

Linda++

Projet Données Réparties (S8)

11 février 2022

Plan

- 1 Réalisé
- 2 Objectifs
- 3 Améliorer les performances des applications Linda
- 4 Evaluer les gains de performance
- 5 Instrumentation
- 6 Organisation

Ce qui a été fait

Interface

- `write(tuple)` : dépôt
- `take(motif)` / `tryTake(motif)` / `takeAll(motif)` : retrait d'un (des) tuple(s) correspondant au motif
- `read(motif)` / `tryRead(motif)` / `readAll(motif)` : recherche d'un (des) tuple(s) correspondant au motif
- `eventRegister(mode, timing, motif, callback)` : abonnement à l'événement d'existence/de dépôt d'un tuple correspondant au motif.

Implémentations

- version en mémoire partagée au sein d'une même JVM
- version client/serveur (le serveur gérant l'espace de tuples)



version centralisée (noyau et applications dans la même JVM)

structures de données (SDD)

- liste de tuples
- listes de requêtes en attente : lecture, écriture, abonnements

difficultés

- concurrence d'accès aux SDD
 - gestion des conflits
 - interblocages
 - (peut n'apparaître qu'avec la version C/S)
 - efficacité du réveil
 - accès au contenu des tuples
 - stockage
 - comparaison

version client/serveur

principe

noyau centralisé rendu accessible aux applications distantes par interposition d'un mandataire (talon) côté client et côté serveur

difficultés

- (désignation)
- notion d'objet accessible à distance (Java : référence = stub)
callbacks distants acrobatiques
- terminaison des applications

Bilan

- noyau opérationnel (en gros et en général)
- abonnement et callbacks +/-
- mal testé
- performances faibles en général
(verrou global sur l'espace de tuples)s
- applications incomplètes
- outillage et instrumentation très limitées

Plan

- 1 Réalisé
- 2 Objectifs
- 3 Améliorer les performances des applications Linda
- 4 Evaluer les gains de performance
- 5 Instrumentation
- 6 Organisation

Objectifs

- (Consolider le noyau existant)
- Améliorer l'efficacité du noyau Linda
- réalisation d'une version répartie multi-serveurs
- Motiver et évaluer les améliorations
 - démarche
 - outillage
 - étude

Plan

- 1 Réalisé
- 2 Objectifs
- 3 Améliorer les performances des applications Linda**
- 4 Evaluer les gains de performance
- 5 Instrumentation
- 6 Organisation

Améliorer les performances des applications Linda

→ réduire les temps d'exécution des applications Linda

Améliorer les performances des applications : parallélisation

- gestion des tâches
- architecture (algorithmique)

Améliorer les performances du noyau

- travail sur les **structures de données** pour faciliter
 - la localisation ou l'accès concurrent aux tuples
 - l'évaluation des requêtes ou des requêtes en attente
- **paralléliser** le noyau
 - externe : accès concurrents à l'espace de tuples
 - interne : parallélisation des opérations
 - limite : nombre de cœurs
- **réduire la charge** du noyau
 - *caches* de tuples chez les clients
- **répartir** le noyau : Linda multi-serveurs
 - *partitionner* l'espace de tuples

Gestion de caches (version client-serveur)

Principe

Chaque client conserve dans un cache local les tuples lus ou écrits

- réduit la charge du serveur
- réduit les temps d'accès aux tuples en caches

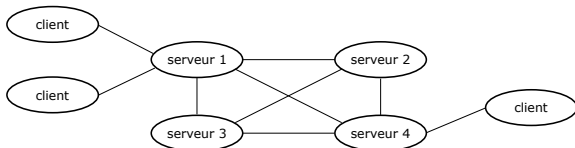
À faire

- adapter le client Linda pour interroger le cache avant le serveur
- gérer la cohérence des caches : les tuples d'un cache doivent toujours être des tuples du serveur
 - invalider (supprimer) les tuples retirés du serveur
 - protocole à développer
 - supprimer exactement (égalité, et non correspondance) les tuples retirés du serveur
 - méthode à développer
 - (simplifier) ne pas conserver de tuples identiques dans un cache

Difficultés

- risques d'interblocage lors de l'accès au cache
- risques d'incohérences lors de la modification des caches

Linda multi-serveurs



- l'espace des tuples est partitionné entre les serveurs
- chaque client est connecté à un unique serveur
- les serveurs sont tous interconnectés entre eux

Protocole

- le client Linda est inchangé
- Quand un client dépose un tuple, le dépôt se fait dans l'espace du serveur auquel il est connecté.
- Quand un client cherche un tuple, la recherche se fait sur le serveur auquel il est connecté
 - En cas d'échec, le serveur propage la requête aux autres serveurs

Plan

- 1 Réalisé
- 2 Objectifs
- 3 Améliorer les performances des applications Linda
- 4 Evaluer les gains de performance**
- 5 Instrumentation
- 6 Organisation

Evaluer les gains de performance

Les performances dépendent du **profil** des applications

- dominées par les accès à l'espace partagé : nombres premiers
- dominées par les calculs : recherche de mots proches

→ pour les besoins de l'étude : développer une application de test (benchmark) permettant de

- faire varier le nombre et le type d'accès
- faire varier les temps d'accès ou de calcul (temporisations)

Démarche

- identifier les paramètres et les critères
- mesures
 - effet attendu
 - pour différents profils d'applications
 - comparaison éventuelle avec une version de référence du noyau (version initiale)
- interprétation
- proposition (éventuelle)

Exemple : degré de parallélisme, pour le calcul des nombres premiers

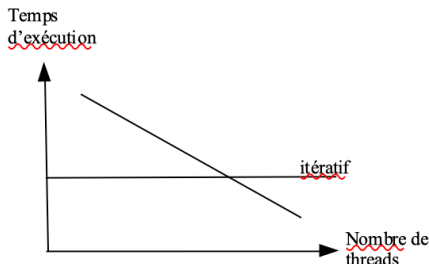
A priori

- le noyau a un surcoût d'exécution dû à la gestion des tuples
- une exécution parallèle permet d'accélérer les calculs

	itératif	Linda
<u>surcoût</u>	0	- - -
<u>accélération</u>	0	+ + +

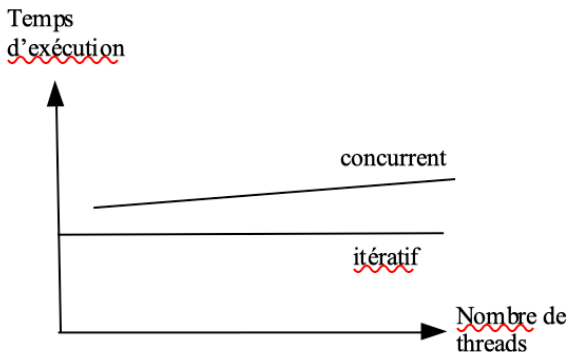
Lequel est le plus grand

Mesures critère : temps d'exécution, paramètre : nombre de threads
Effet attendu



Exemple : degré de parallélisme, pour le calcul des nombres premiers

Observation



Analyse (pourquoi ?)

Proposition ? ?

Plan

- 1 Réalisé
- 2 Objectifs
- 3 Améliorer les performances des applications Linda
- 4 Evaluer les gains de performance
- 5 Instrumentation**
- 6 Organisation

Instrumentation

- interpréteur de commandes
 - jouer des scénarios interactifs ou prédéfinis
 - évaluer les performances des applications exécutées
 - consulter (et éventuellement fixer) des paramètres relatifs aux ressources utilisées (par site)
 - taille (maximale ou courante) de l'espace de tuples,
 - nombre de tuples
 - état des files d'attente
 - état des processus (actifs, attente)
 - volumes des échanges
- instrumentation du noyau pour permettre les mesures

Plan

- 1 Réalisé
- 2 Objectifs
- 3 Améliorer les performances des applications Linda
- 4 Evaluer les gains de performance
- 5 Instrumentation
- 6 Organisation**

Organisation

- voir Moodle
 - calendrier, nouvelles, fournitures, dépôt
 - ! : nouvelle matière - il faut s'inscrire
- Groupes
 - même base qu'au premier semestre
 - ajustements possibles : départs/arrivées au S8
 - ventilation des groupes ayant mal fonctionné au S7
- Séances
 - 3(+/- 1) séances de suivi
 - 1 séance de restitution
- Livrables
 - Linda++ (parallélisme + cache) [Bonus : multi-serveurs]
 - Outillage
 - supervision
 - aide au déploiement et à la configuration
 - Etude expérimentale et analyse de performances
 - améliorations apportées et effet attendu
 - étude et analyse critiques