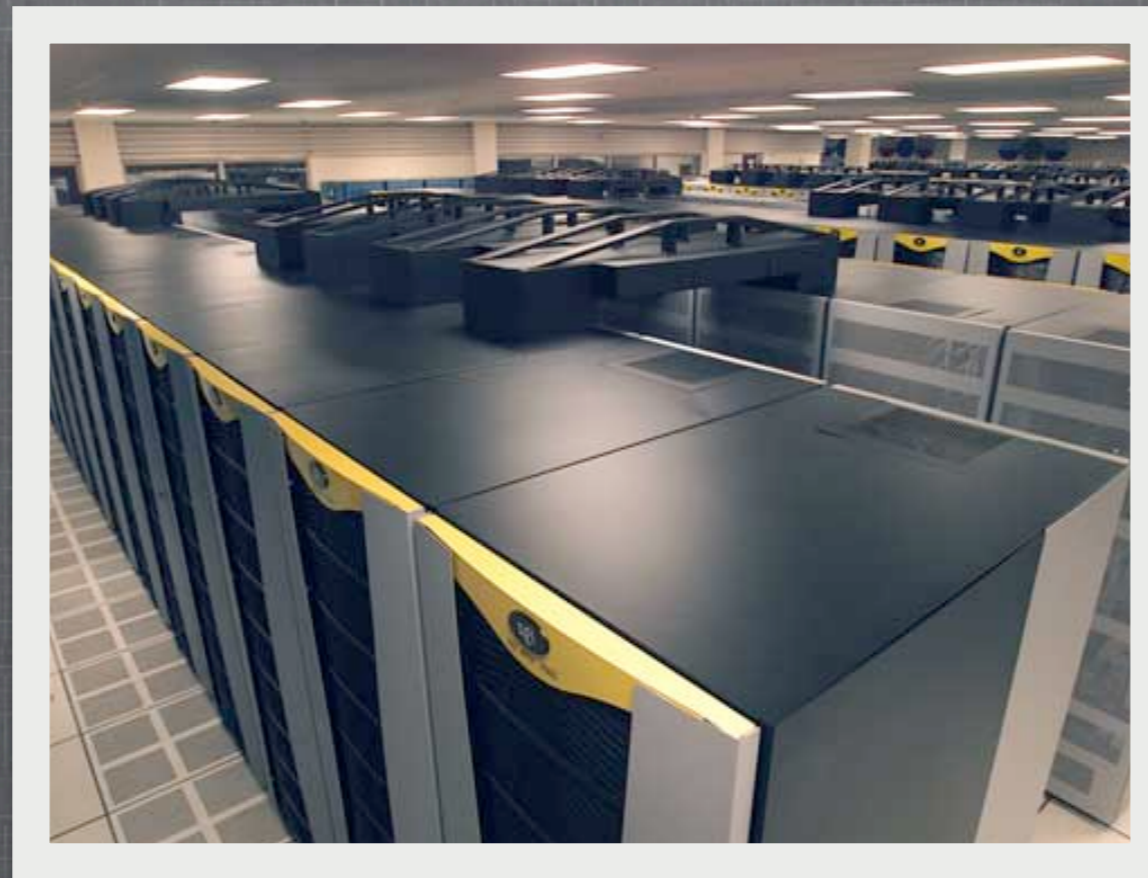


FORMATION

Linux-HA et les systèmes de Cluster



PLAN DE LA PRÉSENTATION

1. Aperçu des différents systèmes de cluster
2. Notions de haute disponibilité
3. Notions spécifiques aux clusters
4. Fonctionnement technique de HA
5. Fonctionnement spécifique du failover
6. Aspects de la configuration
7. Atelier pratique

PLAN DE LA PRÉSENTATION

1. Aperçu des différents systèmes de cluster
2. Notions de haute disponibilité
3. Notions spécifiques aux clusters
4. Fonctionnement technique de HA
5. Fonctionnement spécifique du failover
6. Aspects de la configuration
7. Atelier pratique

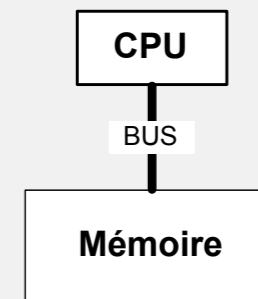
APERÇU DES DIFFÉRENTS TYPES DE CLUSTERS

- *Bref historique sur les clusters*
- *Les clusters hautes performances*
- *Les clusters haute disponibilité*
- *Les clusters de répartition de charge*

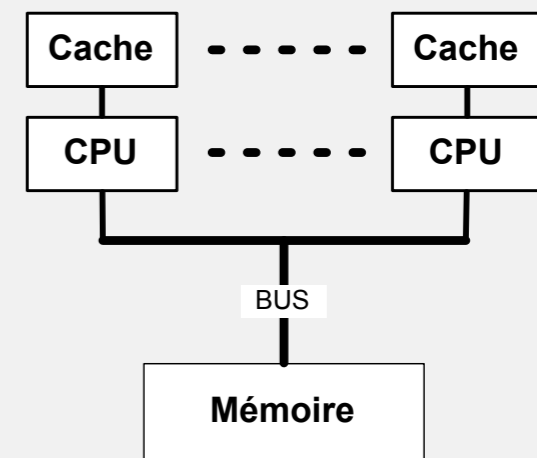
ARCHITECTURE D'UN ORDINATEUR

- **Ordinateur classique :**
 - 1 processeur
 - 1 bus
 - de la mémoire
- **Ordinateur à architecture SMP (Xéon) :**
 - N processeurs
 - 1 bus partagé
 - de la mémoire
- **Ordinateurs à architecture NUMA (SGI) :**
 - N processeurs
 - chaque CPU à sa propre mémoire
 - chaque process à accès à l'ensemble de la mémoire

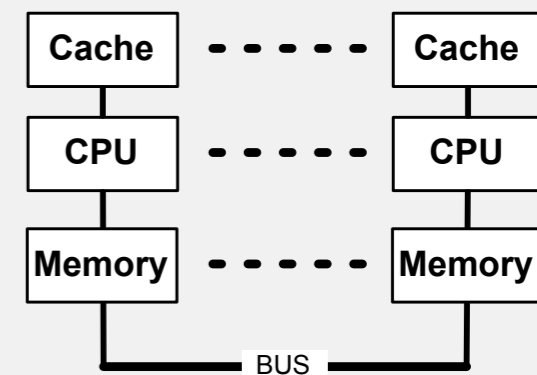
Schéma fonctionnel d'un ordinateur



Uniform Memory Access ou SMP

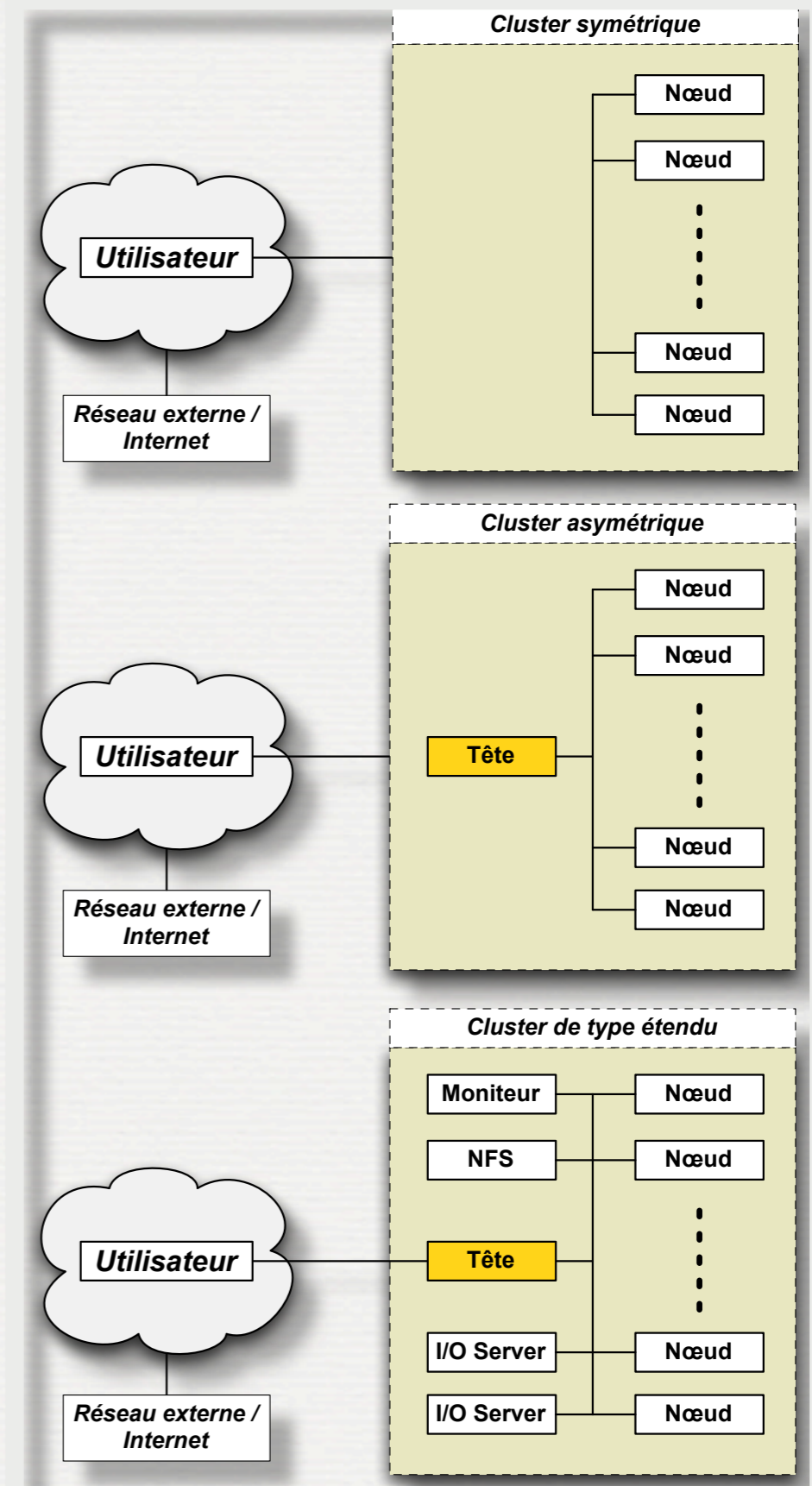


Accès mémoire non uniforme NUMA



DE L'ORDINATEUR AU CLUSTER

- Un cluster est un ensemble de ressources qui forment un système qui améliore les capacités d'un nœud isolé
- Disponibilité :
 - Fiabilité combinée de tous les nœuds.
 - Les services demeurent disponibles même si un nœud isolé tombe
 - Les nœuds restants prennent le relais
- Performance :
 - Puissance combinée de tous les nœuds qui fonctionnent en parallèle
 - Distribution de la charge sur les autres membres du cluster
 - Systèmes qui peuvent se dimensionner



LES TYPES DE CLUSTERS : HAUTE PERFORMANCE

- **Problématique centrale : Améliorer les performances des ordinateurs :**
 - *utiliser un meilleur algorithme*
 - *Utiliser un ordinateur plus puissant*
 - *diviser la calcul entre plusieurs ordinateurs*
- **Serveurs de répartition de charge**
 - *Fourni une répartition de la charge ainsi qu'une protection contre la défaillance en éradiquant les nœuds défaillants.*
 - *Linux Virtual Server*



Computer Science Center
University of Virginia

Burroughs B5500 Computer
Installed July 1964

POINTS DE DÉFAILLANCE

- **Infrastructure :**
 - *Électricité*
 - *Air conditionné*
 - *Câblage en général*
 - *Un centre d'hébergement unique*
- **Réseau :**
 - *Connexion Internet*
 - *Liens Intranet*
 - *Firewalls*
- **Serveurs :**
 - *Ventilateurs*
 - *Disques*
 - *Carte réseau*
- **Applications :**
 - *Applicatifs*
 - *Gestionnaire de cluster*
- **Compétences :**
 - *Un unique administrateur*
 - *Pas de documentation*

LES TYPES DE CLUSTER : HAUTE DISPONIBILITÉ

- Mettre en commun un groupe d'ordinateur pour fournir un service même en cas de défaillance d'un des composants système
 - Quand une machine tombe, les autres prennent la relève
 - *Ceci implique la reprise des adresses IP, des Services, ...*
 - Les nouvelles tâches parviennent à la machine qui a pris la relève
 - Pas d'impact sur les performances
- ➔ Assurer la reprise en cas de défaillance dans un datacenter où en local
- ➔ Politique de protection de site : positionnement dans différents bâtiments où différents continents

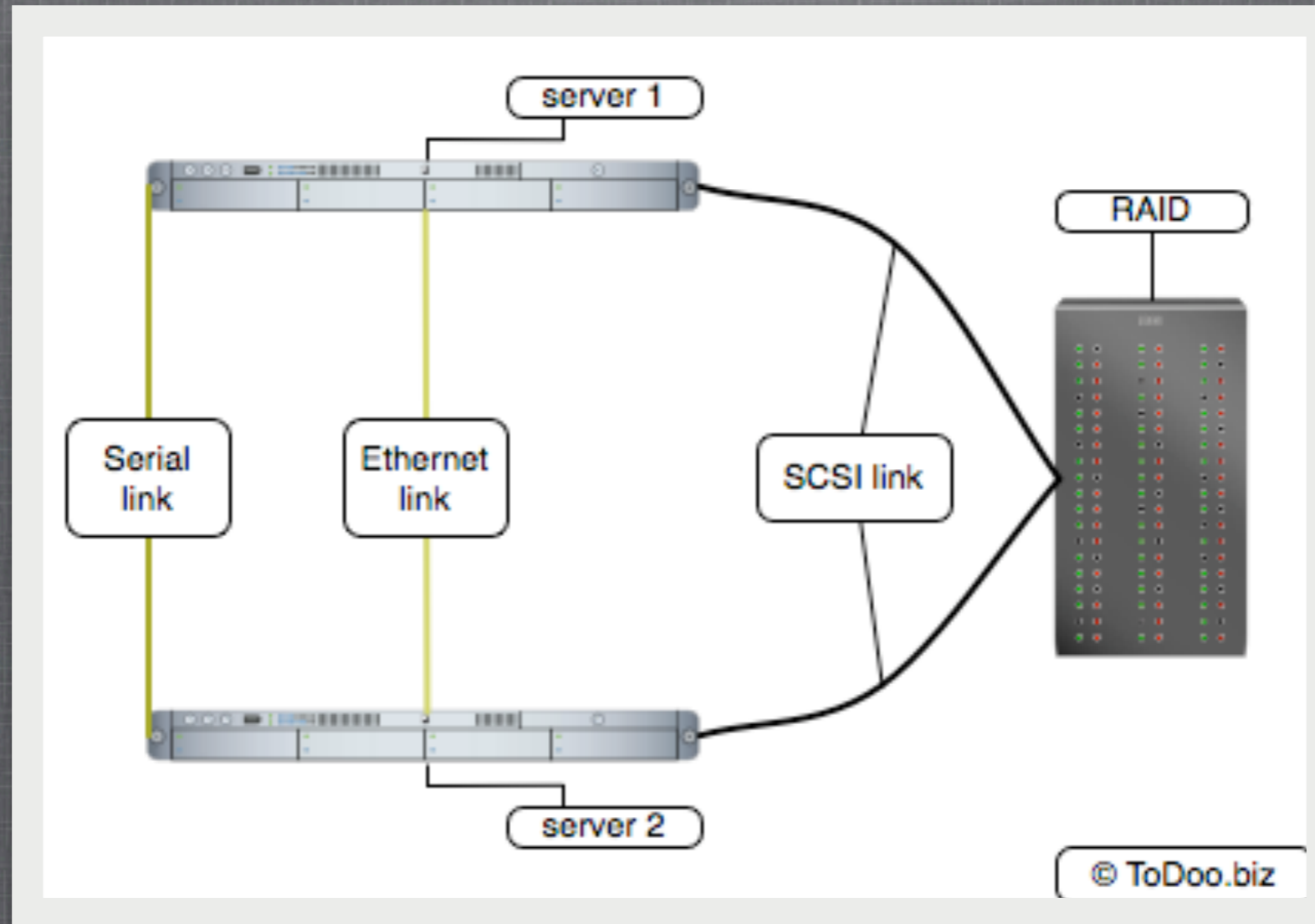
NOTIONS DE HAUTE DISPONIBILITÉ : QU'EST-CE QUE LA HA PEUT FAIRE POUR VOUS ?

- Cela ne permettra pas d'atteindre 100% de disponibilité : c'est impossible !!
- Les clusters HA sont destinées à assurer la continuité de service en cas d'UNE défaillance
- Permet de réduire les temps d'interruption
 - *D'une seconde à quelques minutes*
- Comme le magicien :
 - *Quand cela se passe bien, rien n'est perceptible*
 - *Quand cela se passe pas si bien, cela peut-être légèrement perceptible*
- Un bon système de cluster HA permet d'ajouter un "9" à votre disponibilité de base : 99 -> 99,9 99,9 -> 99,99 ...
- La complexité est l'ennemi de la fiabilité !

PLAN DE LA PRÉSENTATION

1. Aperçu des différents systèmes de cluster
2. Notions de haute disponibilité
3. Notions spécifiques aux clusters
4. Fonctionnement technique de HA
5. Fonctionnement spécifique du failover
6. Aspects de la configuration
7. Atelier pratique

NOTIONS DE HAUTE DISPONIBILITÉ : SCHÉMA DE REPRISE DE CHARGE APRÈS UN INCIDENT



NOTIONS DE HAUTE DISPONIBILITÉ : QUELQUES MENSONGES ET DES STATISTIQUES

La règle des 9 :

99,9999%	30 secondes
99,999%	5 minutes
99,99%	52 minutes
99,9%	9 heures
99%	3,5 jours

NOTIONS DE HAUTE DISPONIBILITÉ : A QUOI LINUX-HA RESSEMBLE-T-IL DE CONNU ?

- **Cela ressemble aux scripts init avec en plus :**
 - *(de façon optionnel) l'ajout de paramètres*
 - *le fait de fonctionner sur plusieurs ordinateurs*
 - *l'ajout de politiques pour :*
 - *L'ordre dans lequel les choses vont se dérouler*
 - *Comment les services interagissent entre eux*
 - *Quand lancer les services*
- **Les cluster HA ressemble aux scripts init "dopés"**

NOTIONS DE HAUTE DISPONIBILITÉ : LES SPÉCIFICITÉS DU SYSTÈME ?

- **Les cluster HA introduisent des concepts nouveaux :**
 - *Cerveau partagé : “Split-Brain”*
 - *Quorum*
 - *Fencing*
- **Le partage des données n’est pas un problème avec un seul serveur - c’est un problème CRITIQUE pour les clusters**

PLAN DE LA PRÉSENTATION

1. Aperçu des différents systèmes de cluster
2. Notions de haute disponibilité
3. Notions spécifiques aux clusters
4. Fonctionnement technique de HA
5. Fonctionnement spécifique du failover
6. Aspects de la configuration
7. Atelier pratique

NOTIONS SPÉCIFIQUES : SPLIT-BRAIN

- Les erreurs de communication peuvent aboutir à partitions séparés sur le cluster.
- Si chacune de ces partitions essayent de prendre le contrôle du cluster, nous arrivons à une situation de Split-Brain
- Si cela se produit, des choses désagréables peuvent survenir :
 - *Perte de données difficilement réversibles*

NOTIONS SPÉCIFIQUES : QUORUM

- La notion de Quorum est une tentative d'éviter la situation de split-brain pour la plupart des cas
- Typiquement : on s'assure qu'une seule partition du cluster est active à un instant t
- Le Quorum est le terme qui définit les méthodes qui permettent de s'assurer de cela
- La plupart des Quorum sont des systèmes de votes qui permettent de déterminer quel nœud peut avoir accès aux ressources (par un calcul).
- Cela évite un fonctionnement avec 2 nœuds

NOTIONS SPÉCIFIQUES : FENCING

- Fencing essayi de mettre une barrière autour d'un ou plusieurs nœud "perdu" afin d'éviter leur accès au cluster
- De cette façon on ne dépend pas de la bonne conduite ou du timing du nœud perdu
 - *On utilise STONITH pour cela : Shoot The Other Node In The Head*
- **D'autres techniques existent aussi :**
 - *Lockout au niveau des switch FiberChannel*
 - ...

PLAN DE LA PRÉSENTATION

1. Aperçu des différents systèmes de cluster
2. Notions de haute disponibilité
3. Notions spécifiques aux clusters
4. Fonctionnement technique de HA
5. Fonctionnement spécifique du failover
6. Aspects de la configuration
7. Atelier pratique

FONCTIONNEMENT TECHNIQUE DE HA : MEMBRES D'UN CLUSTER

- Les clusters doivent avoir une vision précise de quels nœuds sont des membres valides à un instant t
- Les nœuds disposent d'un système de contrôle de type "heartbeat" ou sont monitorés grace à un système tiers.
- L'absence de signal "heartbeat" indique la défaillance d'un nœud
- Difficulté de distinguer entre la défaillance d'un lien, d'un nœud, si tous les liens sont coupés, on arrive à un scénario de "cerveau partagé"
- Parceque les défaillances ne peuvent être détectées que de façon asynchrones, l'état d'un cluster n'est JAMAIS certain

FONCTIONNEMENT TECHNIQUE DE HA : SYSTÈME DE SIGNAUX DANS LES CLUSTERS

- Les signaux sont transportés par le réseau...
- *k*-fiabilité : Pendant un intervalle de temps $t : k$ dysfonctionnement sont tolérés et non retransmis aux couches supérieures du système.
- Ordonnancement : l'ordonnancement global est privilégié
 - *Ordonnancement total* : Tous les nœuds voient tous les messages dans le même ordre
 - *Ordonnancement causal* : un message n'est vu que si tous les messages dont il dépend sont reçus
- La cryptographie protège des défaillances malicieuses

FONCTIONNEMENT TECHNIQUE DE HA : LA GESTION DU TEMPS DANS LES CLUSTERS

- Avec des horloges non synchronisées, les événements peuvent avoir des marquages temporelles totalement aléatoires
- Des horloges partiellement synchronisées (NTP) permettent d'établir que deux horloges ne varient pas de plus que Δt
- Pour simuler un temps totalement monotone, les systèmes ne doivent jamais produire un marquage temporelle supérieur à celui d'un événement reçu d'un autre nœud
- Les bases de cette théorie ont été établies par Leslie Lamport

FONCTIONNEMENT TECHNIQUE DE HA : GESTION DES RESSOURCES

- **Ressource : Encapsulation d'une entité physique ou logique qui fourni un service**
 - *Type de Ressources : Adresse IP, partition montée, SMTP, ...*
 - *Instance de ressources : identifié par type, nom et d'autres paramètres*
 - *Un script de Gestion des Ressources prends en charge les ressources d'un certain type, fournissant une API au Gestionnaire de Ressources*
 - *Seul un nœud peu gérer une ressource à un instant t, la gestion par de multiple nœuds implique une corruption des données et doit être évité*
- **Les ressources sont gérés par groupes**
 - *Un ensemble de ressources doivent fournir un service donné*

FONCTIONNEMENT TECHNIQUE DE HA : ACCÈS AUX DONNÉES

- **Stockage partagé par**

- *Fiber Channel*
- *SCSI partagé*
- *iSCSI / HyperSCSI*
- *Réplication logicielle*

- **Fiabilité Hardware**

- *Réplication*
- *Multi-path*
- *RAID*

- **Fiabilité Software**

- *Software RAID*
- *Réplication*
- *Système de fichiers journalisés*
- *Logical Volume Management (LVM)*

- **Accès concurrents**

- *Système de fichiers type SAN (GFS, PolyServe, IBM GPFS)*
- *Systèmes de fichiers distribués (NFS, Lustre, AFS)*

FONCTIONNEMENT TECHNIQUE DE HA : PROTECTION DES I/O ET STONITH

- **Protection au niveau du nœud**
 - *Tuer l'autre machine dans le tête (en Anglais "STONITH")*
 - *Un peu raide, mais assez efficace*
 - *Un nœud qui est éteint ne modifie pas les données*
- **Protection des I/O**
 - *Les ressources se protègent d'elle même en excluant les non-membres*
 - *Déconnecté le port du switch Fiber-Channel*
 - *Contrôleur RAID auto-protégés*
 - *Réservation SCSI (pas toujours fiable)*

FONCTIONNEMENT TECHNIQUE DE HA :

TYPES DE SYSTÈMES FAILOVER

- **A froid**
 - *Le HW est prêt mais ne fonctionne pas*
 - *Les services sont démarrés manuellement sur le nouveau système*
 - *Les coupures sont très remarquables*
- **Tiède**
 - *Le système fonctionne, les données sont synchronisées*
 - *Les services sont démarrés automatiquement*
- *Les petites coupures sont généralement remarquées ("Reboot rapide")*
- **A chaud**
 - *En état de marche*
 - *Tous les systèmes sont actifs*
 - *Le client communique avec de multiples systèmes en même temps*
 - *Pas de coupure remarquable*
- **Pourquoi tout n'est-il pas à chaud ?**
 - *Coût élevé*
 - *Les logiciels doivent être adaptés*

PLAN DE LA PRÉSENTATION

1. Aperçu des différents systèmes de cluster
2. Notions de haute disponibilité
3. Notions spécifiques aux clusters
4. Fonctionnement technique de HA
5. Fonctionnement spécifique du failover
6. Aspects de la configuration
7. Atelier pratique

FONCTIONNEMENT DU FAILOVER : RÉSUMÉ SUR LE SYSTÈME DE FAILOVER

- Les serveurs écoutent le “heartbeat” réciproquement
- Le maître n’envoie pas de “heartbeat” pendant une période de temps
- L’esclave détecte cela et assume une défaillance
- On évite les corruptions de données en mettant en place une “barrière d’accès”
- On procède à la ré-élection du maître
- L’esclave devient maître, les services sont de nouveau alloués sur ce nœud
- Qualifié de “switchover” si cette action n’est pas provoquée par un défaillance (par exemple remplacer le maître par l’esclave pour une opération de maintenance)
- Le “failback” est soit manuel ou automatique

FONCTIONNEMENT DU FAILOVER : APERÇU DE HEARTBEAT

- **Initié par Alan Robertson en 1999**
 - *Branche stable 1.2.x*
 - *Nouvelle branche 2.x*
- **2 nœuds à chaud pour les “ressources de groupe”**
 - *“failback” manuel ou automatique*
- **Simple à mettre en place**
- **Accent sur la sécurité, base de code assez réduite**
- **Liens redondant pour le “heartbeat” et la communication**
 - *Lien série + unicast IPv4, broadcast, multicast*
 - *Des nœuds à “pinger” peuvent être ajoutés comme pseudo membres*

FONCTIONNEMENT DU FAILOVER : APERÇU DE HEARTBEAT II

- Latence minimale grâce à différents systèmes
- Détection de la mort d'un nœud inférieur à la seconde
- Contient de nombreux composants intéressants
 - *Ipfail* peu monitorer la connectivité externe
 - Couche "Consensus Cluster Membership" pour gérer un nombre de nœuds N .
 - Modules STONITH, librairies IPC, librairies PILS
 - Communication simple et la plupart du temps fiable
 - "Cluster Test System" (CTS) inclus
 - Heartbeat au niveau applicatif, vérification des données, ...

FONCTIONNEMENT DU FAILOVER : LES LIMITES D'HEARTBEAT

- Gestion des ressources limitées à deux nœuds
- Les ressources en elle même ne sont pas monitorés en cas de défaillance
- Les configurations ne sont pas automatiquement synchronisées
- Peu de support pour les ressources répliquées

FONCTIONNEMENT DU FAILOVER : LE FUTUR D'HEARTBEAT

- Fonctionnalité multi-nœuds
- Gestion des ressources améliorées
- Agents de gestion des ressources conformes à OCF RA API (gestion des ressources étendues)
- Configuration auto-répliquée (Cluster information Base)
- *k*-fiabilité, ordonnancement du système de messages
- La plupart de ces fonctionnalités mises en œuvre dans la version 2.x

PLAN DE LA PRÉSENTATION

1. Aperçu des différents systèmes de cluster
2. Notions de haute disponibilité
3. Notions spécifiques aux clusters
4. Fonctionnement technique de HA
5. Fonctionnement spécifique du failover
6. Aspects de la configuration
7. Atelier pratique

ASPECTS DE CONFIGURATION DE HA : LES AGENTS DE GESTION DES RESSOURCES

- Autorise Heartbeat à gérer une ressource type
- Heartbeat supporte aussi l'utilisation des scripts `init.d` comme des Agents de gestion des Ressources
- Ecrit sous forme de script shell
- Prend en paramètre une simple liste d'options et une action déterminée
- Points clés à conserver en tête :
 - *Les Agents doivent être capables de gérer un redémarrage après une extinction brutale*
 - *Soyez très précautionneux lors de leur mise au point*
 - *Pour plus de détails se référer à "Open Cluster Framework Resources Agent API"*

ASPECTS DE CONFIGURATION DE HA : AGENTS DE GESTION DES RESSOURCES

- IPaddr, IPaddr2
- IPsrcaddr
- Filesystem
- LVM
- RAID1
- ServeRAID
- ICP
- LinuxSCSI
- portblock
- xinetd
- db2
- WAS
- La plupart des scripts init.d
- Il est assez simple de créer ses propres Agents

ASPECTS DE CONFIGURATION DE HA : MODULES STONITH

- **Network power switches:**
 - *APC MasterSwitch*
 - *WTI NPS/TPS*
 - *BayTech RPC-xxx*
 - *Night/Ware RPC100S*
- **Serial power switches:**
 - *WTI RPS*
 - *APC Smart UPS*
- **IPMI over ethernet**
- **For testing only:**
 - *ssh*
 - **meatware**
 - *(Manual operation)*

ASPECTS DE CONFIGURATION DE HA : IPFAIL

- Heartbeat peut gérer les nœuds externes en tant que pseudo membres du cluster en les “pingant”
- ipfail utilise cette information pour déterminer où placer les ressources, sur le nœud avec la meilleure connectivité externe
- Configuration hyper simple :
 - *Configurer les nœuds à “pinger”*
 - *Informers heartbeat de démarrer ipfail*

ASPECTS DE CONFIGURATION DE HA : ERREURS CLASSIQUES

- Ne pas avoir les fichiers de configuration synchronisés entre tous les nœuds
- Contrôler les ressources dans les fichiers init du systèmes et pas par l'intermédiaire d'heartbeat
- Ne pas utiliser STONITH quand nécessaire
- Les noms de nœuds sont sensibles à la casse
- Câbles séries de pauvre facture
- Déploiement de heartbeat sur des médias non redondant
- Agents de gestion des Ressources qui ne retournent pas l'état correctement

ASPECTS DE CONFIGURATION DE HA : HA.CF

```
debugfile /var/log/ha-debug
logfile /var/log/ha-log
logfacility      local0
keepalive 2
deadtime 30
warntime 5
initdead 60
udpport 694
baud      9600
serial /dev/cuad1
bcast     em1
auto_failback off
node      mail1.xxx.fr
node      mail2.xxx.fr
debug 1
```

ASPECTS DE CONFIGURATION DE HA : HARESOURCES + AUTHKEYS

- **haresources**

```
mail1.xxx.fr \
    147.250.1.1 \
    Filesystem::/dev/da0s1d::/data::ufs \
    Service::named \
    Service::postfix \
    Service::courier-imap-imapd.sh \
    Service::courier-imap-pop3d.sh \
    Service::courier-imap-imapd-ssl.sh \
    Service::courier-imap-pop3d-ssl.sh \
    MailTo::pbinfo@xxx.fr::ClusterMail
```

- **Authkeys doivent avoir root:root, mode 0600 comme autorisation**

```
auth 1
1 sha1 ClusterMailXXXInstallToDoo
```

ASPECTS DE CONFIGURATION DE HA : QUELQUES COMMANDES

`cl_status` script shell qui permet de récupérer un certain nombre d'information sur l'état du cluster :

`hbstatus` : indique si heartbeat tourne sur le système local

`listnodes` : liste les nœuds du cluster

`nodestatus <nom du nœud>` : donne le statu d'un nœud donné

`nodetype <nom du nœud>` : donne les nœuds d'un type donné (ping | normal)

`listhblinks <nom du nœud>` : liste les interfaces réseau utilisées par heartbeat

`hblinkstatus <nom du nœud>` : Indique le statu d'un lien heartbeat

`clientstatus <nom du nœud> <id du client>` : Montre le statu d'un client.

`rsctestatus` : Indique le statu des ressources du cluster (local | foreign | all | none | transition).

`hbparameter -p <nom du paramètre>` : Retourne la valeur d'un paramètre donné.

http://www.linux-ha.org/cl_status

ASPECTS DE CONFIGURATION DE HA : QUELQUES COMMANDES

`hb_takeover` : Récupère les ressources de l'autre nœud (bascule).

Exécuter un “`hb_takeover`” sur le nœud courant équivaut à exécuter “`hb_standby`” sur l'autre nœud.

Shell Script qui permet de provoquer un basculement du nœud actif vers l'autre nœud. C'est un script qu'il peut-être intéressant de faire exécuter par Nagios par exemple.

`hb_standby [all|foreign|local|failback]`

Accessible depuis `/usr/local/lib/heartbeat/hb_standby`

Pour obtenir de l'aide sur la commande “`--help`”

COMPARAISON DES SYSTÈMES DE CLUSTER : HA & LA REMISE EN ŒUVRE

- **HA :**
 - *Reprise peu coûteuse*
 - *Le temps de panne se compte en secondes (normalement)*
 - *Communication inter-nœuds fiable*
- **Remise en œuvre standard**
 - *Reprise coûteuse*
 - *Le temps de panne se mesure en heures*
 - *Communication inter-nœuds peu fiable*

PLAN DE LA PRÉSENTATION

1. Aperçu des différents systèmes de cluster
2. Notions de haute disponibilité
3. Notions spécifiques aux clusters
4. Fonctionnement technique de HA
5. Fonctionnement spécifique du failover
6. Aspects de la configuration
7. Atelier pratique

ATELIER PRATIQUE

Linux-HA et les systèmes de Cluster

