Installation de Linux sur une grappe de PC

Simon Derr

ID-IMAG Équipe Apache

Grappes 2001

Laboratoire Informatique et Distribution

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

Institut National Polytechnique de Grenoble

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

GRENOBLE 1 UNIVERSITÉ JOSEPH FOURIER SCIENCES,TECHNOLOGIE,MÉDECINE

Plan

- Problématique
- Diffusion des données
- Automatisation de l’installation
- Notes sur les performances
- Conclusion

Installation des machines d’une grappe

Problématique :

Installer

- Un système de fichiers de grande taille
- Éventuellement un secteur de boot
- Des fichiers de configuration

- Sur un grand nombre de machines à la fois
- Dans un temps relativement court
  - Diffusion des données efficace à travers le réseau
- Avec un minimum d ’interventions
  - Limiter les accès physiques au matériel

Diffusion des données

Très grande quantité de données (ordre du Go), seul le débit de la méthode importe.

- Multicast : exemple Ghost
  - Performances correctes, 120 Mo/minute vers 100 nœuds (réseau Fast Ethernet commuté)
- Par étapes, arbre binomial : exemple : Compaq CMU
  - Un peu plus lent pour le nombre de machines concerné (6 à 8 étapes nécessaires)
- Arbre sans étapes : débit=100mbs/arité
  - Avec arité 1 : débit ~= 10Mo/s
  - En théorie débit optimal
  - On va créer une chaîne avec les machines et y envoyer les données à diffuser

Démarrage des machines

- Pas encore de système d ’exploitation sur le disque dur (ou pas le bon)
- Possibilités de démarrage:
  - Disquette/ CDROM : peu pratique, nécessite à priori une intervention, pas toujours disponible sur une grappe
  - Carte réseau : protocole PXE, permet de télécharger puis d ’exécuter un programme
    - Utilisation de DHCP, TFTP
    - Chargement de bpbatch

Bpbatch

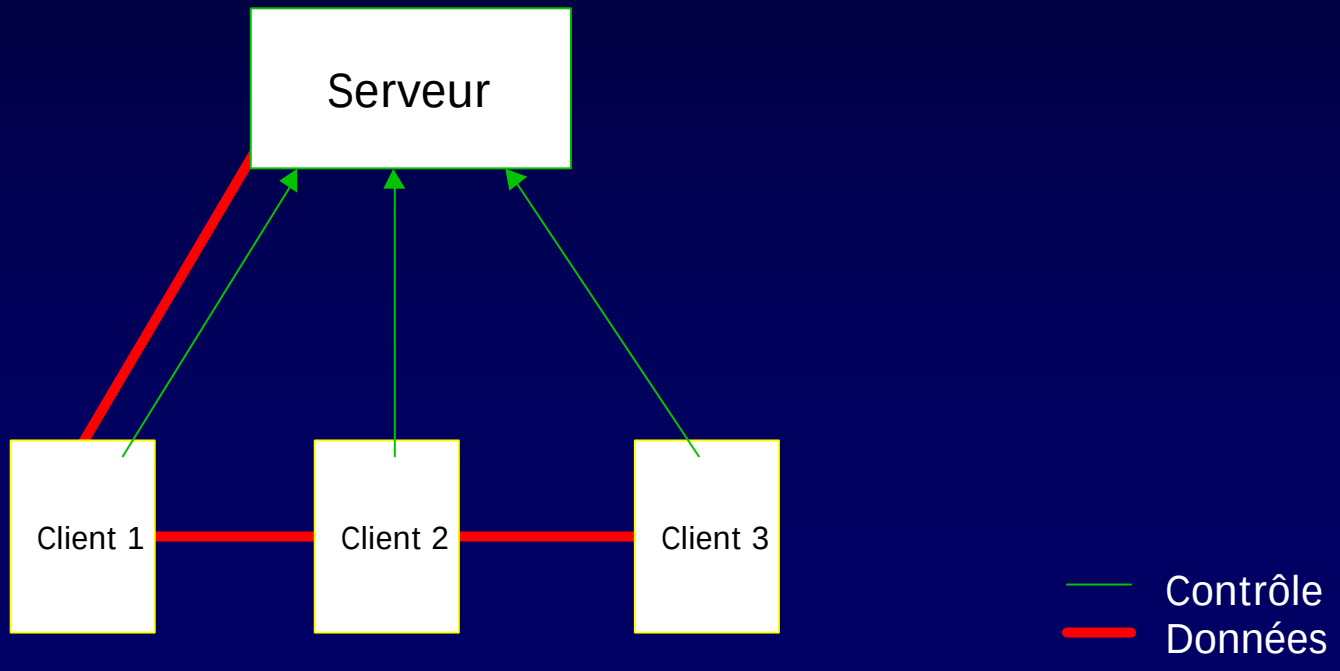
- Programme chargé par PXE
- Exécute un script
- Capable de démarrer la machine de plusieurs façons
  - Sur le disque dur
  - Charger un noyau Linux et l ’amorcer
- Peut effectuer des tests et des branchements conditionnels

Démarrage de la machine

- Bpbatch charge un noyau Linux, avec le paramètre nfs_root : mini système de fichiers sur un serveur distant
- Ce système effectue partitionnement/formatage du disque dur, puis récupère les données et les écrit sur le disque

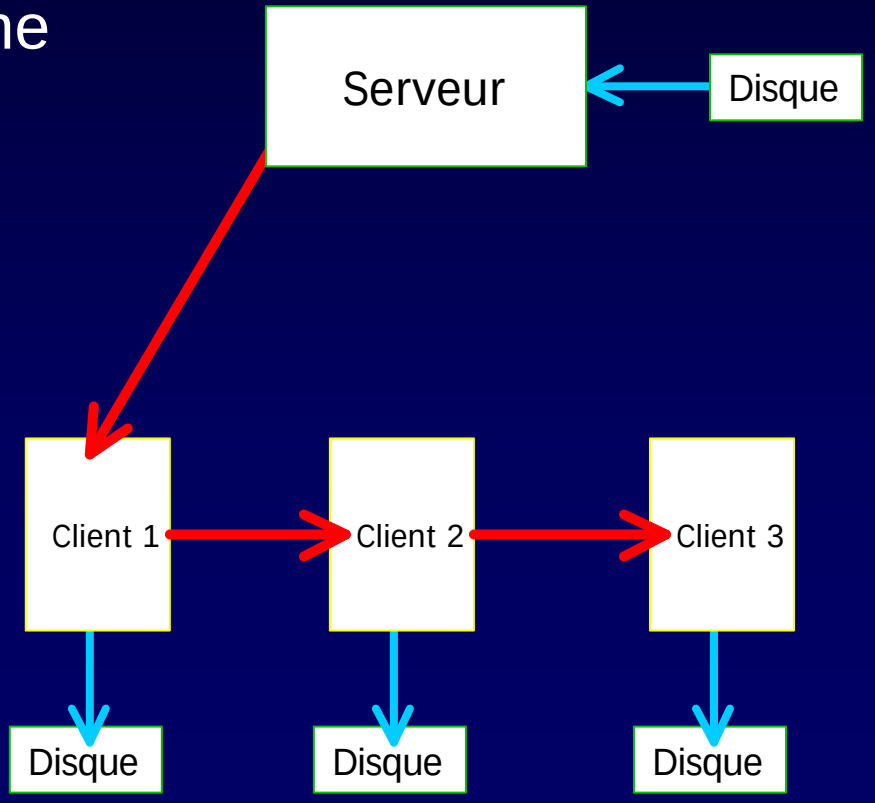
Récupération des données

- Utilisation d’une chaîne
- Les clients demandent à un serveur connu l’IP de la fin de la chaîne et s ’y connectent



Récupération des données

- Quand la chaîne est formée, le serveur envoie les données à travers la chaîne
- Chacun des clients lit les données en provenance de la chaîne, les écrit sur son disque et les envoie au reste de la chaîne



Récupération des données

- Quand toutes les données sont passées, la machine redémarre
  - Problème : comment empêcher l ’installation de recommencer au début ?

Fichier d’état

- Lors des différentes étapes de l ’installation, utilisation d ’un fichier d’état
  - Présent sur un serveur distant, accédé par TFTP
    - Peut être modifié sans intervention sur le noeud
  - Nommé en fonction de l ’IP de la machine
    - Accessible par bpbatch
  - Lu avant de prendre une décision
    - Choix de la méthode de démarrage de la machine
  - Écrit après une opération
    - Écriture de fichiers de configuration et du secteur de boot lors du premier démarrage du système installé

Installation en 3 étapes

- Premier démarrage : Installation
  - le fichier d'état contient « install »
  - démarrage de Linux avec nfs_root, récupération des données
- Deuxième démarrage : mise à jour du système
  - le fichier d'état contient « lilo »
  - démarrage de Linux, utilisation du système installé
    - mise à jour de fichiers de configuration
    - installation du secteur de boot (LILO)
- Troisième démarrage
  - le fichier d'état contient « ready »
  - démarrage sur le disque dur, le système est prêt

Performances

- Réseau sous-utilisé en général, performances limitées par les disques
- Débit variable en fonction de la taille des fichiers : de 11Mo/s pour les gros fichiers à 500ko/s pour les petits
- Résultats très variables et inattendus :
  - Un même test peut prendre 290 secondes pour 89 nœuds et 360 secondes pour 10 nœuds (?)

Performances

- Embouteillages de données ?
  - L ’écriture de petits fichiers (lente) vers la fin de la chaîne semble ralentir l ’écriture de gros fichiers (rapide) sur les premières machines de la chaîne
- Ajout de tampons peu concluant :
    - Linux a déjà ses propres tampons
    - Augmente l ’écart entre les machines
    - Temps largement réduit pour les premières machines, mais relativement peu pour les dernières
    - Mais l ’écart peut être bénéfique : redémarrage des machines moins agressif envers les serveurs
- 1,9 go vers 89 machines en 290 secondes

Améliorations possibles

- Ne pas utiliser la commande tar : lecture brute du disque dur, permettrait un meilleur débit pour les petits fichiers et un débit plus constant en général
- Prise en compte de la topologie du réseau dans la construction de la chaîne
- Gestion des pannes : si une machine tombe
  - Reconstruction de la chaîne
  - Reprise du transfert où il s ’est arrêté

    $\rightarrow$ Nécessité d ’un tampon

Conclusion

- Réinstallation de toutes les machines d’une grappe en 10 minutes
- Solution fonctionnelle, mais activement développée

- Notre outil de diffusion de données:
  `http://ka-tools.sourceforge.net`
- Bpbatch :
  `http://www.bpbatch.org`