un SGBD adapté à ce mode de fonctionnement, par exemple :
  • DB2 Enterprise Edition;
  •Oracle Parallel Server.

Page 31

Cluster NT

Architecture Microsoft Cluster Server (MSCS)



Page 32

Cluster NT(2)

- **Concepts :**
  - Service Cluster : ensemble de logiciels implantés sur chacun des nœuds et gérant les activités spécifiques du cluster
  - Ressource:
    - Entité gérée par le Service Cluster
    - Dite «en ligne» lorsqu'elle est supportée par nœud et fournit son service
    - Accédée au moyen de DLL
  - Groupe : collection de ressources (entité de gestion)
- **Composants :**
  - Gestionnaire du nœud (Node Manager) : gère l 'appartenance du nœud au cluster et surveille l 'état des autres nœuds (technique des battements de cœur)
  - Gestionnaire de la base de données (Database Manager) : maintient la base de données (répliquée) représentant l'état du cluster
  - Gestionnaire de ressource/gestionnaire de recouvrement : gère les ressources et les groupes et lance les actions appropriées en cas de changement d'état
  - Traitement des événements : assure la connexion entre les composants du Service Cluster et prend en charge les opérations communes
  - Gestionnaire de communications
  - Gestionnaire des mises à jour : prend en charge les fonctions de mise à jour des composants du Service Cluster
  - Moniteurs de ressource : s 'assurent de l 'état de bon fonctionnement des ressources (au moyen de «call backs»); implémentés sous forme de processus, ils communiquent avec les services cluster au moyen de RPC
  - Service de temps : fournit un temps homogène au sein du cluster

Cluster NT(3)

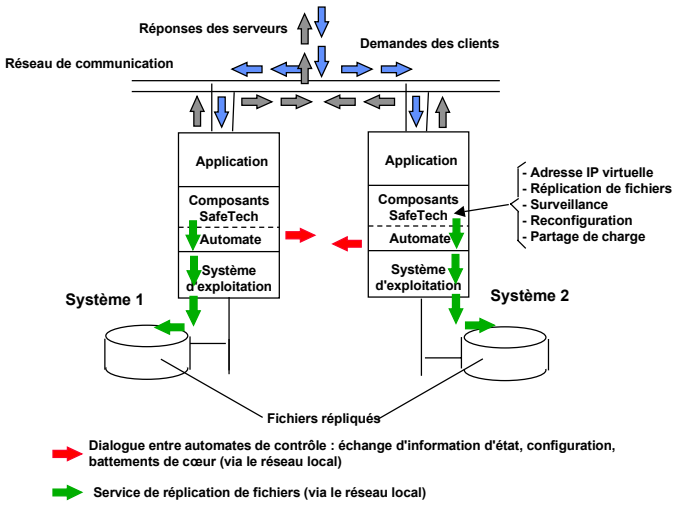
- **Principe de fonctionnement**
  - En cas de défaillance d'une ressource, le gestionnaire de la ressource décide de :
    - de la redémarrer
    - de la mettre «hors ligne» et de la redémarrer sur un autre nœud (action prise en charge par le gestionnaire de recouvrement)
  - En cas de défaillance d 'un nœud, la détermination du ou des nœuds prenant en charge les groupes de ressources qui étaient supportés par le nœud défaillant est négocié entre les «survivants»
  - Temporisation du redémarrage d 'une ressource pour éviter les défaillances «oscillantes»
  - Processus d 'initialisation du cluster automatique avec découverte dynamique de la configuration
  - Un nœud peut quitter un cluster (*cluster exit*)
  - Le temps de recouvrement est de l'ordre de la minute
- **Visibilité des défaillances**
  - Défaillance invisible pour les connexions sans état (ou stateless, ex. navigateurs) si elle intervient entre deux requêtes (sinon c'est à l'application cliente -qui est notifiée- d 'assurer la relance)
  - Pour des connexions avec état (statefull), c 'est aux applications de gérer la reconnexion
- **Intégrité des données : pas assurée en dehors du transactionnel (idem cluster HACMP)**

SafeTech

- Technologie générique permettant de bâtir des solutions à haute disponibilité sur des systèmes standard (NT ou Unix). Permet de résister aux défaillances du matériel et du logiciel
- Nécessite au moins deux systèmes interconnectés par un réseau local
- Ensemble de composants middleware installé sur la plate-forme et fonctionnant sous le contrôle d 'un automate :
  - Virtualisation de l 'adresse IP
  - Système de fichiers redondant distribué
  - Mécanisme de surveillance des systèmes
  - Mécanisme de reconfiguration
  - Mécanisme de partage de charge
- Peut nécessiter une adaptation des applications pour pouvoir bénéficier de la continuité de service
- Deux modes de fonctionnement :
  - primaire-secours (un seul système actif, l 'autre «suit»)
  - partage de charge
- Résultat d 'un coopération IRISA/Bull (voir http://www.evidian.com)
- Utilisé dans différents produits (ex pare-feu, serveur Web,…)

SafeTech(2)

- Architecture générale

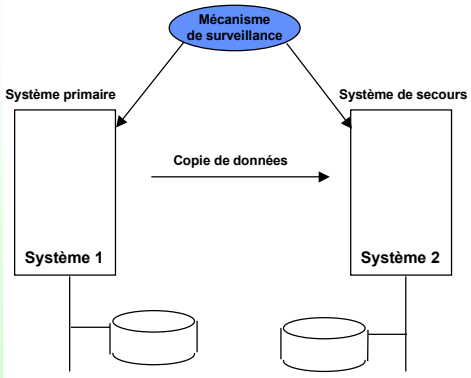


Recouvrement après catastrophe

- Terme anglo-saxon : Disaster Recovery
- Objectif : assurer la permanence du service (après interruption éventuelle) en cas de catastrophe sur le site du système d 'information (ex inondation, incendie, tremblement de terre, émeute, ….)
- Implique des sites géographiquement distants
- Solutions coûteuses (ex 2 sites de même capacité de traitement avec mise à jour synchronisée des données)
- Quelques critères de choix d'une solution :
  - Écart pouvant exister entre l 'état des données sur le site principal et le site de secours
  - Temps nécessaire à la reprise du traitement
  - Coût de la solution
- Quelques exemples pour diminuer le coût des solutions :
  - Système de secours dimensionné comme le système primaire mais utilisé pour d'autres tâches
  - Les deux sites jouent des rôles symétriques
  - Recours à un prestataire de service

Recouvrement après catastrophe(2)

- Architecture générale d'une solution de recouvrement



Classification des stratégies et des pratiques de recouvrement (Gartner Group) :
- Niveau 0 : pas de recouvrement
- Niveau 1 : sauvegardes mais sans test des sauvegardes
- Niveau 2 : sauvegardes de la totalité du stockage, à des moments arbitrairement choisis, avec test de leur contenu
- Niveau 3 : sauvegardes systématiques de la totalité du stockage avec test de leur contenu
- Niveau 4 : sauvegarde systématiques des données relatives aux applications critiques avec test du contenu des sauvegardes
- Niveau 5 : mise à jour systématique et en synchronisme avec le déroulement des applications

Estimation de la disponibilité des systèmes

AMDEC/AEEL

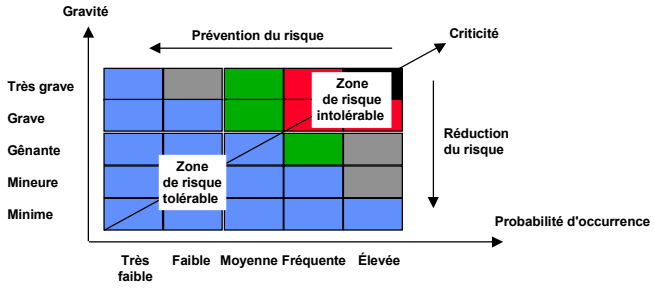
- AMDEC (Analyse des Modes de Défaillance, de leurs Effets et de leur Criticité)
  - Initialement utilisée pour le matériel, a été étendue au logiciel (AEEL Analyse des Effets des Erreurs du Logiciel)
  - Consiste à déterminer, pour chaque fonction ou constituant d 'un système, ses modes de défaillance possibles ainsi que les conséquences
  - Quantification de la probabilité d'occurrence et de la gravité des effets
  - Évaluation de la criticité (synthèse sous la forme d'une matrice)
- Utilisable lors de la conception des systèmes
  - permet de prévoir des couches défensives dans la programmation
  - pilotage des test de validation
  - choix d 'une stratégie de maintenance

AMDEC/AEEL(2)

Structure d'un tableau AMDEC

Fonction (composant)	Mode de défaillance	Cause	Effet Conséquences	Moyen de détection	Probabilité (P)	Gravité (G)	Criticité (P x G)	Actions

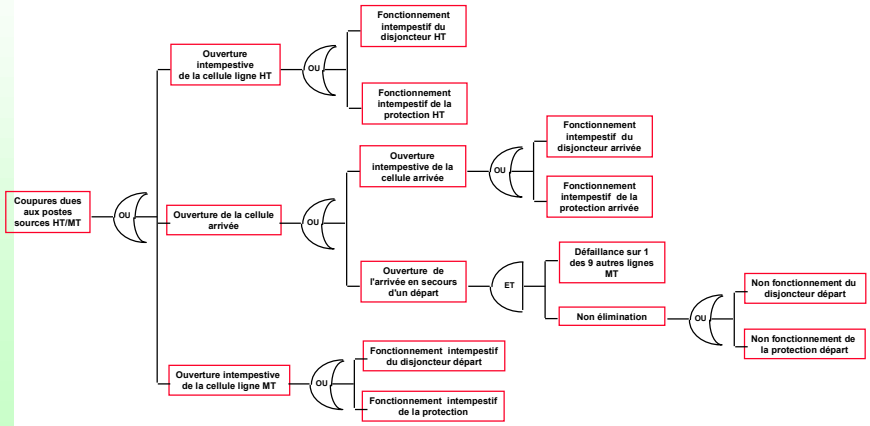
Matrice de criticité (Source [MEI98])



La matrice peut être complétée par une troisième dimension qui représente la difficulté de détection.

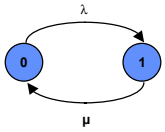
Arbres de défaillance

Complète l'AMDEC qui ne traite pas les combinaisons des défaillances. Permet de rechercher les causes d 'occurrence d 'un événement redouté par la combinaison de clauses logiques «et» et «ou». Exemple :

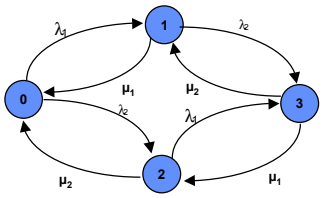


Estimation quantitative de la disponibilité

- Les chaînes de Markov [SIE92] ou les réseaux de Petri stochastiques permettent l 'estimation quantitative de la disponibilité des systèmes. Dans le cas des chaînes de Markov, le graphe état/transition est valué par des taux de défaillance λ et des taux de réparation μ. La résolution du modèle donne la probabilité que le système a de se retrouver dans différents états stationnaires.



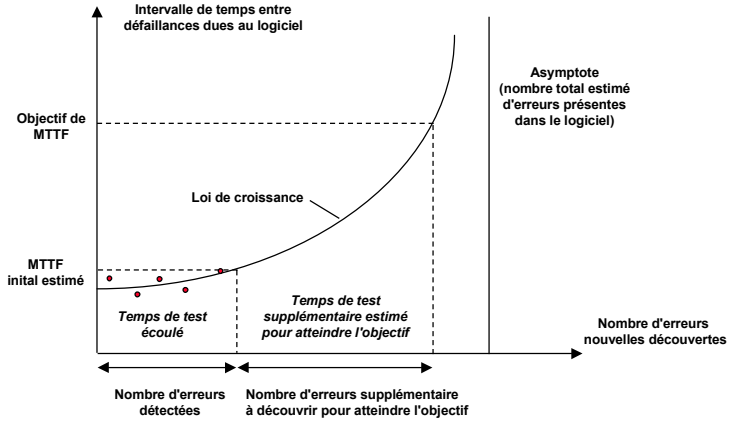
Modèle de Markov
Système à 2 états



Modèle de Markov
Système doublé

Modèles de fiabilité du logiciel

- Différents modèles de croissance de la fiabilité du logiciel ont été proposés. Pratiquement, ces modèles servent à calibrer (à partir d 'une loi de croissance) l'effort de test restant encore à effectuer pour atteindre un objectif de fiabilité.



Comparaison des solutions

- **Comparaison (partielle) des propriétés des solutions**

	« Pair and Spare » (matériel)	Points de reprise (logiciel)	Cluster (logiciel)	Environnement d'exécution SafeTech (logiciel)
Tolérance aux défaillances du matériel	Oui	Oui	Oui	Oui
Tolérance aux défaillances du logiciel de base	Non	Oui (*Heisenbugs*)	Oui (*Heisenbugs*) et si absence de partage des données	Oui (*Heisenbugs*) et si absence de partage des données
Tolérance aux défaillances du logiciel d'application	Non	Oui (*Heisenbugs*)	Oui (*Heisenbugs*) et si absence de partage des données	Oui (*Heisenbugs*) et si absence de partage des données
Programmation spécifique des applications	Non	Oui (logique transactionnelle)	Oui en cas de partage et de mise à jour des données	Oui en cas de partage et de mise à jour des données
Dégradation des performances suite à défaillance	Non	Oui	Oui	Oui
Tolérance à la réparation	Oui (nécessairement)	Oui (implicitement)	Oui (implicitement)	Oui (implicitement)

- **Coût/Performance**

	Performance	Temps de recouvrement O(x) signifie de l'ordre de x	Coût
Système de base	1	Pas applicable	1
« Pair and Spare »	<1	O(seconde)	>>3
Système « Stand By »	~1	O(minute)	~2
Redondance participative (deux systèmes se partageant la charge de travail)	<2	O(minute)	~>2

Page 45

Perspectives

- **Augmentation du besoin de continuité de service**
- **Augmentation de la fiabilité du matériel**
  - **Intégration des composants**
  - **Moyens de validation des composants**
  - **Intégration de la redondance et des moyens de masquage des défaillances**
  - **Qualité de fabrication et de contrôle**
  - **Auto-surveillance des composants**
- **Difficultés croissantes avec le logiciel**
  - **Volume sans cesse croissant du logiciel**
  - **Coût de la validation**
  - **Voies d 'exploration :**
    - **Conception de logiciel résistant aux défaillances**
    - **Conception du logiciel en vue du test**
    - **COTS**
- **Concentration de l 'industrie et standardisation des solutions (pour les systèmes d 'information car il reste des exigences non couvertes par les technologies standard)**
  - **Diminution de la part des solutions spécifiques de type matériel en raison des progrès des solutions standard (clusters)**

Page 46

Ouvrages et sites de référence

[CHE00] René J. Chevance «Serveurs multiprocesseurs, clusters et architectures parallèles»
Eyrolles 2000

[GRA90] Jim Gray "A Census of Tandem System Availability Between 1985 and 1990 "
IEEE Transaction on Reliability Vol 39., No 4., October 1990 pp409-418

[GRA91] Jim Gray, Daniel Siewiorek, "High Availability Computer Systems"
IEEE Computer, Vol. 24, N° 9, September 1991, pp. 39-48.

[MEI98] Jean-Pierre Meinadier «Ingénierie et intégration des systèmes»
Hermès, 1998

[SIE92] Daniel P. Siewiorek, Robert S. Swarz, "Reliable Computer Systems Design and evaluation", 2nd Edition,
Digital Press, 1992.

Les sites des différents constructeurs de systèmes (parmi ceux qui ont été cités : Compaq/Tandem, Bull/Evidian, IBM, Microsoft, Stratus et Sun)