Institut de Mathématiques de Bordeaux

Les ressources de calcul et leur utilisation

- **Plateformes de calcul :**
  - **Ressources locales :**
    PlaFRIM (Plateforme Fédérative pour la Recherche en Informatique et Mathématiques)
    https://plafrim.bordeaux.inria.fr
  - **Ressources régionales :**
    MCIA (Mésocentre de Calcul Intensif Aquitain)
    http://www.mcia.univ-bordeaux.fr
  - **Ressources nationales :**
    CINES (Centre Informatique National de l'Enseignement Supérieur)
    http://www.cines.fr

- **Notions de base sur la programmation parallèle**

- **SVN : logiciel de gestion de versions**

23 et 24 octobre 2012

Khodor KHADRA, Ingénieur de Recherche Calcul Scientifique

Compte PlaFRIM

- **Mutualisation des moyens de calcul : Institut de Mathématiques de Bordeaux (IMB), Laboratoire Bordelais de Recherche en Informatique (LaBRI), Centre de Recherche INRIA Bordeaux – Sud-Ouest**

  https://plafrim.bordeaux.inria.fr

- **Condition nécessaire pour avoir un compte PlaFRIM : avoir un compte IMB**

- **Générer depuis son compte IMB sa clé ssh protégée par une *« passphrase »* :**

  http://www.math.u-bordeaux1.fr/imb/cellule/spip.php?article190#outil_sommaire_2

- **Demande de création d'un compte, remplir le formulaire en ligne :**

  https://dihpes.bordeaux.inria.fr/faccount

Passerelle d'entrée pour accéder à l'ensemble des ressources de calcul

- Pour l'IMB, il s'agit de la machine mygale = devel01

- Depuis son poste de travail au bureau :

  ssh -Y id_plafrim@mygale (avec la passphrase)

- Depuis l'extérieur sur son ordinateur portable ou fixe de domicile :

  ssh -Y id_bdx1@acces.math.u-bordeaux1.fr (avec le mot de passe IMB)

  ssh -Y id_plafrim@mygale (avec la passphrase)

- L'option -Y permet d'obtenir le déport graphique (elle n'est pas obligatoire)

- ATTENTION : les deux comptes IMB et PlaFRIM sont disjoints et en général ils ont le même nom d'identifiant id, celui de Bordeaux 1

Espaces de stockage

- https://plafrim.bordeaux.inria.fr/doku.php?id=plateforme:stockage:stockage
- Chaque utilisateur a accès à un certain nombre d'espaces de stockage avec des tailles différentes en Go qui lui sont dédiés
- Par défaut, le répertoire de travail est /home/id_plafrim
- Tous les espaces de stockage sont accessibles sur l'ensemble des machines de calcul
- Choix de l'espace de stockage en fonction de la :
  - taille des données
  - sauvegarde des données
  - rapidité des accès disques au moment de l'écriture des fichiers de résultats

Transfert de fichiers IMB <---> PlaFRIM

- La plupart du temps, comme la taille des données de calcul est relativement grande, elles restent stockées sur le compte PlaFRIM et n'ont pas besoin d'être rapatriées sur le compte IMB pour être exploitées

- Les éventuels transferts de fichiers ou répertoires entre les deux comptes IMB et PlaFRIM se font via la commande scp, par exemple à partir du compte IMB :

```
scp -r id_plafrim@mygale:cheminPlaFRIM/repertoiredonnees cheminIMB/.

scp -r cheminIMB/repertoiredonnees id_plafrim@mygale:cheminPlaFRIM/.
```

Quelques définitions

- Un nœud de calcul = une machine de calcul qui comprend :
  - sa mémoire vive
  - son disque dur local
  - plusieurs processeurs à plusieurs cœurs de calcul chacun
- Un processus est défini par :
  - un ensemble d'instructions à exécuter un programme
  - un espace mémoire pour les données de travail
- Un job de calcul = un ensemble de processus liés à l'exécution d'un code calcul
- Pour un calcul séquentiel, un processus est rattaché à un seul cœur de calcul
- Pour un calcul parallèle, P processus tournent sur C cœurs de calcul, on peut choisir P > C mais il est préférable de choisir P=C, c'est à dire un seul processus par cœur de calcul

Les machines interactives

- https://dihpes.bordeaux.inria.fr/doku.php?id=plateforme:configurations:devel

- Une machine interactive est un nœud à partir duquel on va pouvoir se connecter directement pour écrire, compiler et exécuter un code de calcul séquentiel ou parallèle

- Il y en a dix : devel01, …, devel10

- A partir de mygale = devel01, se connecter sur les autres develxx via

  ssh -Y develxx

- Chaque nœud develxx comprend 24 Go de RAM et 8 cœurs de calcul

Les machines interactives

- devel01, devel09, devel10 sont les passerelles d'entrée respectives pour l'IMB, l'INRIA et le LaBRI
- Il est vivement recommandé de NE PAS effectuer des opérations gourmandes en mémoire et CPU sur ces trois develxx, car saturées elles peuvent s'arrêter, et les utilisateurs perdent ainsi l'accès à toutes les autres machines
- Rien n'empêche P > 8 processus de tourner sur une develxx
- Avoir le réflexe de taper la commande top afin de s'assurer de l'ensemble des processus qui tournent afin d'éviter une surcharge de la machine et de conduire à son éventuel « crash »
- ATTENTION : tous les utilisateurs se partagent l'ensemble des coeurs

Les clusters de calcul

- Un cluster de calcul est un ensemble de Nmax nœuds
- Chaque nœud a Pmax processeurs ayant Cp cœurs de calcul chacun
- Il y a donc Cmax = Pmax * Cp cœurs par noeud
- Pour un cluster donné, il y a donc Nmax*Cmax cœurs de calcul
- Architecture homogène de tous les noeuds d'un même cluster
- Les nœuds communiquent entre eux par l'intermédiaire d'un réseau Infiniband
- Type de calcul dédié : séquentiel et parallèle

Les clusters de calcul

- **Quatre clusters de calcul :**
  - **fourmi (68 nœuds : fourmi001, … , fourmi068)**
    https://plafrim.bordeaux.inria.fr/doku.php?id=plateforme:configurations:fourmi
  - **bonobo (24 nœuds : bonobo001, … , bonobo024)**
    https://plafrim.bordeaux.inria.fr/doku.php?id=plateforme:configurations:bonobo
  - **mirabelle (16 nœuds : mirabelle001, … , mirabelle016)**
    https://plafrim.bordeaux.inria.fr/doku.php?id=plateforme:configurations:mirabelle
  - **mirage (9 nœuds : mirage001, … , mirage009)**
    https://plafrim.bordeaux.inria.fr/doku.php?id=plateforme:configurations:mirage

- **On accède à un cluster de calcul depuis mygale en mode batch, et il y a deux façons de travailler en mode batch**

Le premier principe de base du batch : le batch standard

- Il n'y a pas besoin de se connecter directement sur les nœuds d'un cluster
- Depuis mygale, via un fichier et des commandes dits de batch :
  - choisir le cluster
  - réserver N nœuds (N <= Nmax) à C cœurs par nœud (C < =Cmax)
  - lancer son job sur les N*C cœurs
  - le job est soit en train de s 'exécuter, soit placé en queue dans une file d'attente prêt à être exécuté selon les disponibilités des ressources demandées
  - contrôler l'état du job à tout moment : les nœuds sur lequels ils tournent et le temps écoulé
- Avantage : durant l'exécution d'un job, l'utilisateur est assuré d 'avoir en permanence N*C cœurs réservés à lui tout seul

Le fichier de batch

- https://plafrim.bordeaux.inria.fr/doku.php?id=utilisation:batchs:fichier_de_batch
- On le nomme comme on veut, c'est un fichier texte de quelques lignes
- Il contient des commentaires de texte, des commandes shell UNIX standard et des instructions de batch
- Tout commentaire de texte ou de commande UNIX ne doit pas comporter d'accent et doit être précédé du symbole #
- Toute instruction de batch doit être précédée du symbole #PBS car le logiciel de gestion de batch s'appelle PBS
- On commente une instruction de batch avec le symbole ##PBS
- PBS est le logiciel qui va permettre via des commandes dites de batch d'exécuter le fichier de batch sur les N*C coeurs de calcul d'un cluster

Que précise t-on dans un fichier de batch ?

- Le nom du job
- le nom du cluster sur lequel le job va tourner, par défaut fourmi quand on ne précise pas le nom
- le temps maximum de restitution du calcul en heures, minutes, secondes
- la mémoire en MégaBytes (Mb) ou GigaBytes (Gb) requise par l'ensemble des processus de calcul
  - en mode séquentiel monocoeur, un processus est attaché à un cœur
  - en mode parallèle multicoeurs, on attache souvent un processus par cœur
- dans le cas d'un code parallèle en MPI, le nombre de nœuds N et de cœurs C
- une suite de commandes shell UNIX standard comme en mode interactif (positionnement dans le répertoire de travail, compilation, exécution, …)

Classes de batch à l'exécution d'un fichier de batch

- https://plafrim.bordeaux.inria.fr/doku.php?id=utilisation:batchs:queues_de_routage

- A l'exécution d'un fichier de batch , le job est :
  - rattaché à un numéro attribué par le gestionnaire de batch
  - placé dans une classe de batch en fonction du temps de calcul et du nombre de nœuds demandés
  - soit placé dans une file d'attente plus ou moins prioritaire en fonction des ressources demandées (nombre de cœurs et temps de calcul) et celles disponibles à un instant donné sur le cluster (mode Queue)
  - soit en train de s'exécuter sur les N*C cœurs avec le temps écoulé (mode Run)

Quelques commandes de batch à exécuter depuis mygale

- https://plafrim.bordeaux.inria.fr/doku.php?id=utilisation:batchs:commandes_de_batch

- **Exécuter un fichier de batch :**

  `qsub monfichier_batch`

- **Voir l'état de tous les jobs sur tous les clusters :**

  `qstat -n`
  `qstat -a`

- **Voir uniquement l'état de ses jobs sur tous les clusters :**

  `qstat -u id_plafrim`

- **Supprimer un job :**

  `qdel numero_job`

Quelques conseils

- Si vous avez estimez un temps trop court, le job tourne et s'arrête à la fin du temps que vous avez demandé (le job est « tué »)
- Si vous avez estimez une mémoire insuffisante par rapport à votre jeu de données, le job s'arrête immédiatement (le job est « tué »)
- On peut estimer la mémoire utilisée en ayant fait tourner au préalable sur un jeu de données de taille raisonnable le code sur une machine interactive, repéré via la commande top dans la colonne RES la mémoire allouée, et extrapolé ainsi sur des tailles de données plus grandes
- Prévoyez un supplément de 20% par rapport à vos estimations temps/mémoire

Quelques conseils

- **Ne pas donner un temps de calcul et une mémoire très élevés si votre code ne requiert pas ces ressources. Vous risquez de vous bloquer dans les priorités des files d'attente des jobs, et de monopoliser les ressources du cluster inutilement**

- **Avant de lancer un job en mode batch sur un cluster, s'assurer auparavant qu'il a bien été compilé et que l'exécutable a bien été généré sur une machine interactive develxx**

A la fin du déroulement d'un job en mode batch

Deux fichiers de sortie sont générés dans le répertoire où est exécuté le fichier de batch :

- le fichier nom_job.onumero qui contient la sortie standard (print ecran) ; ce qui signifie que l'on n'est pas obligé de générer cette sortie standard dans un autre fichier puisqu'elle se fait automatiquement

- Le fichier nom_job.enumero qui contient les temps de calcul (quand l'exécutable du code est lancé avec la commande time) ainsi que les éventuels messages d'erreur si le code de calcul ne s'est pas déroulé comme prévu jusquà la fin

Deuxième possibilité du batch : le batch interactif

- https://plafrim.bordeaux.inria.fr/doku.php?id=utilisation:batchs:travailler_en_interactif#le_choix_des_ressources
- **Idée : ne pas soumettre un job de manière “différée” en réalisant un fichier de batch mais travailler de manière “interactive” sur plusieurs nœuds d'un cluster**
- **Réserver un ou plusieurs nœuds d'un cluster avec plusieurs coeurs, et travailler de la même manière que sur les machines develxx**
- **Ce n'est pas vraiment du batch, c'est du mode interactif, mais la réservation des noeuds et cœurs utilise le gestionnaire de batch PBS, d'où le nom « batch interectif » même si ce nom peut prêter à confusion**
- **Avantage comme en batch standard : durant l'exécution d'un job, l'utilisateur est assuré d 'avoir en permanence N*C cœurs réservés à lui tout seul**

Deuxième possibilité du batch : le batch interactif

- **A partir de mygale, taper la commande :**

  qsub -I -Y -qclustername -l nodes=N:ppn=C -l walltime=h:m:s -l mem=xxxxM(G)b

  - **L'option -I permet de demander une soumission interactive**
  - **L'option -Y permet d'obtenir le déport graphique (elle n'est pas obligatoire)**
  - **L'option -qclustername est à préciser si le nom du cluster est autre que fourmi**
  - **N et C : nombre de noeuds et de cœurs demandés**
  - **h, m, s : nombre d'heures, minutes et secondes du temps maximum d'exécution du code**
  - **xxxx : mémoire en Mégabytes ou Gigabytes**

Deuxième possibilité du batch : le batch interactif

- **Lorsque l'ensemble des ressources demandé est disponible, on attérit sur un des nœuds réservés fourmixxx, ou bonoboxxx ou mirabellexxx ou miragexxx**

- **Attention : pas de fichiers de sortie .o et .e générés comme en mode batch standard**

Voir la charge de l'ensemble des ressources

- https://plafrim.bordeaux.inria.fr/ganglia
- https://plafrim.bordeaux.inria.fr/monika

Choix du nombre de nœuds et de cœurs en mode batch

- On souhaite faire tourner un code parallèle sur un nombre total de cœurs CT

  Question :
  comment répartir les CT cœurs sur un ensemble de N nœuds avec C cœurs par nœud, c.a.d comment choisir de façon optimale N et C tels que N*C = CT ?

- Cmax étant le nombre de coeurs total disponibles par nœud, le plus optimal est de choisir C le plus proche possible de Cmax, mais la règle n'est pas aussi simple car elle dépend de :
  - la charge des ressources (les Cmax cœurs ne sont pas disponibles par nœud)
  - la mémoire vive requise (par exemple on peut être ramené à réserver plus de nœuds avec juste quelques cœurs par nœud)

Exclusivité des nœuds en mode batch

- **ATTENTION : par défaut les nœuds ne sont pas exclusifs, c'est à dire qu'il est possible d'avoir plusieurs jobs en même temps sur un noeud**
- **Si vous avez besoin d'utiliser des nœuds de manière exclusive, dans votre commande qsub vous devez rajouter l'option :**

  -W=NACCESSPOLICY:SINGLEJOB

- **Par exemple :**

  qsub -W=NACCESSPOLICY:SINGLEJOB monfichier_batch

  qsub -I -Y -l nodes=N -W=NACCESSPOLICY:SINGLEJOB -l walltime=h:m:s -l mem=xxxxM(G)b

- **Cela vous assure d'être le seul à utiliser l'intégralité des nœuds pendant le déroulement du job**
- **Cela peut être utile quand on veut être seul à exploiter l'intégralité de la RAM d'un nœud**

Mode Interactif sur les devel xx

- Mise au point : écriture du code source, compilation, tests immédiats sur des petits jeux de données et sur un petit nombre de cœurs
- Accès immédiat aux noeuds
- Débogage avec GDB, DDT
- Noeuds non exclusifs
- Tous les utilisateurs se partagent l'ensemble des cœurs => un job peut être en mode Run ou en attente selon embouteillage si P > 8 processus tournent

ou

Mode Batch sur les clusters

- Plus de ressources
- En batch standard, lancer son job sans attendre la disponibilité des ressources devant sa console. Au mieux le job est en mode Run, sinon en mode Queue jusqu'à la disponiblité des ressources
- On peut rendre les nœuds exclusifs
- Quand un job est en mode Run, l'utilisateur est assuré d 'avoir en permanence N*C cœurs réservés à lui tout seul jusqu'à la fin du job
- Débogage avec GDB, DDT en batch interactif
- Noeuds pas toujours disponibles immédiatement au moment de l'exécution du batch

Travailler en mode interactif avec SCREEN

- **SCREEN est un utilitaire permettant entre autre de détacher une session sur laquelle tourne un code de calcul, sans interrompre le code, et de s'y rattacher à tout moment depuis n'importe quel poste de travail distant**

- **Avantage d'utiliser SCREEN : on peut détacher et rattacher à n'importe quel moment sa session sans perdre le contrôle de ses calculs, en particulier les impressions écran et les messages d'erreur en cas d'arrêt brutal du code de calcul**

- **Toutes les sessions SCREEN sont créées sur mygale**

Travailler en mode interactif avec SCREEN

Exemple d'utilisation :

- **sur mygale, taper la commande**

  screen -S nom_session

  la session screen nom_session s'ouvre alors sur mygale

- **depuis cette session screen, on peut se connecter sur un nœud donné xx (ssh develxx ou qsub en mode batch interactif) et exécuter un code de calcul**

- **on peut détacher cette session screen via la commande**

  [CTRL a][d]

  et on se retrouve déconnecté du nœud xx

Travailler sur une machine interactive avec SCREEN

- on se retrouve à nouveau sur mygale hors session screen
- on peut se déconnecter de mygale via la commande exit, puis se déconnecter de son poste de travail alors que le code continue à tourner sur le nœud xx
- depuis n'importe quel poste distant, on peut se connecter à nouveau sur mygale via ssh, se rattracher à la session screen précédente via la commande

  screen -r nom_session

- on se retrouve sur le nœud xx exactement dans le même état qu'avant le détachement de la session screen, et on peut contrôler à nouveau l'état du job de calcul

Travailler en mode interactif avec SCREEN

- **Commandes utiles :**
  - **créer à partir de mygale autant de sessions screen via la commande**
    screen -S nom_sessionxxx
  - **lister sur mygale l'ensemble des sessions screen ouvertes via la commande**
    screen -ls
  - **depuis n'importe quelle machine de calcul on se détache d'une session screen via la commande**
    [CTRL a][d]
  - **depuis mygale, on se rattache à une session screen via la commande**
    sreen -r nom_sessionxxx
  - **pour supprimer la session screen, on tape dans cette même session screen et sur mygale la commande**
    exit

Les points de reprise dans un code de calcul

- **INDISPENSABLE : dans les impressions, en dehors des sorties standard "print", pensez à générer dans le code de calcul des fichiers de sortie avec des points d'arrêt à intervalles réguliers**

- **Cela présente les avantages suivants :**
  - **avoir le contrôle des calculs à intervalle de temps régulier via les fichiers de résultats, et en cas de nécessité arrêter le code à temps en cas de divergence des calculs**
  - **en cas d'arrêt des machines, reprendre le calcul à partir du dernier point d'arrêt et non pas de l'état initial**

Les modules

- https://plafrim.bordeaux.inria.fr/doku.php?id=utilisation:modules:modules#les_modules
- Un module est un fichier qui va permettre d'utiliser un compilateur, une bibliothèque de calcul ou un logiciel installé sur les nœuds de calcul
- La liste des modules est accessible sur toutes les machines de calcul
- Cette liste est homogène sur l'ensemble des machines de calcul à l'exception des nœuds develxx et bonoboxxx qui contiennent en plus les modules associés à matlab, maple, ou à des logiciels de post-traitement graphiques
- Les modules sont classés dans quatre partitions en fonction de leurs usages

Les modules

- En général, un nom de module est composé de trois champs : xxx/yyy/nnn
  - xxx est le type de module (compilateur, bibliothèque, …)
  - yyy est le nom du type du module
  - nnn est le numéro de version

  Par exemple compiler/gcc/4.3.2

- Pour un module donné, il existe plusieurs numéros de version dont une « stable » (c'est à dire déjà validée) et une « latest » (la dernière installée en cours de validation)

- Nous recommandons de travailler avec des versions stables

- Lorsque qu'un module est chargé et qu'on ne précise pas son numéro de version, par défaut c'est la version « latest » qui est chargée

Les modules

- **Pour obtenir la liste des modules avec leur numéro de version :**

  module av

- **Pour charger un module :**

  module add nom_module

  Cette commande positionne automatiquement un certain nombre de variables d'environnement nécessaires à l'utilisation de bibliothèques

- **Pour connaître la liste des modules qui sont chargés dans votre environnement :**

  module liste

- **Pour enlever tous les modules de votre environnement :**

  module purge

Les métamodules

- https://plafrim.bordeaux.inria.fr/doku.php?id=utilisation:modules:modules#le_metamodule
- Idée : plutôt que de charger plusieurs modules pour compiler et exécuter un code de calcul, certains sont regroupés dans un « pack » que l'on appelle métamodule
- On utilise les commandes av, add, liste, … comme pour les modules
- Tous les métamodules portent le nom

  meta/nom_du_metamodule
- Tous les modules contenus dans un métamodule sont avec la version stable
- Pour connaître la liste des modules contenus dans un métamodule :

  module liste meta/nom_du_metamodule

Les modules et métamodules

- En mode interactif (incluant le batch interactif) les charger en tapant directement les commandes dans la fenêtre terminal
- En mode batch standard les charger dans le fichier de batch
- Si c'est un module ou métamodule que l'on utilise fréquemment, il suffit de le charger une fois pour toute dans le fichier .bashrc
- Il peut y avoir des dépendances (pré-requis) dans le chargement d'un module : par exemple un module M2 ne peut être chargé avant que le module M1 ne le soit. Un message vous le signalera avec les pré-requis nécessaires à charger au préalable

Les modules spécifiques à un utilisateur

- https://plafrim.bordeaux.inria.fr/doku.php?id=utilisation:modules:modules#les_modules_utilisateurs_plafrim-dev
- **Chaque utilisateur de PlaFRIM a la possiblité d'installer son propre module dans une partition spéciale plafrim-dev et de le faire partager à l'ensemble des utilisateurs**
- **Pour avoir les droits d'écriture et de dépôt dans cette partition, envoyer un courriel à plafrim-support@inria.fr. Vous recevrez alors toutes les indications et recommandations pour installer des nouveaux modules sur la plateforme**
- **Un utilisateur qui installe un module dans plafrim-dev a la charge de sa mise à jour**

Exemples de modules

compiler/intel/nnn mpi/intel/nnn	Partition /opt/cluster/modulefiles Compilateur Intel + Bibliothèque MPI
tools/debug/ddt/nnn	Partition /opt/cluster/modulefiles Débogeur DDT avec interface graphique
petsc/nnn	Partition /opt/cluster/plafrim-dev/modulefiles Bibliothèque PETSC
maple/nnn matlab/nnn scilab/nnn paraview/nnn visit/nnn tecplot/nnn	Partition /opt/cluster/softs/modulefiles Uniquement disponible sur les machines develxx et bonoboxxx

nnn : numéro de version

Compilation

- Si le code nécessite un débogage, utiliser l'option -g de compilation.
- Une fois le processus de débogage terminé, enlever l'option -g car elle ralentit les temps d 'exécution. On utilise souvent l'option de base d'optimisation -O
- Les compilateurs GNU sont installés par défaut avec le système et n'ont pas besoin d'être chargés via des modules. On utilise respectivement pour le Fortran et C les compilateurs gfortran et gcc

Compilation

- **Pour les compilateurs intel pour un code séquentiel, on charge le module :**

  module add compiler/intel/11.1-075

  et on utilise respectivement pour le Fortran et C les compilateurs ifort et icc

- **Pour les compilateurs intel pour un code parallèle avec MPI, on charge les modules :**

  module add compiler/intel/11.1-075

  module add mpi/intel/4.0.0.028

  Ces deux modules sont regroupés dans le méta-module meta/intel-impi

  et on utilise respectivement pour le Fortran et C les compilateurs mpif90 et mpicc

Post-traitement des données

Comment exploiter graphiquement ses résultats de calcul ?

- **Directement sur les machines develxx et bonoboxxx ?**
  - **à déconseiller**
  - **car ces logiciels fonctionnent avec une interface graphique et ces machines ne possèdent pas des cartes graphiques puissantes => lenteur ou problème d'affichage depuis votre poste de travail distant**
  - **les logiciels graphiques quand il s'agit de faire du post-traitement sont installés sur ces machines uniquement comme palliatif à l'impossibilité d'un usage local sur votre poste de travail distant ou pour exploiter les moteurs de traitement sur des petits cas tests et non pas l'affichage graphique**

Post-traitement des données

- **Transférer et visualiser les résultats sur votre compte IMB ?**
  - **à déconseiller**
  - **car problème de taille de stockage des données**

- **Solution :**

  visualiser directement à partir de son poste de travail distant en accédant à ces données à distance sans les recopier sur son compte IMB, avec la procédure sshfs

  https://dihpes.bordeaux.inria.fr/doku.php?id=utilisation:faq:connexion#comment_acceder_a_ses_donnees_via_sshfs

Post-traitement des données

- **Sur votre compte IMB, créer un dossier temporaire :**

  mkdir /tmp/donneesxxx

  Ne pas créer ce dossier dans /home/imb/id_bdx1

- **Monter le répertoire qui contient vos données sur votre compte PlaFRIM, par exemple dans votre espace de travail /lustre/id_plafrim.**
  Sous /tmp taper la commande :

  sshfs -o uid=`id -u` id_plafrim@mygale:/lustre/id_plafrim/donneesxxx /tmp/donneesxxx

- **Sur votre compte IMB dans le répertoire /tmp/donneesxxx toutes vos données /lustre/id_plafrim/donneexx sont montées et non dupliquées**

- **A la fin de votre travail, démonter le dossier :**

  fusermount -u /tmp/donneesxxx

Contacts PlaFRIM

- **Pour toute question technique relative au compte, aux données, à un problème système, à une demande dans plafrim-dev :**

  plafrim-support@inria.fr

- **Pour toute question technique relative à un logiciel ou l'utilisation d'une bibliothèque :**

  plafrim-users@inria.fr

  Cette mailing liste contient la liste de tous les utilisateurs de PlaFRIM

Mésocentre de Calcul Intensif Aquitain (MCIA)

- http://www.mcia.univ-bordeaux.fr (site de présentation générale)
- Direction Informatique, Université Bordeaux 1
- Objectif : mettre à disposition des laboratoires de recherche et des entreprises d'Aquitaine un plateau technique de qualité et un lieu d'échange d'expériences et de compétences dans le domaine du calcul intensif

Mésocentre de Calcul Intensif Aquitain (MCIA)

- **Condition nécessaire pour avoir un compte MCIA : avoir un compte Bordeaux 1**

- **Demande de création de compte :**

  http://www.mcia.univ-bordeaux.fr/index.php?id=inscriptions

- **Connection sur le cluster *avakas* avec l'identifiant et le mot de passe Bordeaux 1 à partir de son compte IMB ou à partir de son compte PlaFRIM sur mygale :**

  ssh id_bdx1@avakas.mcia.univ-bordeaux.fr

Mésocentre de Calcul Intensif Aquitain (MCIA)

- **http://redmine.mcia.univ-bordeaux.fr (site de description des ressources)**
  - **S'autentifier avec CAS avec son identifiant et mot de passe**
  - **Cluster de calcul *avakas* : http://redmine.mcia.univ-bordeaux.fr/projects/cluster-avakas/wiki**
  - **Soumission en batch et chargement de modules**
  - **Poster un ticket en cas de problème http://redmine.mcia.univ-bordeaux.fr/projects/cluster-avakas/issues/new**

Centre Informatique National de l'Enseignement Supérieur (CINES)

- http://www.cines.fr (site de présentation générale)
- Etablissement public national, basé à Montpellier et placé sous la tutelle du Ministère de l'Enseignement Supérieur et de la Recherche
- Condition nécessaire pour avoir un compte CINES : avoir un compte IMB

Centre Informatique National de l'Enseignement Supérieur (CINES)

- **Demande de création de compte :**

  http://www.cines.fr/spip.php?rubrique283

- **Cluster de calcul *jade***

  http://www.cines.fr/spip.php?rubrique291

- **Connections :**

  - **à partir de son compte IMB sur la machine *fermat* de l'IMB**

    ssh fermat

  - **puis sur le cluster de calcul *jade* du CINES :**

    ssh id_cines@jade.cines.fr

PROGRAMMATION PARALLÈLE

Qu'est ce que la parallélisation ?

- C'est un ensemble de techniques logicielles et matérielles permettant l'exécution simultanée de séquences d'instructions indépendantes sur plusieurs cœurs de calcul

PROGRAMMATION PARALLÈLE

Pourquoi paralléliser ?

- La première question à se poser, c'est savoir si la parallélisation de l'application est nécessaire
- Ecrire un programme séquentiel est dejà du travail, souvent difficile; la parallélisation le rendra encore plus dur
- Il existe toujours des applications scientifiques qui consomment "trop" de ressources en temps de processeur ou en mémoire

PROGRAMMATION PARALLÈLE

Pourquoi paralléliser ?

- **Les temps de calcul en séquentiel sont extrêmement élevés :**
  - **Il y a urgence à obtenir des résultats dans des délais raisonnables**
  - **les centres de ressources de calcul ne permettent pas de monopoliser les nœuds pendant des semaines, voire des mois**
- **La taille des données du problème à résoudre devient très élevée et la mémoire des nœuds ne permet plus de faire tourner le code sur un seul cœur de calcul (donc en mode séquentiel) même en exploitant toute la mémoire vive du nœud de calcul**
- **la seule solution, pour des raisons techniques ou économiques, reste la parallélisation**

PROGRAMMATION PARALLÈLE

Bénéfice de la parallélisation

- **Exécution plus rapide du programme (gain en temps de restitution) en distribuant le travail sur différents cœurs de calcul**

- **Résolution de problèmes avec un nombre de degré de libertés très élevé (plus de ressources matérielles accessibles, notamment la mémoire)**

PROGRAMMATION PARALLÈLE

Remarques importantes

- **Bien optimiser le code séquentiel avant de se lancer dans la parallélisation**

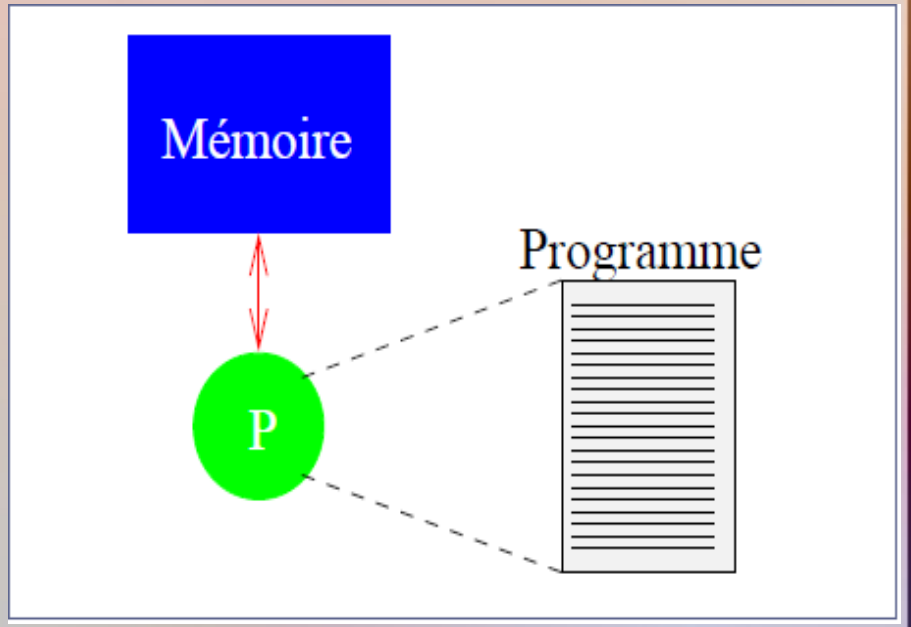
- **Paralléliser un code séquentiel ne consiste pas à reécrire le code de calcul :**
  - **ne pas dénaturer les algorithmes de calcul**
  - **garder les instructions de calcul du code séquentiel**
  - **modifier les boucles de calcul qui parcourent par exemple des données locales de tableaux**
  - **de façon judicieuse placer les instructions de parallélisme au bon endroit pour que lorsqu'elles sont omises, on retrouve les instructions de calcul du code séquentiel**

- **Bien s'assurer que le code parallèle sur C > 1 cœurs de calcul reproduise des résultats « identiques » ou le plus proches possible que ceux obtenus avec C = 1**

PROGRAMMATION PARALLÈLE

Programmation séquentielle

- Le programme est écrit dans un langage classique (Fortran, C, C++, …)
- Le programme est exécuté par un et un seul processus
- Toutes les variables du programme sont allouées dans la mémoire allouée au processus
- Un processus s'exécute sur un cœur physique de la machine



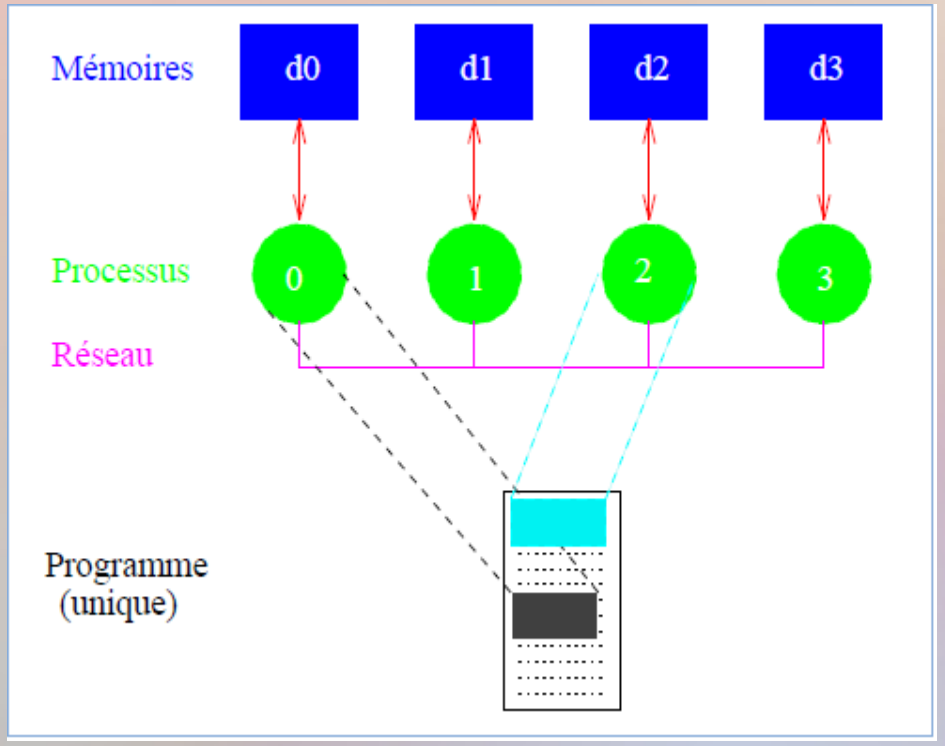
Programmation parallèle

- Noeuds de calcul à mémoire distribuée
  - le système est programmé en utilisant des échanges de messages (communications) entre les cœurs de calcul
  - protocole réseau
  - utilisation de bibliothèques MPI (Message Passing Interface)
- Noeuds de calcul à mémoire partagée
  - nœuds SMP (Symmetric Multi-Processor)
  - le mouvement des données est transparent pour l'utilisateur
  - utilisation de la bibliothèque OpenMP (Open Multi-Processing)

PROGRAMMATION PARALLÈLE

Programmation parallèle avec échanges de messages

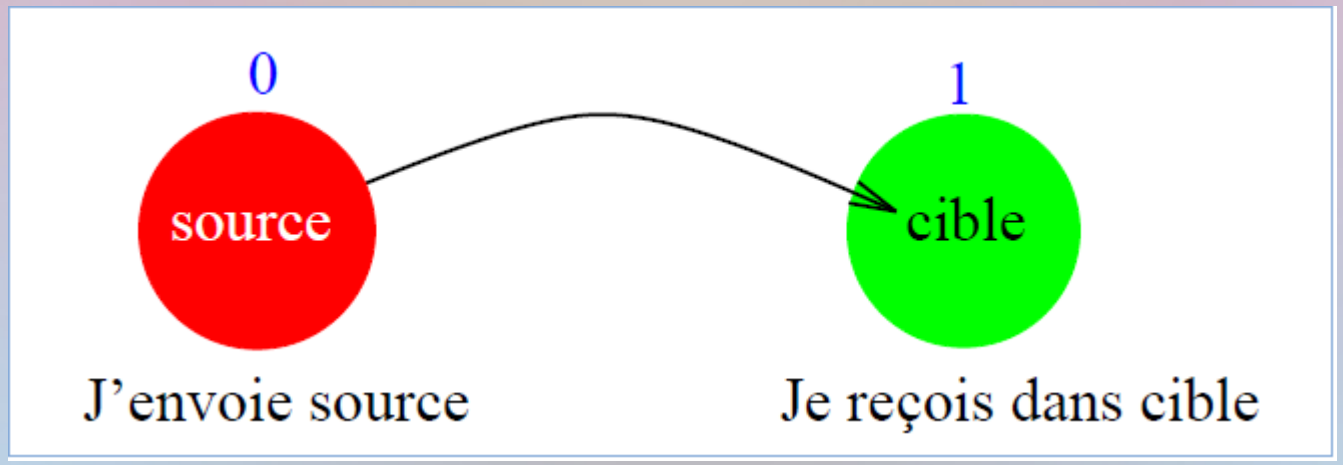
- Programme écrit dans un langage classique (Fortran, C, C++, ...)
- Bonne règle : exécuter un seul processus sur un cœur de la machine
- Variables allouées dans la mémoire locale allouée au processus
- Le même programme est exécuté par tous les processus selon le modèle SMPD (Single Program Multiple Data)
- Des données sont échangées entre plusieurs processus via des appels utilisant la bibliothèque MPI



PROGRAMMATION PARALLÈLE

Programmation parallèle avec échanges de messages

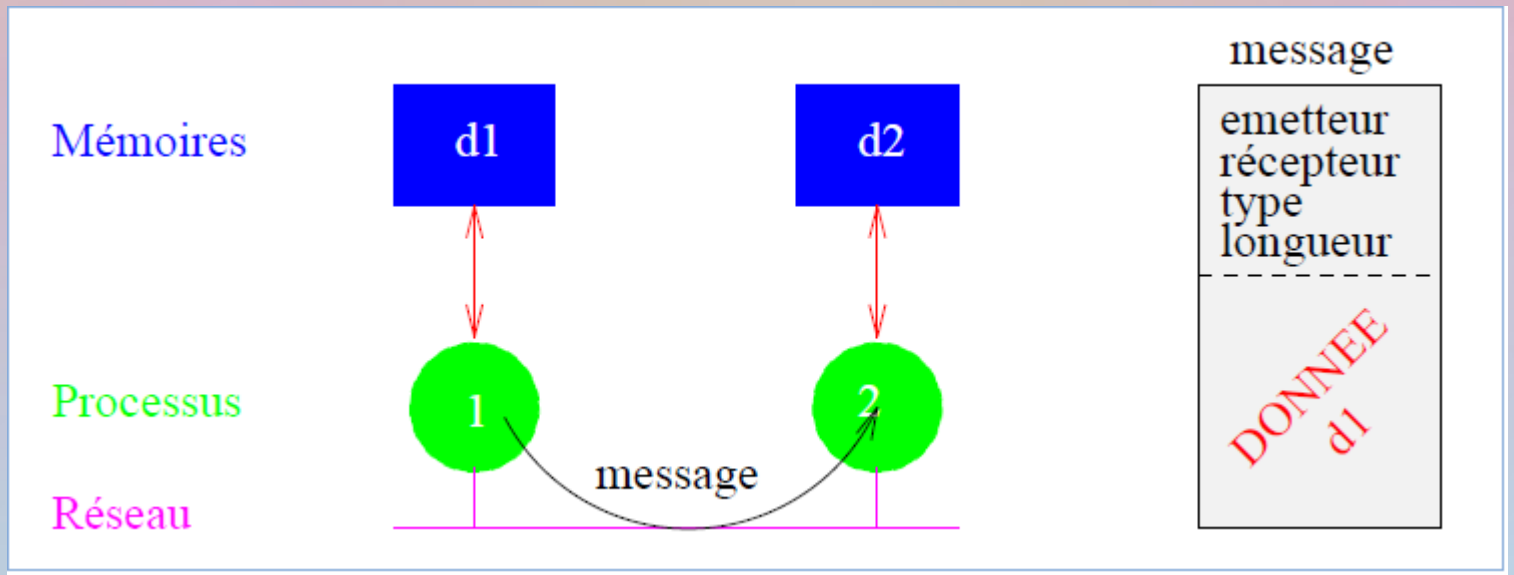
- Un processus est défini par :
  - un ensemble d'instructions à exécuter un programme
  - un espace mémoire pour les données de travail
- Si un message est envoyé à un processus, celui-ci doit ensuite le recevoir
- Un message est constitué de paquets de données transitant du processus émetteur au(x) processus récepteur(s)



PROGRAMMATION PARALLÈLE

Programmation parallèle avec échanges de messages

- **En plus des données (variables scalaires, tableaux, etc.) à transmettre, un message doit contenir les informations suivantes :**
  - **l'identificateur du processus émetteur**
  - **le type de la donnée**
  - **sa longueur**
  - **l'identificateur du processus récepteur**



PROGRAMMATION PARALLÈLE

Programmation parallèle avec échanges de messages

- Les messages échangés sont interprétés et gérés par un environnement qui peut être comparé à la téléphonie, à la télécopie, au courrier postal, à la messagerie électronique, ...
- Le message est envoyé à une adresse déterminée
- Le processus récepteur doit pouvoir classer et interpréter les messages qui lui ont été adressés

PROGRAMMATION PARALLÈLE

Programmation parallèle avec échanges de messages

- **L'environnement en question est MPI (Message Passing Interface). Une application MPI est un ensemble de processus autonomes exécutant chacun leur propre code et communiquant via des appels à des sous-programmes de la bibliothèque MPI**

- **Une parallélisation efficace doit minimiser les communications par rapport aux calculs**

- **Supports de cours :**

  http://www.idris.fr/data/cours/parallel/mpi/IDRIS_MPI_cours_couleurs_presentation.pdf

  http://www.math.sciences.univ-nantes.fr/~moebs/intr_mpi.pdf

PROGRAMMATION PARALLÈLE

Accélération et efficacité

- Les deux sont une mesure de la qualité de la parallélisation
- Soit T(C) le temps d'exécution sur C cœurs
- L'accéleration A(C) et l'efficacité E(C) sont définies comme étant :
  - A(C) = T(1) / T(C) (C = 1, 2, 3, ...)
  - E(C) = A(C) / C
- Pour une accélération parallèle parfaite on obtient :
  - T(C) = T(1) / C
  - A(C) = T(1) / T(C) = T(1) / (T(1) / C) = C
  - E(C) = A(C) / C = C / C = 100%

PROGRAMMATION PARALLÈLE

Scalabilité

- **Propriété d'une application a être exécutée efficacement sur un très grand nombre de cœurs de calcul**
- **Raisons possibles d'une faible scalabilité :**
  - **charge de la machine**
  - **analyser le comportement du code parallèle, améliorer les performances**
  - **un peu des deux …**
- **La loi d'Amdahl :**
  - **plus une règle qu'une loi**
  - **principe : la partie non parallèle d'un programme limite les performances et fixe une borne supérieure a la scalabilité**
- **Les programmes scalables demeurent efficaces pour un grand nombre de cœurs (scalables = passage à l'échelle)**

PROGRAMMATION PARALLÈLE

Scalabilité

- **Prenons un exemple avec une parallélisation de l'ordre de 80% du code :**

  T(1) = (T(partie parallèle) = 80)

  + T(partie non parallèle) = 20)

- **Sur C coeurs, on obtient**

  T(C) = (80 / C) + 20

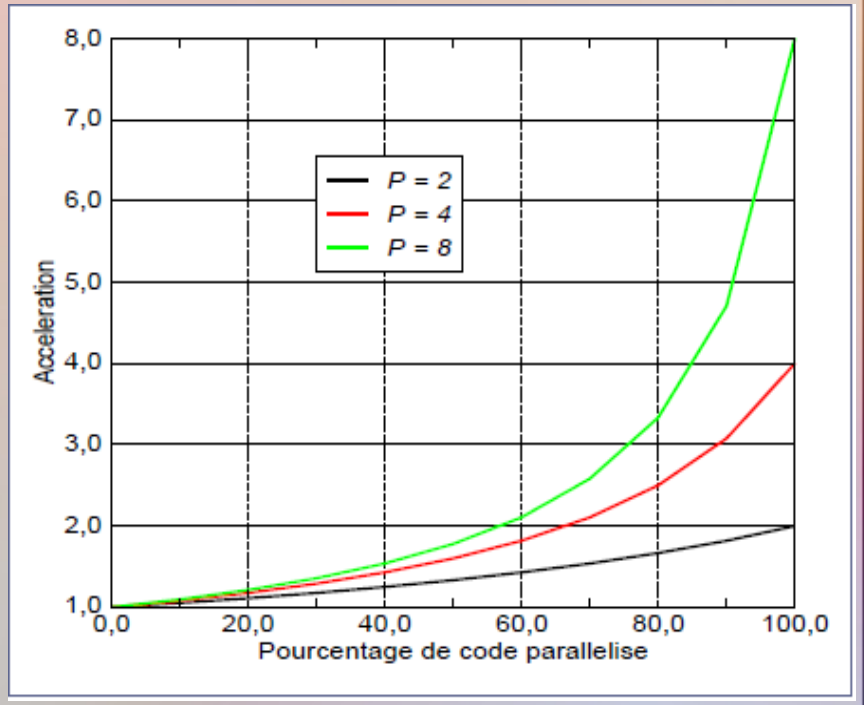
- **Quel que soit C, T(C) > 20**

- **A(C) = T(1) / T(C)**

  = 100 / ((80 / C) + 20) < 100/20 = 5

- **E(C) = (A(C) / C) < (5 / C) → 0 !!!**

  quand C → +∞



Dans le schéma, p = C

PROGRAMMATION PARALLÈLE

Exemple : anneau de communication (Fortran)

```
program anneau
     use mpi
     implicit none
     integer, dimension(MPI_STATUS_SIZE) :: statut
     integer, parameter                  :: etiquette=300
     integer                             :: nb_processus, rang,num_proc_precedent, num_proc_suivant, code
     integer                             :: i
     real, dimension (1:2)               :: T, V
     call MPI_INIT (code)
     call MPI_COMM_SIZE (MPI_COMM_WORLD, nb_processus, code)
     call MPI_COMM_RANK (MPI_COMM_WORLD, rang, code)
     num_proc_precedent = mod(nb_processus+rang-1, nb_processus)
     num_proc_suivant   = mod(rang+1, nb_processus)
     do i = 1, 2
      T(i)=0.1*rang*i
     end do
     if (rang == 0) then
       call MPI_SEND (T, 2, MPI_REAL, num_proc_suivant, etiquette, MPI_COMM_WORLD, code)
       call MPI_RECV (V, 2, MPI_REAL, num_proc_precedent, etiquette, MPI_COMM_WORLD, statut, code)
     else
       call MPI_RECV (V, 2, MPI_REAL, num_proc_precedent, etiquette, MPI_COMM_WORLD, statut, code)
       call MPI_SEND (T, 2, MPI_REAL, num_proc_suivant, etiquette, MPI_COMM_WORLD, code)
     end if
     print *,'Moi=',rang,', j''ai recu le tableau de valeurs ',V(1),' et ',V(2),' du coeur=',num_proc_precedent
     call MPI_FINALIZE(code)
end program anneau
```

PROGRAMMATION PARALLÈLE

Exemple : anneau de communication (C)

```
#include "mpi.h"
int main (int argc,char *argv[])
{
     int              nb_processus, rang, num_proc_precedent, num_proc_suivant ;
     int               i ;
     const int        etiquette=300 ;
     MPI_Status   statut ;
     float             T[2], V[2];
     MPI_Init (&argc, &argv) ;
     MPI_Comm_rank (MPI_COMM_WORLD, &rang) ;
     MPI_Comm_size (MPI_COMM_WORLD, &nb_processus) ;
     num_proc_precedent = (nb_processus+rang-1) % nb_processus ;
     num_proc_suivant = (rang+1) % nb_processus ;
     for (i = 0; i <= 1; i++)
      T[i]=0.1*rang*(i+1) ;
     if (rang == 0)
       {
       MPI_Send (T, 2, MPI_FLOAT, num_proc_suivant, etiquette, MPI_COMM_WORLD) ;
       MPI_Recv (V, 2, MPI_FLOAT, num_proc_precedent, etiquette, MPI_COMM_WORLD, &statut) ;
       }
     else
       {
       MPI_Recv (V, 2, MPI_FLOAT, num_proc_precedent, etiquette, MPI_COMM_WORLD, &statut) ;
       MPI_Send (T, 2, MPI_FLOAT, num_proc_suivant, etiquette, MPI_COMM_WORLD) ;
       }
       printf ("Moi=%d, j'ai recu le tableau de valeurs %f %f du coeur=%d\n",
             rang, V[0], V[1], num_proc_precedent) ;

     MPI_Finalize() ;
}
```

PROGRAMMATION PARALLÈLE

Exemple : anneau de communication

- **Chargement du module :** module add meta/intel-impi
- **Compilation :** mpif90 -O -o anneauF anneau.f90

  mpicc -O -o anneauC anneau.c
- **Exécution :** mpirun -np 8 ./anneauF

  mpirun -np 8 ./anneauC

Moi=1, j'ai recu le tableau de valeurs 0.000000 0.000000 du coeur=0

Moi=2, j'ai recu le tableau de valeurs 0.100000 0.200000 du coeur=1

Moi=3, j'ai recu le tableau de valeurs 0.200000 0.400000 du coeur=2

Moi=4, j'ai recu le tableau de valeurs 0.300000 0.600000 du coeur=3

Moi=0, j'ai recu le tableau de valeurs 0.700000 1.400000 du coeur=7

Moi=6, j'ai recu le tableau de valeurs 0.500000 1.000000 du coeur=5

Moi=7, j'ai recu le tableau de valeurs 0.600000 1.200000 du coeur=6

Moi=5, j'ai recu le tableau de valeurs 0.400000 0.800000 du coeur=4

PROGRAMMATION PARALLÈLE

Résolution numérique d'un problème sur un maillage global avec échanges de messages MPI

- On découpe le domaine global de résolution en SD sous-domaines de tailles le plus homogène possible
- On souhaite faire tourner le code de calcul sur SD cœurs
- On définit une topologie de numérotation des sous-domaines et ceux-ci sont numérotées de 0 à SD-1
- On affecte de façon bijective un processus à un sous-domaine et à un cœur de calcul
- Les tableaux sont alloués localement à la taille des sous-domaines
- Les processus sont identifiés par les coordonnées des sous-domaines
- Les SD processus exécutent tous le même programme en parallèle et s'échangent des données aux interfaces des sous-domaines

SUBVERSION

A quoi sert SVN ?

- SVN : logiciel de gestion de versions d'un projet scientifique (code de calcul)
- http://www/imb/cellule/spip.php?article116

  http://www/imb/cellule/spip.php?article118
- Gérer des versions d'un même code de calcul et partager ces versions avec d'autres personnes
- Utile quand plusieurs personnes participent au développement d'un même code de calcul
- Garder la trace des modifications faites lors du développement d'un code, et pouvoir récupérer à tout moment une version (révision) numéro n de l'ensemble du code ou de quelques fichiers de ce code

SUBVERSION

Serveur SVN

- Tous les projets et les révisions d'un projet donné sont déposés, mis à jour, centralisés et gérés sur le serveur SVN svn.math.u-bordeaux1.fr
- On appelle révision une version numérotée du projet
- Ce serveur est accessible depuis votre compte IMB ou depuis mygale sur votre compte PlaFRIM

SUBVERSION

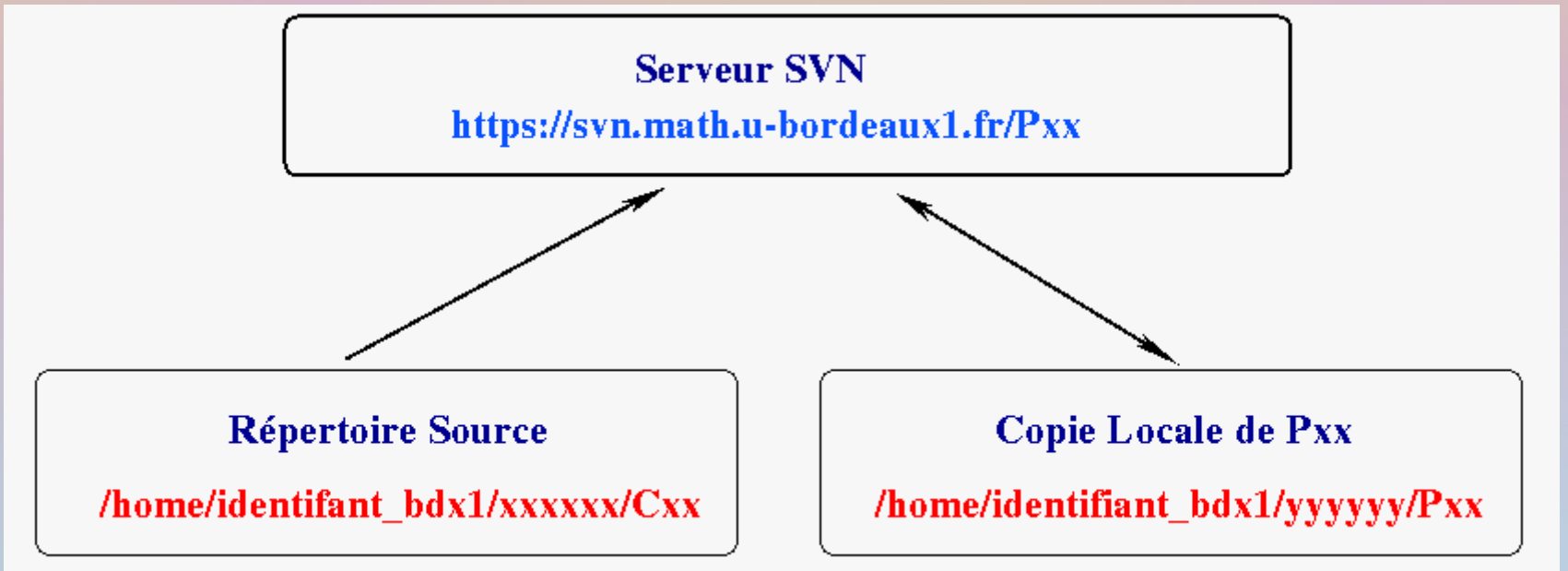
Demande de création d'un projet sur le serveur SVN

- Un porteur de projet que l'on nomme administrateur de projet :
  - demande auprès de la cellule informatique la création d'un projet qu'il souhaite nommer Pxx sur le serveur SVN
  - fournit la liste des participants à ce projet
- L'URL https://svn.math.u-bordeaux1.fr/Pxx :
  - est alors accessible pour l'ensemble des membres de ce projet
  - et leur permet de suivre l'historique de toutes les révisions

SUBVERSION

Principe de fonctionnement

- Soit Cxx le répertoire source d'un code de calcul
- ETAPE 1 : déposer Cxx dans https://svn.math.u-bordeaux1.fr/Pxx
- ETAPE 2 : faire une copie locale *« labellisée SVN »* de Pxx qui contient Cxx
- Tous les transferts de versions de Pxx se feront entre la copie locale et le serveur SVN via des commandes SVN qui commencent toujours par svn



SUBVERSION

Serveur SVN et copie locale

- **Sur le serveur SVN :**
  - **on initalise le projet en y créant un dépôt initial qui contient la première révision de Pxx**
  - **on centralise toutes les révisions de Pxx**
- **La copie locale :**
  - **est créée (donc initialisée) en y recopiant la première révision de Pxx du serveur SVN**
  - **ne contient que la révision courante sur laquelle on travaille**
- **Toute action de propagation de révision de la copie locale vers le serveur SVN ne se fait que via la commande svn ci (commit)**

SUBVERSION

Préalables

- Positionner la variable d'environnement EDITOR au nom de votre éditeur de texte préféré ; par exemple en ajoutant export EDITOR=emacs dans votre fichier .bashrc
- Ce sera via cet éditeur de texte que l'utilisateur pourra commenter toutes ses actions de travail avec SVN
- Lors de la première utilisation de SVN, il vous sera demandé d'accepter le certificat du serveur svn.math.u-bordeaux1.fr, de façon temporaire ou permanente

SUBVERSION

Déposer son code de calcul sur le serveur
Création du dépôt initial sur le serveur SVN

- **Se positionner dans le répertoire qui contient le répertoire Cxx, par exemple /home/id_plafrim/xxxxx**

- **Déposer le contenu du répertoire Cxx sur le serveur SVN :**

  svn import Cxx https://svn.math.u-bordeaux1.fr/Pxx/Cxx

- **Ainsi la révision n° 1 se crée dans l'URL https://svn.math.u-bordeaux1.fr/Pxx qui contient le code de calcul Cxx**

SUBVERSION

Créer la copie locale du dépôt initial

- **Se positionner dans le répertoire /home/id_plafrim/yyyyy où l'on souhaite créer la copie locale**

- **Taper la commande :**

  svn co https://svn.math.u-bordeaux1.fr/Pxx (checkout)

- **Il se crée alors dans le répertoire /home/id_plafrim/yyyyy la copie conforme du dépôt initial qui se trouve sur le serveur, c'est à dire le répertoire Pxx avec tout son contenu dont bien entendu Cxx**

SUBVERSION

Pourquoi une copie locale du dépôt initial ?

- **Question naturelle :**
  quelle est la différence à l'initialisation entre
  /home/id_pladrim/xxxxx/Cxx et /home/id_plafrim/yyyyy/Pxx/Cxx ?

- **Aucune sauf que tout le répertoire /home/id_plafrim/yyyyy/Pxx (avec ses contenus dont Cxx) à l'initialisation est « labellisé » SVN**

- **Dorénavant, on travaille dans la copie locale /home/id_plafrim/yyyyy/Pxx en correspondance avec l'URL https://svn.math.u-bordeaux1.fr/Pxx**

- **Tous les changements de révisions de Pxx (modifications, rajouts, suppressions, mises à jour de sous-répertoires ou de fichiers) vont pouvoir s'opérer via des commandes SVN entre la copie locale et le serveur SVN**

- **Toutes les commandes SVN devront être exécutées dans la copie locale**

SUBVERSION

Modifier des fichiers dans la copie locale et propager la nouvelle révision vers le serveur SVN

- **Des modifications ont été apportés à certains fichiers de /home/id_plafrim/yyyyy/Pxx/Cxx**

- **Comment propager cette nouvelle révision de la copie locale vers le serveur SVN ?**

  svn ci (commit)

SUBVERSION

Ajouter des fichiers ou répertoires dans la copie locale et propager la nouvelle révision vers le serveur SVN

- Supposons que l'on crée un nouveau répertoire de fichiers CCxx sous /home/id_plafrim/yyyyy/Pxx

- Comment propager cette nouvelle révision de la copie locale vers le serveur SVN ?

  Sous /home/id_plafrim/yyyyy/Pxx taper respectivement les commandes :

  ```
  svn add CCxx
  svn ci
  ```

SUBVERSION

Avoir un aperçu de l'ensemble des révisions

- **svn log**
- **On trace ainsi l'ensemble des révisions accompagnées des commentaires sur toutes les actions qui ont été menées**

SUBVERSION

Voir les différences entre deux révisions

- Soient np et nq les numéros respectifs de deux révisions Rp de Rq de Pxx
- Comment voir la différence globale sur tout Pxx des révisions Rp et Rq ?

  svn diff -r np:nq

- Comment voir la différence sur un répertoire zzzzz de Cxx des révisions Rp et Rq ?

  svn diff -r np:nq Cxx/zzzzz

- Comment voir la différence sur un fichier fffff du répertoire zzzzz de Cxx des révisions Rp et Rq ?

  svn diff -r np:nq Cxx/zzzzz/fffff

SUBVERSION

Synchroniser la copie locale avec la dernière révision du serveur SVN

- **Mettre à jour**

  /home/id_plafrim/yyyyy/Pxx avec https://svn.math.u-bordeaux1.fr/Pxx

  svn up (update)

SUBVERSION

Récupérer une révision antérieure d 'un répertoire ou d'un fichier du serveur SVN vers sa copie locale

- Soir np le numéro d'une révision Rp sur le serveur SVN
- Comment mettre à jour la globalité de la copie locale avec la globalité de la révision np du serveur SVN ?

  svn up -r np
- Comment mettre à jour un répertoire zzzzz de Cxx de la révision Rp dans sa copie locale ?

  svn up -r np Cxx/zzzzz
- Comment mettre à jour un fichier fffff du répertoire zzzzz de Cxx de la révision Rp dans sa copie locale ?

  svn up -r np Cxx/zzzzz/fffff

Institut de Mathématiques de Bordeaux

Les ressources de calcul et leur utilisation

Supports de formation (transparents et TP) :

http://www.math.u-bordeaux1.fr/imb/cellule/spip.php?article81

23 et 24 octobre 2012
Khodor KHADRA, Ingénieur de Recherche Calcul Scientifique