



Gestion des Entrées/Sorties Parallèles dans les grappes SMPs.

Adrien LEBRE

Directeur de thèse : Brigitte PLATEAU

Co-encadrant ID : Yves DENNEULIN

Co-encadrant BULL : Pascale ROSSE-LAURENT

Laboratoire ID-IMAG (UMR 5132), Grenoble, France.

BULL - HPC, Échirolles, France.



Institut National
Polytechnique
de Grenoble



CENTRE NATIONAL
DE LA RECHERCHE
SCIENTIFIQUE



INSTITUT NATIONAL
DE RECHERCHE EN
INFORMATIQUE ET
EN AUTOMATIQUE



GRENOBLE
UNIVERSITÉ
JOSEPH FOURIER
SCIENTIFICO-TECHNOLOGIQUES





Plan

- 1 Introduction
 - Contexte
 - Entrées/Sorties parallèles
- 2 Le système aOLi
 - Préambule
 - Gestion “classique” des I/O
 - Principes et fondements
 - Notes techniques
- 3 Résultats préliminaires
 - POSIX vs aOLi
 - MPI I/O vs aOLi
- 4 Conclusion





Contexte

Environnement

- Grappe de SMPs
- Linux
- High Performance Computing
- Applications I/O intensives \Rightarrow Goulet d'étranglement

Entrées/Sorties parallèles

- Gestion des accès concurrents à un même fichier
- Accès : différence en taille, position. un exemple : la multiplication de matrices





Contexte

Environnement

- Grappe de SMPs
- Linux
- High Performance Computing
- Applications I/O intensives \Rightarrow Goulet d'étranglement

Entrées/Sorties parallèles

- Gestion des accès concurrents à un même fichier
- Accès : différence en taille, position. un exemple : la multiplication de matrices

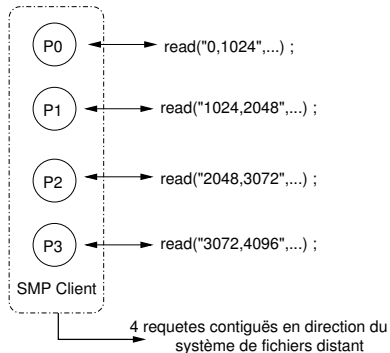




Entrées/Sorties parallèles (1)

Produit de matrices

- Parties spécifiques à récupérer (selon le découpage : colonne, ligne, BLOCK/BLOCK, BLOCK/CYCLYC ...)
- Plusieurs requêtes disjointes/contiguës au même instant !
- Type de comportement "fatal" pour les performances I/O

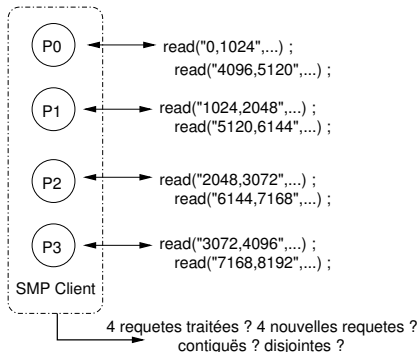




Entrées/Sorties parallèles (1)

Produit de matrices

- Parties spécifiques à récupérer (selon le découpage : colonne, ligne, BLOCK/BLOCK, BLOCK/CYCLYC ...)
- Plusieurs requêtes disjointes/contiguës au même instant !
- Type de comportement "fatal" pour les performances I/O

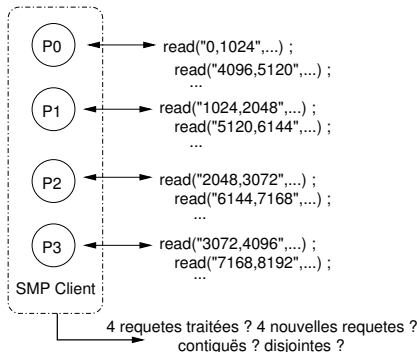




Entrées/Sorties parallèles (1)

Produit de matrices

- Parties spécifiques à récupérer (selon le découpage : colonne, ligne, BLOCK/BLOCK, BLOCK/CYCLYC ...)
- Plusieurs requêtes disjointes/contiguës au même instant !
- Type de comportement "fatal" pour les performances I/O





Entrées/Sorties parallèles (2)

Besoins / difficultés

- Vecteur d'accès (readv) \Rightarrow API plus complexe
- opération collective \Rightarrow synchronisation
- Vues physique (des fichiers) \Rightarrow placement physiques (blocs disque)

Solutions logicielles

- Systèmes de fichiers parallèles : plus ou moins complets/efficaces
 - "génériques" : PVFS, NFSP, PFS, Lustre
 - spécifiques aux parallèles I/O : PIOUS, VESTA ...
- Bibliothèques : maximiser les aspects portabilité
 - Nombreuses : un standard MPI I/O accepté
- API plus sophistiqué \Rightarrow Dépendance / Coût de développement





Entrées/Sorties parallèles (2)

Besoins / difficultés

- Vecteur d'accès (readv) \Rightarrow API plus complexe
- opération collective \Rightarrow synchronisation
- Vues physique (des fichiers) \Rightarrow placement physiques (blocs disque)

Solutions logicielles

- Systèmes de fichiers parallèles : plus ou moins complets/efficaces
 - “génériques” : PVFS, NFSP, PFS, **Lustre**
 - spécifiques aux parallèles I/O : PIOUS, VESTA ...
- Librairies : maximiser les aspects portabilité
 - Nombreuses : un **standard MPI I/O** accepté
- **API plus sophistiqué \Rightarrow Dépendance / Coût de développement**





Le système aOLi

Objectifs

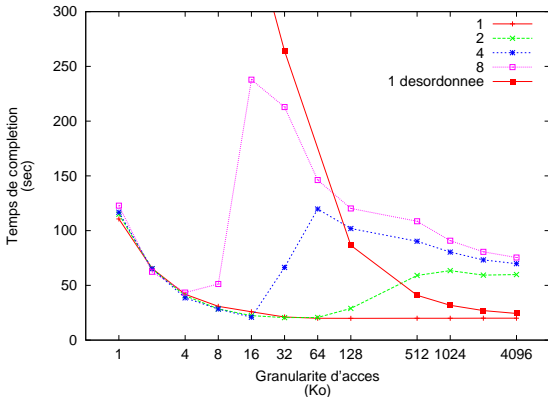
- Intégrer les algorithmes d'Entrées/Sorties parallèles
 - ré-ordonnancement
 - agrégation des accès ⇒ efficacité
 - recouvrement
- Exploiter uniquement les interfaces présents : POSIX
 - open/creat/lseek/read/write/close ⇒ simplicité
- Minimiser les surcoûts de mise en œuvre
 - éliminer au maximum les mécanismes de synchronisations





Évaluations de la pile I/O disque "standard"

Analyse d'une décomposition d'un fichier de 1 Go
(kernel 2.4.27, grappe IDPOT, NFS version 3)

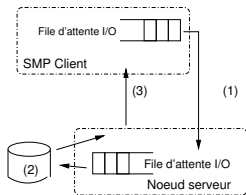




Les fondements

Définir une "fenêtre de tir"

- Maximiser l'utilisation des serveurs de stockages
 - Une requête au moins dans la queue coté serveur
- Appliquer les algorithmes d'Entrées/Sorties parallèles
 - Juste une requête dans la queue coté serveur !



- (1) Une première requête est transmise au système de fichiers.
- (2) Elle est exécutée sur le périphérique de stockage rattaché.
- (3) La réponse est renvoyée au client.

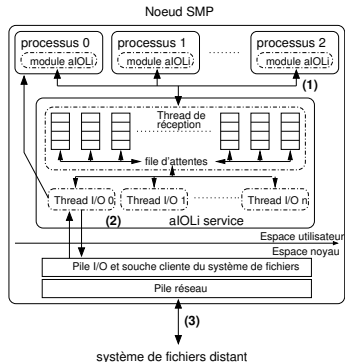




Conception d'un prototype

Librairie en espace utilisateur

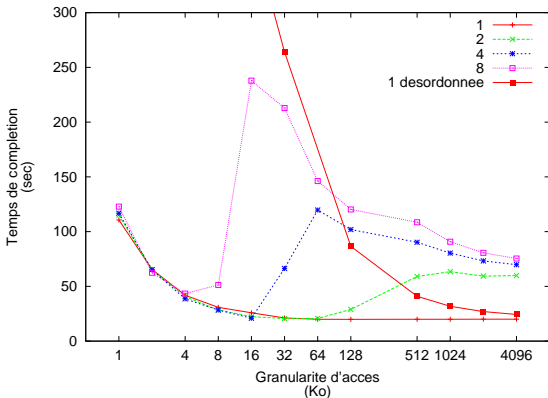
- Composant client surchargeant les appels POSIX (lié à l'application)
- Un service aIOLi
 - "Multi-threaded"
 - Réalise les différentes optimisations
 - Exécute les Entrées/Sorties effectives
- Mécanismes IPC et mémoire partagée





Évaluations : Décomposition aOLi

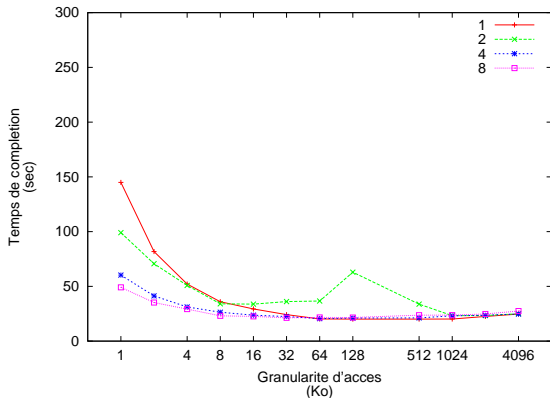
Analyse d'une décomposition d'un fichier de 1 Go
(kernel 2.4.27, grappe IDPOT, NFS version 3)





Évaluations : Décomposition aOLi

Analyse d'une **décomposition d'un fichier de 1 Go**
(kernel 2.4.27, grappe IDPOT, NFS version 3)

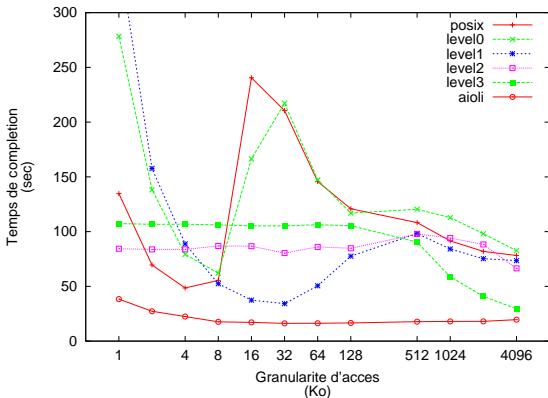




Évaluations : MPI I/O vs aIOli

décomposition d'un fichier de 1 Go

(kernel 2.4.27, grappe IDPOT, NFS version 3, mpich 1.2.5, ROMIO)





Conclusion

Points positifs

- Efficace, résultats encourageant
- Simplicité du modèle
- Pas synchronisation intra-processus
- API standard

Points négatifs

- Gestion du placement physique
- Surcoût sur accès standard
- Dépendant de l'ordonnanceur noyau
- ...

travaux futures

- Portage en module noyau
- Analyse sous SMP 16 voies et Lustre
- Ajout des considérations "data stripping"
- Étude de la coordination entre plusieurs SMPs
- ...





Question ???

<http://aioli.imag.fr>



Projet LIPS
BULL - INRIA - Laboratoire ID

