Architecture Distribuée

Cours n°2

A.Saval

Objectifs du cours “Architectures distribuées”

- Compréhension des motivations
- Compréhension de la logique de conception d'une architecture distribuée
- Maîtrise des principaux modèles
- Aperçu des problèmes posés
- Aperçu de quelques frameworks existants

Aperçu du cours

- Introduction
- Problème de conception d'architecture
- Architecture logique & matérielle
- Système distribué
- Modèles d'architecture
  - Client/serveur
  - 3-tiers
  - N-tiers
  - Pair à pair
  - Virtualisation

SYSTEME DISTRIBUE

Définition(s)

Distributed system (Tanenbaum, van Steen):

> A collection of independent computers that appears to its users as a single, coherent system.

Distributed system (Lukkien):

> The hard- and software of a collection of independent computers that cooperate to realize some functionality.

Distributed system (Lamport):

> A system in which the failure of a computer you didn't even know existed can render your own computer unusable.

Quelques caractéristiques

- “Sert” à des utilisateurs mais les utilisateurs ne sont pas forcément impliqués directement
  - Ex: DNS, DHCP
- Indépendant :
  - Appel concurrent
  - Robuste aux pannes
  - Pas d'horloge partagée
  - Autonome
  - Hétérogène (capacités/support/hardware)
  - Distribué spatialement
- Seule, chaque partie du système distribué n'est pas capable de réaliser la fonction globale

Intérêts des systèmes distribués

- Par besoin propre :
  - Distribution physique de l'environnement de production (une contrainte “de facto”)

    séparation physique des données et des clients d'accès aux données (partitionnement administratif ou encore besoin de sécurité qui justifie la séparation physique)
  - Distribution physique des resources : Internet

Intérêts des systèmes distribués

- Comme une solution à un problème :
  - Introduction de distribution pour addresser des fonctionnalités supplémentaires
    - Accès concurrent – performance
    - Réplication – robustesse
    - Sécurité – séparer les contrôle d'accès de la consultation standard
    - Modularité, spécialisation – facilité de conception

Intérêts complémentaires

- Partage et développement distribué
  - Partage des fonctionnalités et de l'expertise associée : Application basée sur des fonctions, distribuées existantes, Dévelopement et MCO indépendant
  - Partage de resources/données : Stockages, BDD, capacité de traitement centralisée
  - … évident quand la distribution est une contriante de l'environement
- Transparence
  - Cache l'implémentation (complexe) des sous-systèmes
  - … c'est aussi une conséquence de l'AD

Intérêts complémentaires

Passage à l'échelle (scalability)

- En volume: beaucoup de clients, de serveurs, de données...
- Échelle géographique: distribution des serveurs, des clients, mobilité...

- Performance
  - Parallélisation des traitements lourds
  - Distribution équilibrée des stockages, des communications, des traitements...

ADs et transparence

Transparency	Description
Access	Hide differences in data representation and how a resource is accessed
Location	Hide where a resource is located
Migration	Hide that a resource may move to another location
Relocation	Hide that a resource may be moved to another location while in use
Replication	Hide that a resource may be shared by several competitive users
Concurrency	Hide that a resource may be shared by several competitive users
Failure	Hide the failure and recovery of a resource
Persistence	Hide whether a (software) resource is in memory or on disk

Passage à l'échelle

… capacité d'un système à “augmenter ses capacités”...

- Quelles capacités ?
- Quel système ?
- Passage à l'échelle en largeur ? En profondeur ?
- Rapport au parallélisme ?

Passage à l'échelle : analyse (1)

- Paramètres du passage à l'échelle : k
  - Peut recouvrir de nombreuses variables différentes et diverses dans l'ensemble du système considéré
    - Ex : nombre de serveurs, de processeurs, de clients, de requêtes, de jours de fonctionnement...
- Métrique du passage à l'échelle : m(k)
  - Mesure les performances/capacités du système à l'échelle 'k'
    - Ex : temps de réponse, occupation de la bande passante, coût...

Passage à l'échelle : analyse (1)

- Critère/objectif/limite de passage à l'échelle : Z(k)
- La capacité de passage à l'échelle : une relation.
  - Ex : m(k)<Z(k) ; m(k)~=Z(k) ; m(k)>Z(k)
- Z(k) n'est souvent pas explicite en comparaison de systèmes
  - Ex : m1(k) > m2(k) => le système 1 passe mieux à l'échelle que le système 2

    le système X “ne passe pas l'échelle” => mX(k) a une forme “décourageante”

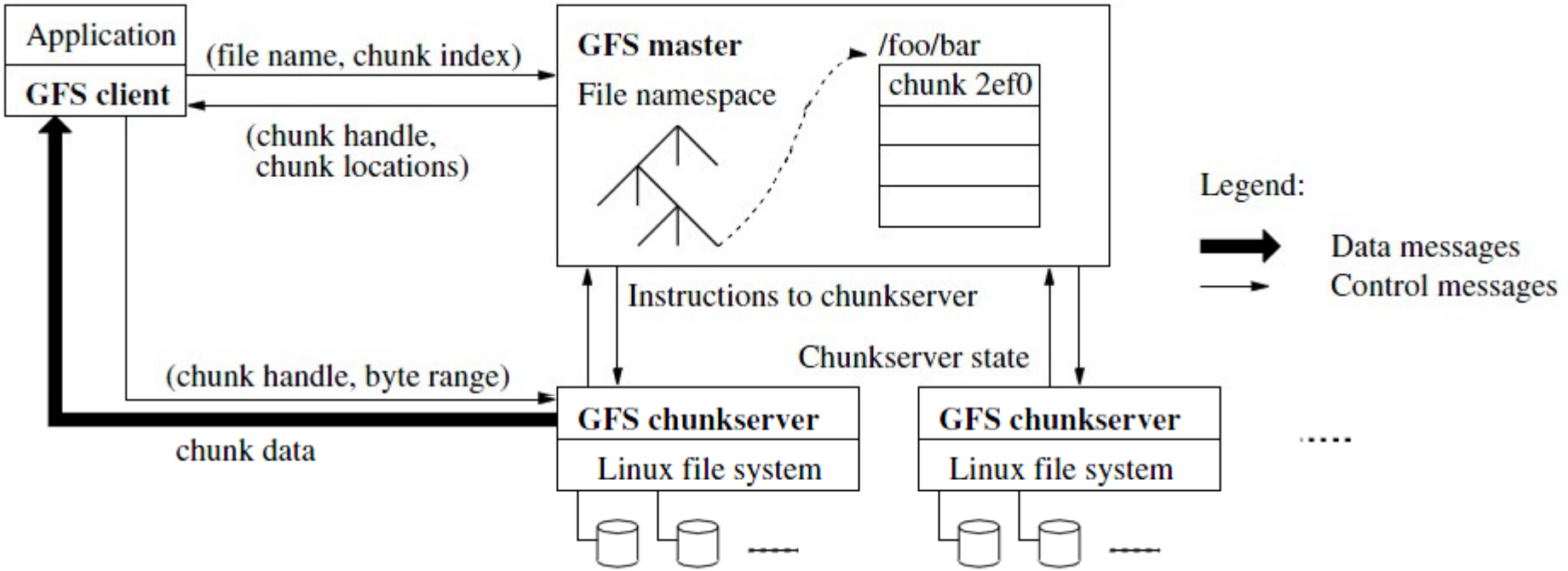
Exemples (1)

- Système de fichier vs taille des disques
  - Sur un ordinateur personel
    - UFS, extN, NTFS...
  - En réseau
    - CODA, NFS, ...
  - Base de données (BDD) traditionnelles
    - MySQL, Posgresql, ...
  - Stockage à grande échelle
    - Google File System (GFS), Hadoop, ...
  - Bases de données NoSQL
    - MongoDB, Cassandra...

Exemple (2)

- Google File System (GFS)

  Objectifs : gestion d'un (très) grand système de (gros) fichiers avec un nombre de clients importants sur des machines à faible coût.

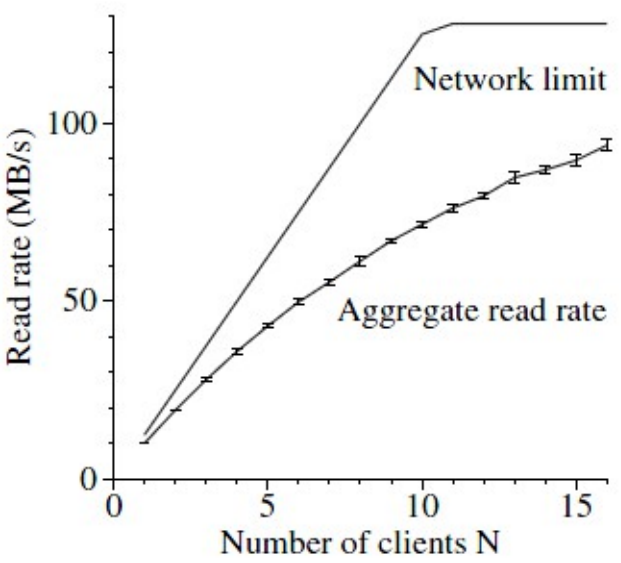


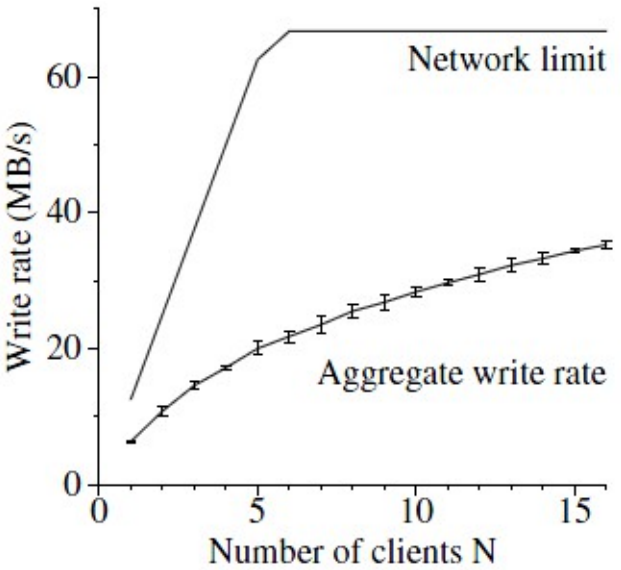
Picture from 'The Google File System', by Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, published at http://labs.google.com/papers/gfs-sosp2003.pdf (download June 2011)

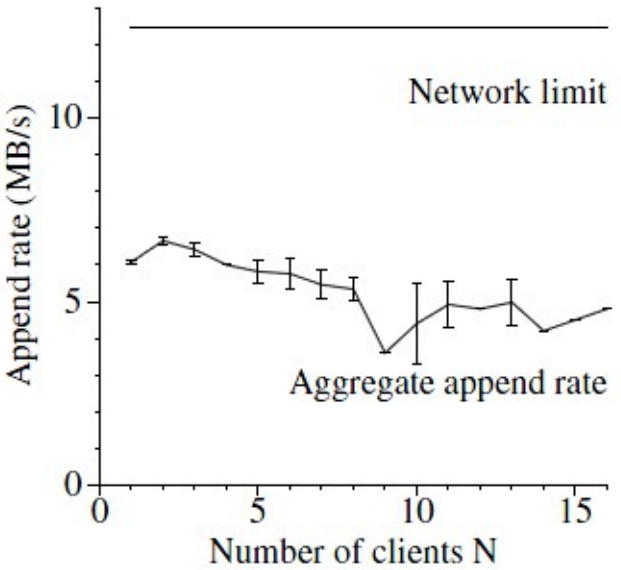
MAGNETEK
Made in China
Google P/N: 1502103
Input: 200 - 240VAC 50 - 60Hz
Output : + 13.65V/20.5A
HITACHI Deskstar
GIGABYTE
GA-9IVDP

Exemples (3)

- Nombreux paramètres pour le passage à l'échelle :
  - k = nb clients
  - m(k) = vitesse de lecture (ou écriture) en MB/s
  - Z(k) = limite des capacités réseaux

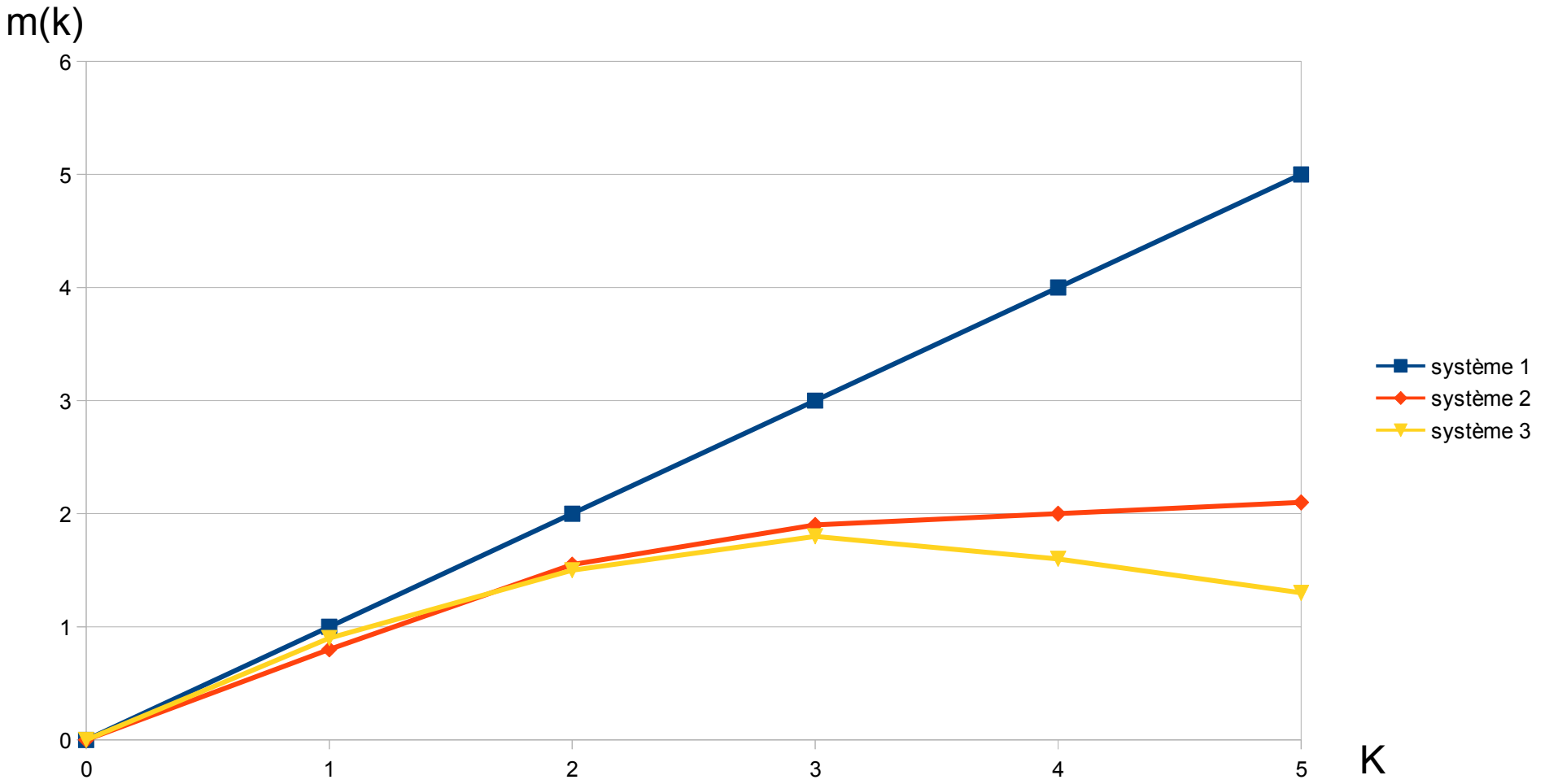






Picture from 'The Google File System', by Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, published at http://labs.google.com/papers/gfs-sosp2003.pdf (download June 2011)

Exemple (4)



Passage à l'échelle : analyse (2)

/!\ l'analyse des problèmes de passage à l'échelle doit être faite :

- Toujours vis-à-vis d'une métrique !
- En prenant en compte les bons facteurs (tous !)
- En comparaison à une échelle de référence k0

  ex : m(k) / m(k0)

- Métrique du gain réel :

  v(k) = mesure effective de m(k) en “production”
- Coût du passage à l'échelle : C(k)

  ex : coût de l'ajout de serveurs, de processeurs...
- Evaluation de la rentabilité, du rapport qualité/prix, de l'efficacité

  (value for money)

  ex : v(k)/C(k)

Passage à l'échelle : analyse (3)

- Loi d'Amdahl :

  “La loi d'Amdahl, énoncée par Gene Amdahl, exprime le gain de performance qu'on peut attendre d'un ordinateur en améliorant une composante de sa performance.” (wikipedia)

- Vue dans le contexe :

  “On peut améliorer les performances d'une architecture en distribuant des traitements... mais il y a des limites.”

- Si on a :
  - “N” la dimension du problème
  - “P” le nombre de serveurs (ou processeurs)

  → m(P,N) = seq(N) + par(N)/P

Passage à l'échelle : conclusions (1)

- L'amélioration d'un sous-système d'un système aura un impact limité
- Si on se focalise sur un sous-système, l'amélioration sera limitée
- Lorsque l'amélioration d'un sous-système rend son “coût” suffisament faible : aller voir ailleurs ! (c'est à dire améliorer un autre sous-système)

  Ex :

  - Éviter les points d'étranglement comme un serveur central
  - Communication en temps masqué
  - Améliorer l'efficacité du code

Passage à l'échelle : conclusions (2)

- Quelques règles de base :
  - Limiter les dépendances entre les sous-systèmes
    - Problème sur un serveur ne bloque pas tout
    - Chaque serveur traite les données indépendamment
    - Privilégier les traitements en temps masqué
  - Favoriser les traitements paralèlles
    - Construction d'objet en paralèlle (ex : une page Web + AJAX)
    - Transformer ou répliquer les données à traiter
    - Distribuer les traitements sur des serveurs différents

MODELES D'ARCHITECTURE

Références (one more time)

- Software Architecture: IEEE Standard 1471-2000
- P. Kruchten, Architectural Blueprints—The “4+1” View Model of Software Architecture, IEEE Software 12 (6), Nov. 1995, pp42-50
- Tanenbaum & van Steen, Distributed Systems, Principles and Paradigms, seconde édition
- Architecture of Distributed Systems, cours de Johan Lukkien, 2011
- Architectural Patterns Revisited – A Pattern Language, Paris Avgeriou & Uwe Zdun, 2005
- Software Architecture, Foundations, Theory, and Practice, R.N. Taylor, N. Medvidovic, E.M. Dashofy, Wiley & Sons, 2009
- Software Architecture in Practice, Second Edition, L. Bass, P. Clements, R. Kazman, SEI Series in Software Engineering, Addison-Wesley, 2003