

Inria

Journée SED 2022

Algèbre linéaire :
Chameleon, Diodon, Compose

Sommaire

01. Introduction
02. Chameleon
03. Diodon
04. Compose

Inria Team **Concace, Topal, Storm, Tadaam**

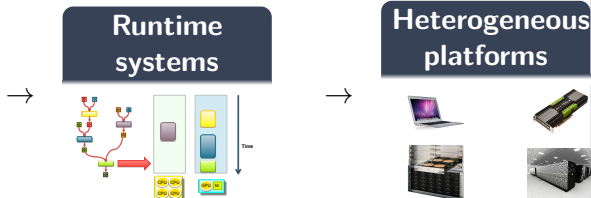
Linear algebra

$$AX = B$$

Task-Flow

```
for (j = 0; j < N; j++)
  Task (A[j]);
```

Direct Acyclic Graph

**Main goal** : performances, scaling

Chameleon: **dense matrix** solver

- BLAS: basic scalar, vector, matrix operations
- LAPACK: linear systems $\mathbf{AX} = \mathbf{B}$, least-squares, eigen val.

PaStiX/Qrmumps: **sparse matrix direct** solver

- linear systems $\mathbf{AX} = \mathbf{B}$, factorization \mathbf{LL}^T , \mathbf{LDL}^T , \mathbf{LU}

MaPHyS: **sparse matrix hybrid** solver

- linear systems $\mathbf{AX} = \mathbf{B}$, **CG/GMRES + Preconditioner**
- Preconditioner = direct solver \rightarrow MUMPS/PaStiX/Qrmumps

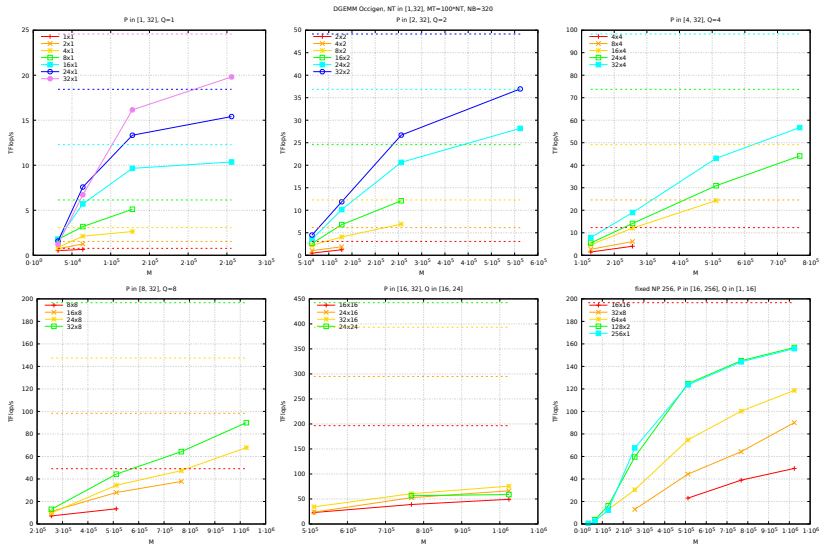
Paradigms: Threads, CUDA, MPI

Runtime systems: OpenMP, ParSEC, StarPU

02

Chameleon

Benchmarks - GEMM, QR



3 interfaces disponibles :

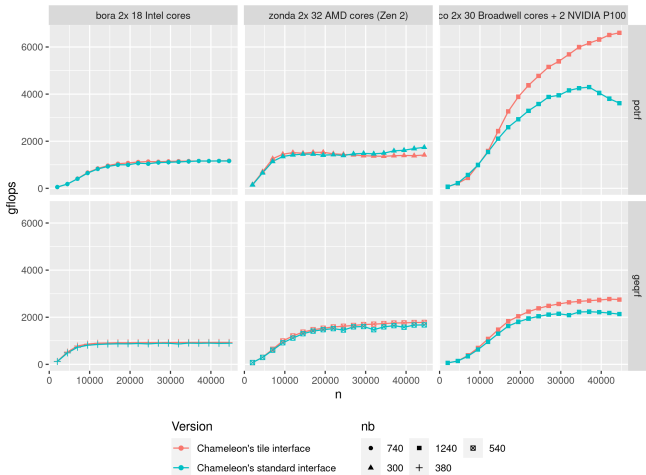
- Lapack : CHAMELEON_dgemm → non testée
- Chameleon : CHAMELEON_dgemm_Tile → testée
- Chameleon Async : CHAMELEON_dgemm_Tile_Async → testée

Validation et évaluation de l'interface Lapack de Chameleon



Couverture de tests 60% → 75%

Encadrement stage Alycia Lisito



Performances *dpotrf* et *dgeqrf* API Lapack

- l'interface Lapack s'exprime

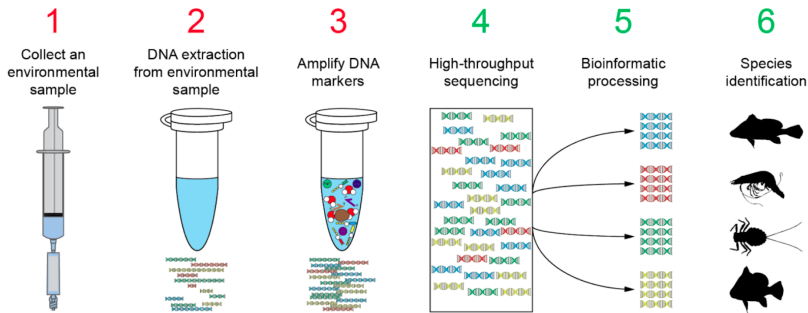
```
int CHAMELEON_dgemm( cham_trans_t transA, cham_trans_t transB, int M, int N, int K,  
                    double alpha, double *A, int LDA,  
                    double *B, int LDB,  
                    double beta, double *C, int LDC )
```

- les noms de fonctions et les types de paramètres ne sont pas strictement équivalents à l'interface Blas/Lapack

En projet : fournir une interface API compatible avec Lapack, puis si possible Scalapack

03

Diodon



- Généralement : $\approx 10\ 000$ individus
- Challenge : 1 000 000 individus \Rightarrow **calcul parallèle distribué**
- Données : Inrae, diatomées (algues) du lac Léman

Matrice de distance D

$$\begin{pmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,n} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ d_{n,1} & d_{n,2} & \cdots & d_{n,n} \end{pmatrix}$$

Projection sur un
espace Euclidien

$$r \ll n$$



Conservation des
distances

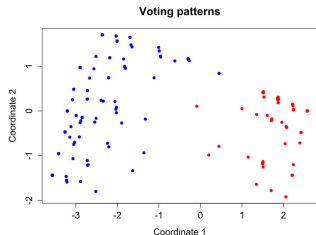
Nuage de points X

$$\begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,r} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,r} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,r} \end{pmatrix}$$

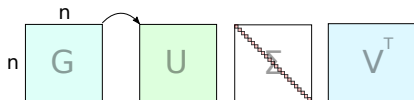
Visualisation de X en 2 ou 3 dimensions

$$\begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,r} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,r} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,r} \end{pmatrix}$$

X représenté sur un plan

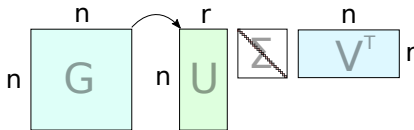


Calcul de SVD d'une matrice carré



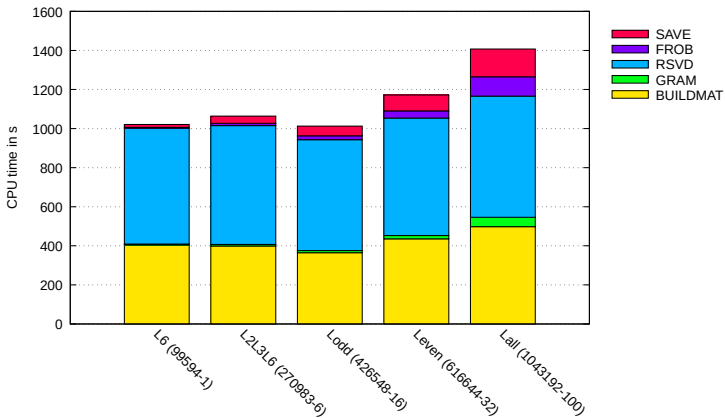
beaucoup trop complexe pour $n = 10^6$

Lorsque rang de G est $r \ll n$, la SVD peut être tronquée



- résolution par approche Randomized SVD
- GEMM, QR bien parallélisés avec Chameleon

Temps CPU de la MDS, $r = 10^4$, cas complexe à résoudre
 La RSVD passe bien à l'échelle !



CINES Occigen, $n \in [10^5, 10^6]$, #Nodes $\in [1, 100]$, #CPUs $\in [24, 2400]$

Réduction de dimension - RSVD parallèle :

Multi-Dimensional Scaling (MDS), Principal Component Analysis (PCA),
 Canonical Correlation Analysis (CCA), Correspondence Analysis (CoA).

Exemple PCA : matrice centrée réduite

$$\begin{pmatrix} \frac{A_{1,1}-\widetilde{A}_1}{\sigma(A_1)} & \cdots & \frac{A_{1,n}-\widetilde{A}_n}{\sigma(A_n)} \\ \vdots & \ddots & \vdots \\ \frac{A_{m,1}-\widetilde{A}_1}{\sigma(A_1)} & \cdots & \frac{A_{m,n}-\widetilde{A}_n}{\sigma(A_n)} \end{pmatrix}$$

$$(U, \Sigma, V) = SVD(A)$$

$$Y = U\Sigma$$

- EP : Pleiade, Concace, Topal, Storm, Tadaam
- bientôt diffusé en licence libre
- nouvelles collaborations ?

Diodon C++
MDS, PCA, CCA, CoA, MCoA, ...
FMR C++
Randomized SVD
Chameleon C
GEMM, QR, Gram
StarPU C
Task scheduling
OpenMPI/Nmad C
Communications

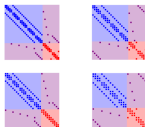
Pile logiciel Diodon

04

Compose

Logiciel Compose (ex. Maphys Fortran)

- code C++ moderne (17, 20) pour composer un large spectre de méthodes d'algèbre linéaires parallèles et **distribuées** :
 - > matrices denses, creuses
 - > opérations de bases : $+$ \times $<$, $>$ $||\cdot||$
 - > résolution méthodes directe et itérative : LU , [Bi]CG, GMRES
 - > décomposition de domaine algébrique



- soutien à l'intégration dans des logiciels d'équipes partenaires
 - > Memphis : mécanique des fluides, interaction fluides/solides
 - > Serena (Paris) : hydrogéologie, réseaux de fractures de roche
 - > autres ?