Pascal PORTEAU

Q MANAGER HAUTE DISPONIBILITE

avec

MULTI INSTANCE

SOMMAIRE

Configuration de PRODUCTION tout va bien

•La machine B : son Q MANAGER est Actif.

•En local, les process QM series actifs sont fonctionnels sur le disque local .

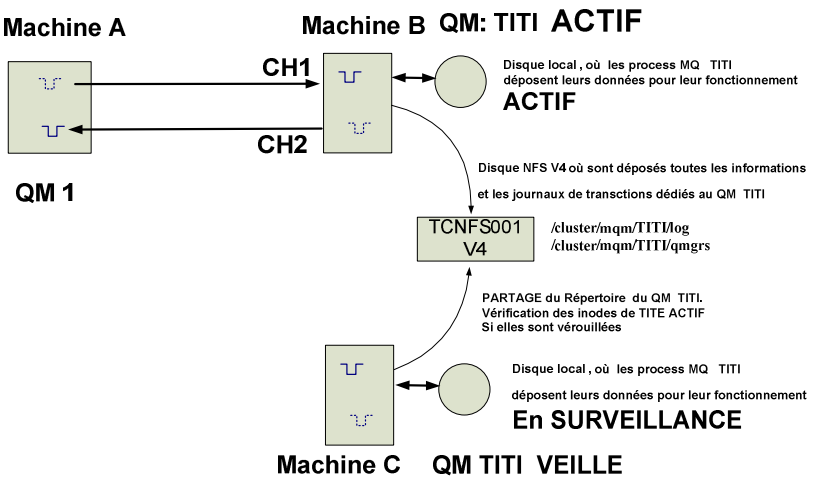
• Les datas et fichiers log sont dirigés vers sur le serveur NFS V4

•La machine C : son Q MANAGER surveille les fichiers partagés.

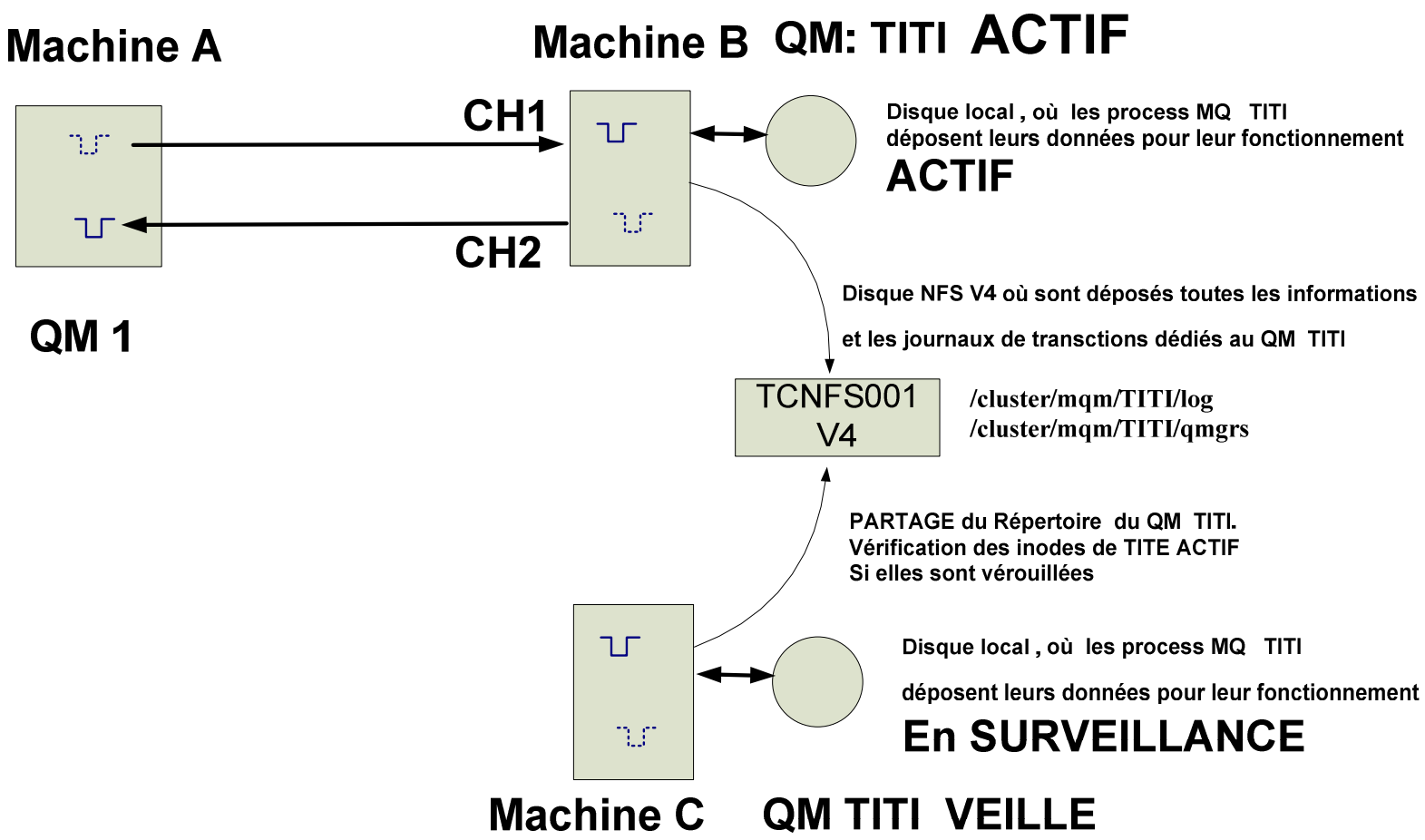
• les process MQ Series sont actifs , ils scrutent les journaux de transactions, les status et si les inodes sont retenues par la machine B

Le schéma suivant donne une meilleure vue

Configuration de PRODUCTION

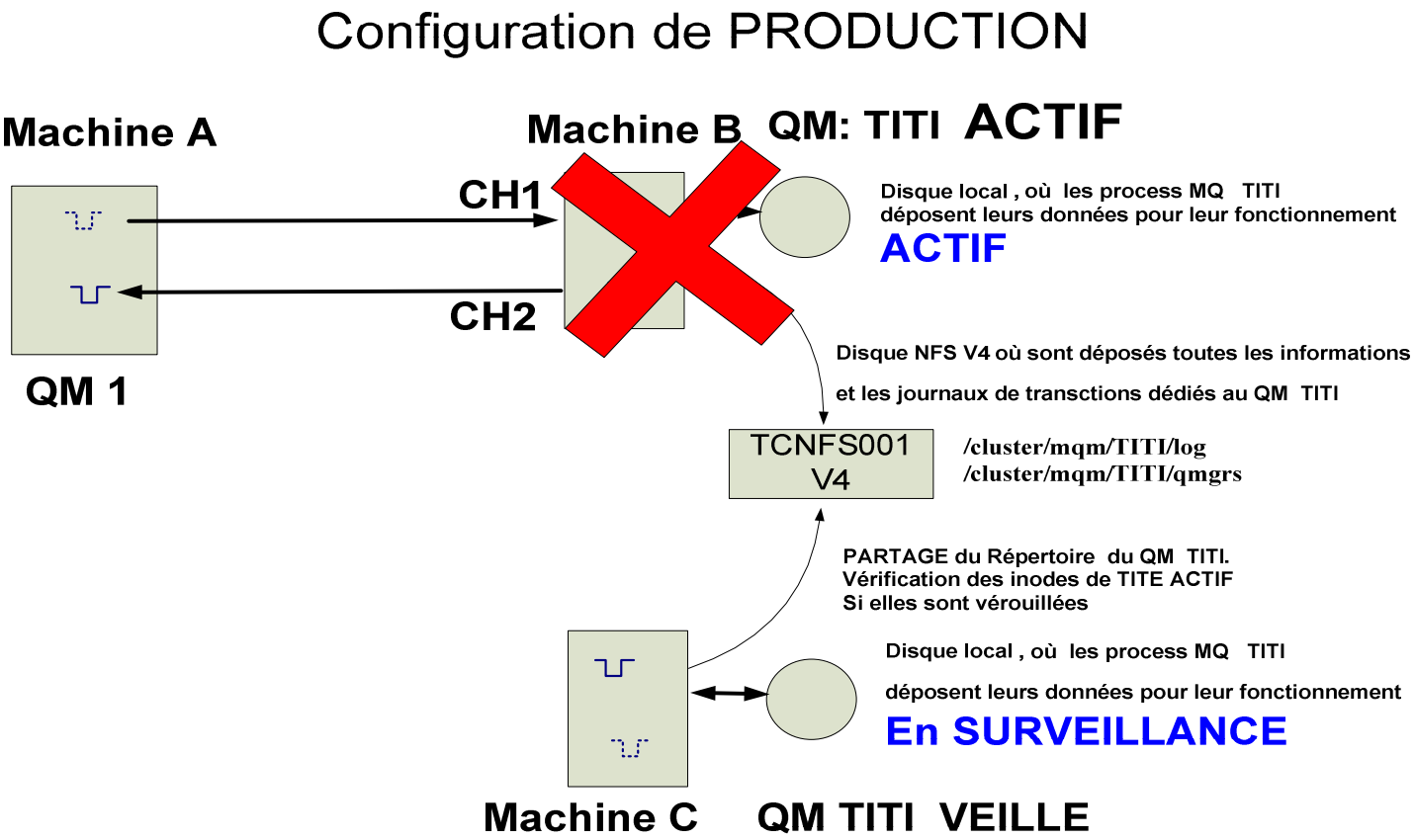


Configuration de PRODUCTION



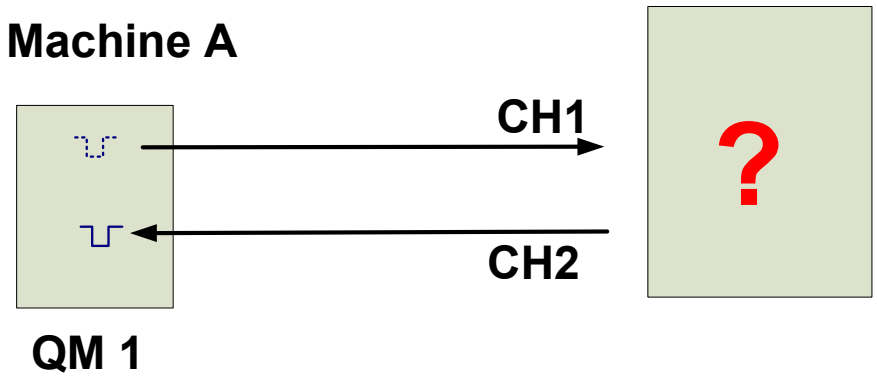
Problème sur LE Q MANAGER: C'est le Drame….  ..... OU pas

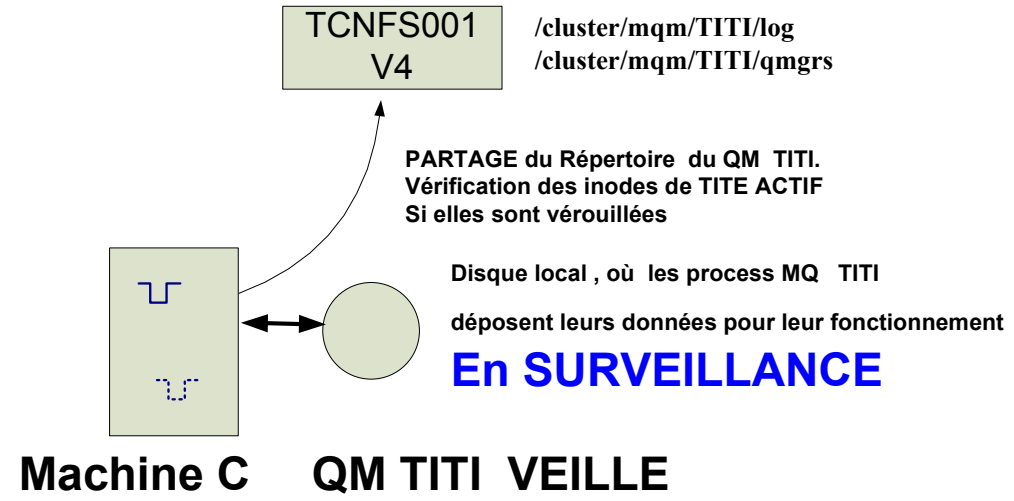
- **Un problème sur la machine B rend celle-ci inutilisable.**
- **Le dialogue entre A et B est arrêté ,donc la production par la même occasion**



Configuration critique pour la production

- Ceci représente la situation de crise par rapport à la situation standard
- Qui va déclencher la bascule vers la machine C ? Oui mais qui?

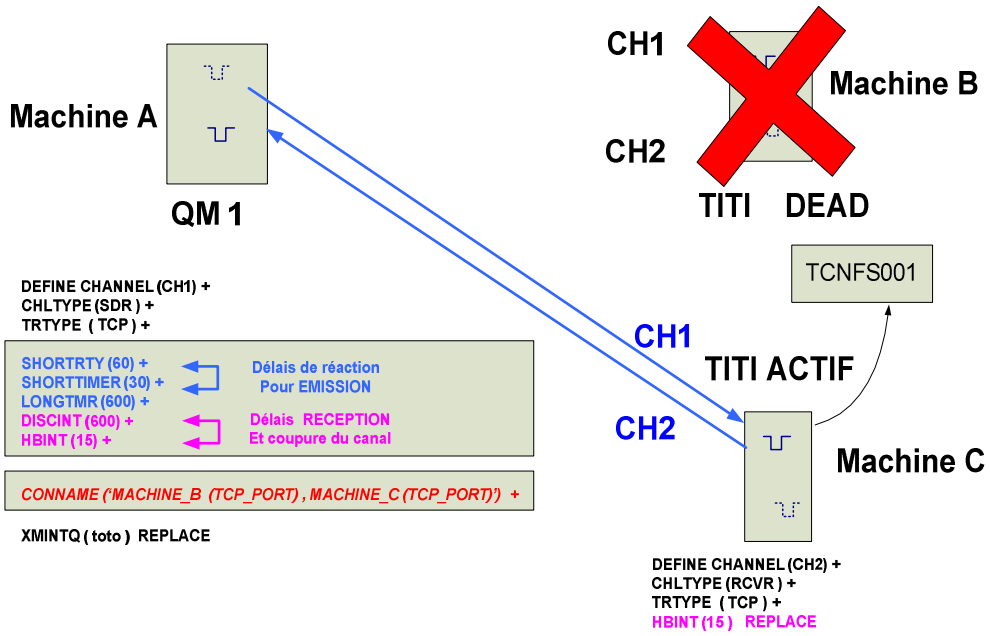




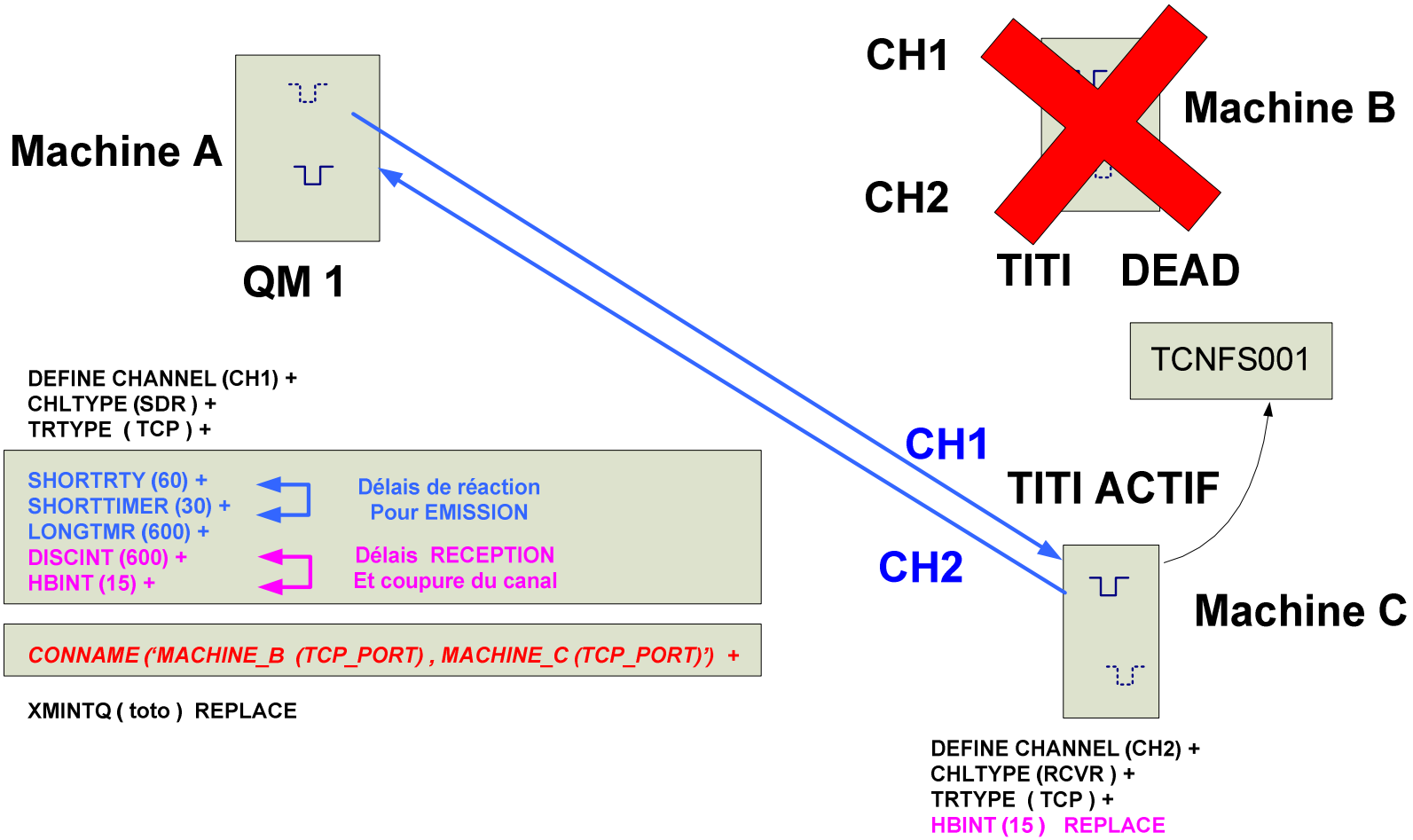
Configuration MULTI- INSTANCES

- **Configuration du Q Manager pour le temps de réactivité à la situation**
- **La machine A garde ses socket TCP durant un timer dans l'opérating système TCP_KEEP_ALIVE. Il faut diminuer cette valeur**
- **Il faut configurer le canal émetteur de la machine A, pour avoir une réaction rapidement.**
- **Voir le prochain schéma pour plus de détail**

Bascule de la MACHINE vers la Machine en VEILLE grâce aux MULTI-INSTANCES



Bascule de la MACHINE vers la Machine en VEILLE grâce aux MULTI-INSTANCES



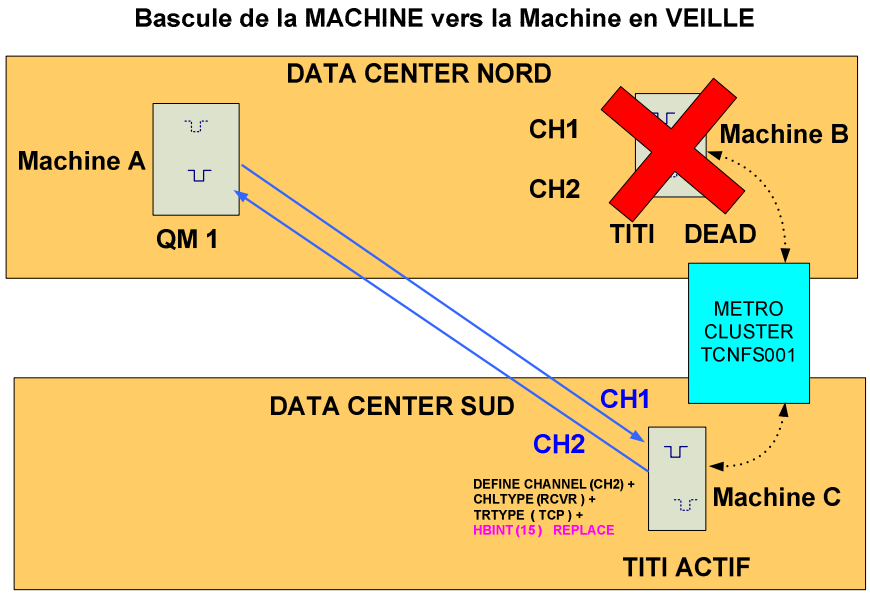
MULTI INSTANCE : Il dirige les flux de données vers la machine C

- La machine primaire est la machine B , la secondaire est la machine C

CONNAME ('IP_MACHINE_B (TCP_PORT) , IP_MACHINE_C (TCP_PORT)') +

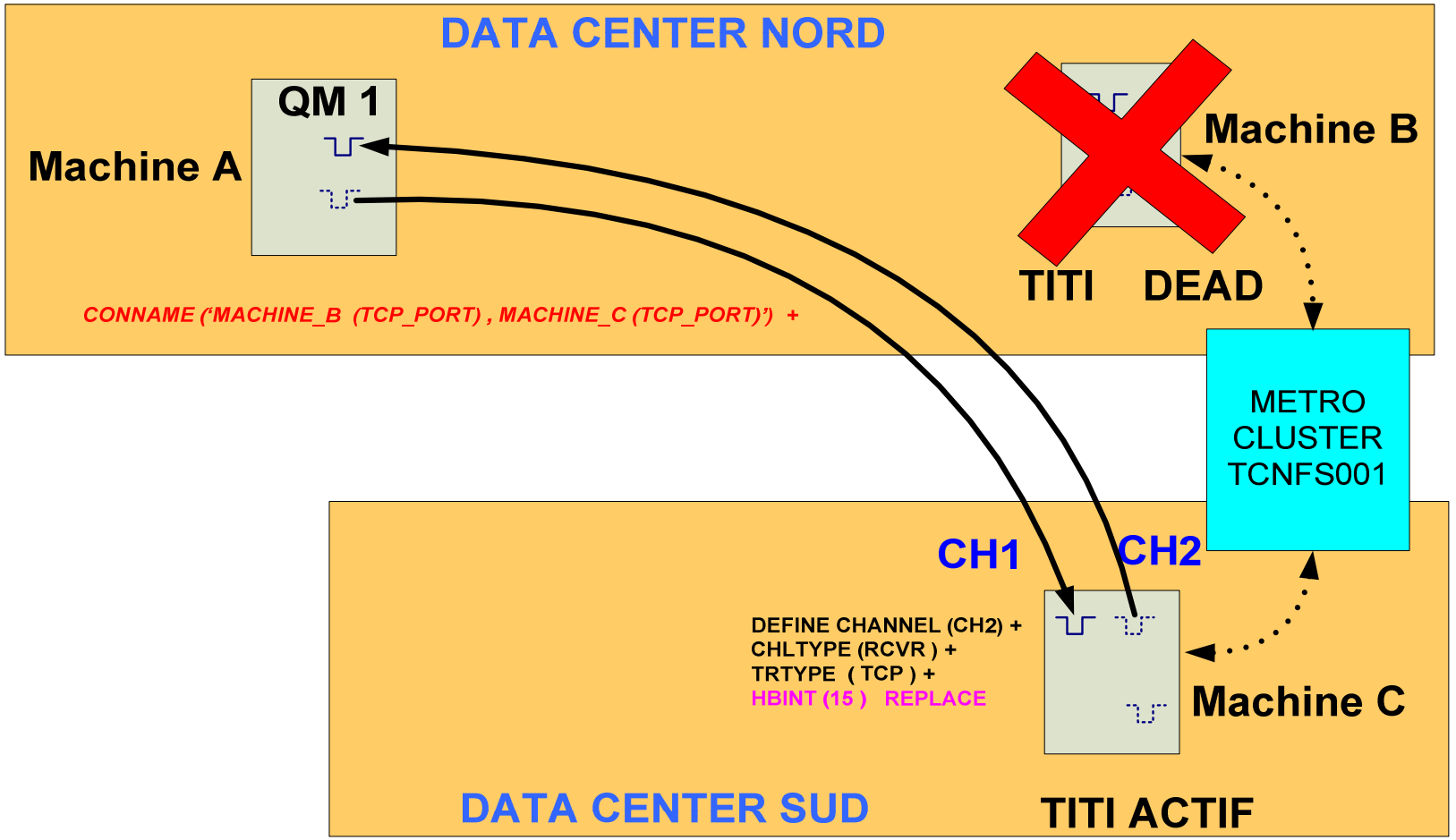
L'intêret est que les adresses IP Machine B et de la machine C peuvent être de classe d'adressage différentes et aussi des ports TCP différents .

Cela permet de diriger le flux vers un autre DATA CENTER , si le serveurs NFS est mapper entre les deux DataCenter (METRO CLUSTER NETAPP…..ou autres méthodes)



MULTI INSTANCE : Il dirige les flux de données vers la machine C

Bascule de la MACHINE vers la Machine en VEILLE



Préparation du serveur NFS V4

- **Seule la version NFS V4 est supporté par IBM.**
- ***Seul MQ SERIES gère ses fichiers et ses caches , pas l'opérating système d'où le noac , le plus direct possible …….***
- **Dans le fichier UNIX /etc/ fstab , on trouve une ligne de ce type:**

SERVEUR NFS:/répertoire_partagé Point_de_montage

TCNFS001:/cluster_WMQ/TITI /cluster/mqm/TITI

Il faut rajouter les valeurs suivantes:

nfs4 ,intr,async,noac,rsize=32768,wsize=32768

nfs4 **:** pour obliger le serveur a chaque extrémité a dialoguer en V

noac: **L'usage de l'option noac offre une plus grande cohérence du cache aux clients NFS qui accèdent aux mêmes fichiers, mais au prix d'une pénalisation significative des performances. C'est pour cette raison qu'une utilisation judicieuse des blocages (locking) de fichiers est de préférence recommandée. La section COHÉRENCE DES DONNÉES ET DES METADONNÉES est de plus importante**

Préparation du serveur NFS V4

- **Délégations pour une mise en mémoire cache côté client**

NFSv4 diffère des versions précédentes du NFS en autorisant un serveur à déléguer à un client des actions spécifiques sur un fichier afin de permettre une mise en cache plus agressive des données côté client ainsi que la mise en cache de l'état de verrouillage. Un serveur cède le contrôle des mises à jour de fichiers et de l'état de verrouillage à un client via une délégation. Le temps d'attente est réduit en autorisant le client à effectuer diverses opérations et de mettre des données en cache localement. Deux types de délégations existent actuellement : la lecture et l'écriture. Le serveur a la possibilité de rappeler la délégation d'un client en cas de contention d'un fichier.

Une fois qu'un client détient une délégation, il peut effectuer des opérations sur les fichiers dont les données ont été mises en cache localement afin d'éviter des lenteurs du réseau et d'optimiser les E/S. La mise en cache plus agressive résultant de délégations peut constituer un atout considérable pour les environnements dont les caractéristiques sont les suivantes :

- **Pour le exemple sur un fas 3510 DE netapp :**

>>>>>nfs.v4.read_delegation off (value might be overwritten in takeover)

>>>>>nfs.v4.write_delegation off (value might be overwritten in takeover)

Préparation du serveur NFS V4

- **Configuration des variables réseau TCP**
- **[mqm@]$ cd /proc/sys/net/ipv4**
- **[mqm@]$ cat tcp_keepalive_time**

7200

- **[root@]# echo 600 > tcp_keepalive_time**
- **[root@]# echo 60 > tcp_keepalive_intvl**
- **[root@]# echo 20 > tcp_keepalive_probes**
- **sysctl -w**

net.ipv4.tcp_keepalive_time=600

**net.ipv4.tcp_keepalive_intvl=60 **

net.ipv4.tcp_keepalive_probes=20

Création de la configuration du Q MANAGER

1. **Création du Q MANAGER sur le serveur qui sera actif sur machine B :**

```
[mqm@]$ mkdir   /cluster/mqm/mqm/titi/qmgrs
[mqm@]$ mkdir  /cluster/mqm/titi/log

 crtmqm -ll -ld /cluster/mqm/titi/log -md /cluster/mqm/mqm/titi/qmgrs  TITI

WebSphere MQ queue manager created.
Directory '/cluster/mqm/titi/qmgrs/titi' created.
Creating or replacing default objects for TITI.
Default objects statistics : 65 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
[mqm]$
```

Création de la configuration du Q MANAGER primaire

- **Q MANAGER est créé , on vérifie par la demande d'informations:**

[mqm@]$ dspmqinf TITI

QueueManager:
Name=TITI
Directory=TITI
Prefix=/var/mqm <<<<< C'est ici que sont pris les process MQseries
DataPath=/cluster/mqm/titi/qmgrs/titi <<<<< C'est ici que les messages iront
Les logs iront dans /cluster/mqm/titi/qmgrs/titi

Démarrage du Q MANAGER :

[mqm@]$ strmqm -X TITI
WebSphere MQ queue manager 'TITI' starting.
11 log records accessed on queue manager 'TITI' during the log replay phase.
Log replay for queue manager 'TITI' complete.
Transaction manager state recovered for queue manager 'TITI1'.
WebSphere MQ queue manager 'TITI' started.
[mqm@]$
Q MANAGER est OK:
[mqm@]$ dspmq
QMNAME(TITI) STATUS(Running)
[mqm@]$

Création de la configuration du Q MANAGER secondaire

- **Demande des informations du Q MANAGER sur la machine secondaire :**

  [mqm@p]$ dspmqinf -o command TITI

  AMQ7271: WebSphere MQ configuration information does not exist.

- **Création du Q MANAGER sur la machine C qui sera en veille :**

  [mqm@]$ addmqinf -s QueueManager -v Name=TITI -v Directory=TITI -v Prefix=/var/mqm –v

  DataPath=/cluster/mqm/titi/qmgrs/titi

  WebSphere MQ configuration information added.

- **Demande des informations du Q MANAGER :**

  [mqm@]$ dspmq

  QMNAME(TITI) STATUS (Running elsewhere)

- **Running elsewhere :**

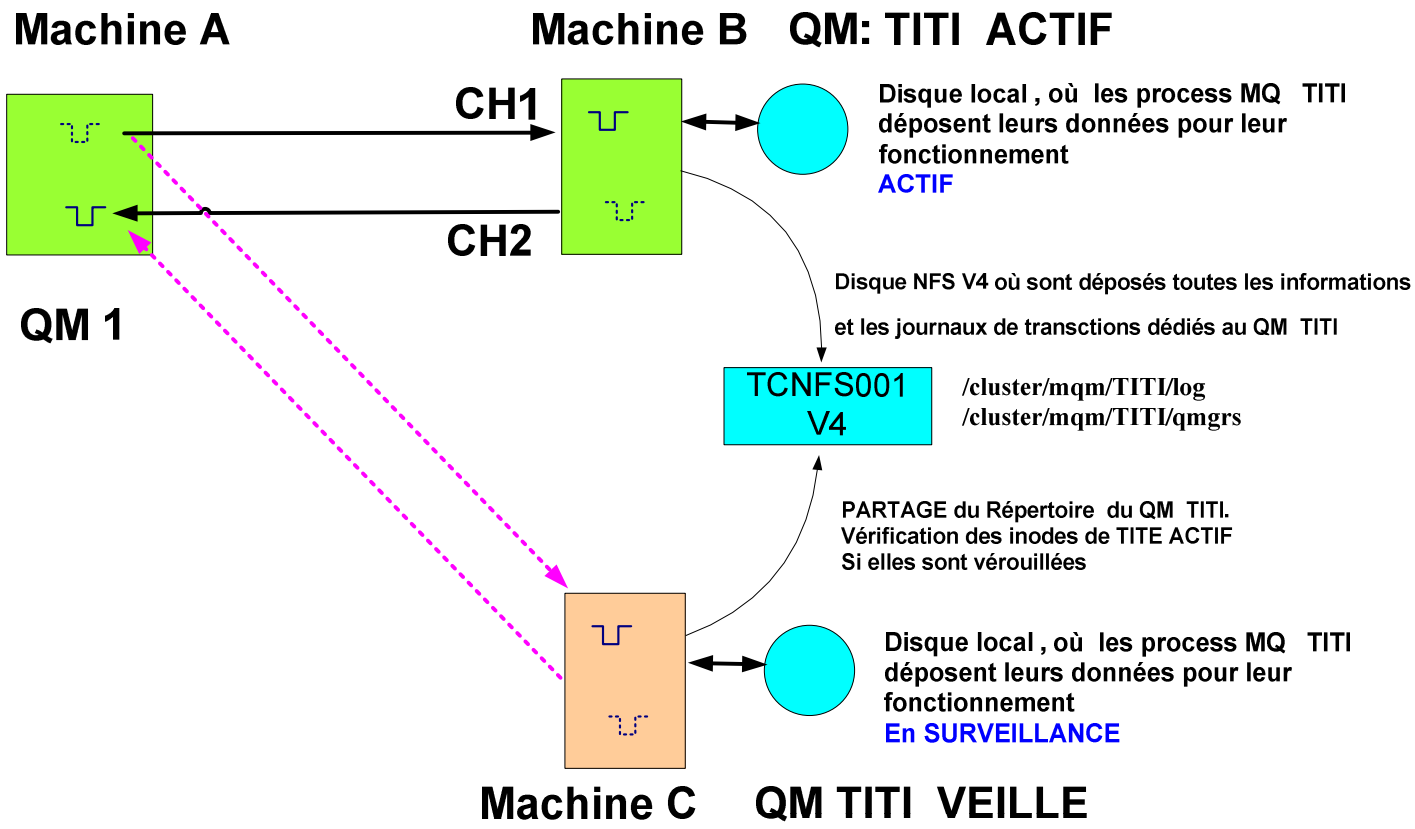
Les deux machines voyent bien le QM TITI , mais il n est pas surveillé par la machine C , car le statut est *STATUS (Running elsewhere)*

Création de la configuration du Q MANAGER secondaire

- **Démarrage du Q MANAGER sur la machine C ; serveur en VEILLE :**

  [mqm@]$ strmqm -x TITI

  WebSphere MQ queue manager 'TITI ' starting.



Commandes liées au Q MANAGER

- start en Haute Dispo a faire sur les deux machines:

  strmqm -x TITI

- Switching du serveur primaire vers le secondaire.

  Sur la machine primaire

  endmqm –s

- Avec 1000 messages de 4K cela prend moins de 30 secondes

- rajout dans /cluster /mqm/qmgrs/titi/mq.ini pour plus de 4096 canaux pour le QM TITI

Channels:

MaxChannels=8192

MaxActiveChannels=8192