Bibliothèques pour le calcul numérique parallèle ; synthèses et perspectives

Serge PETITON

Laboratoire d'Informatique Fondamentale de Lille (LIFL),
URA CNRS 369,
Université des Sciences et Technologies de Lille (Lille I)

et

Laboratoire d'Applications Scientifiques pour le Calcul Intensif (ASCI),
UPR CNRS 9029.

`petiton@lifl.fr` ou `petiton@asci.fr`
`http:\\www.lifl.fr\~petiton`

Observatoire Français des Techniques Avancées, 24 Octobre 1996

Bibliothèques pour le calcul numérique parallèle ; synthèses et perspectives

PLAN

I Introduction

II Historique, de LINPACK à ScaLAPACK

III Des bibliothèques vers les *toolkits* pour le calcul scientifique parallèle

IV Du *out of core* et autres problèmes

V Conclusion

VI Perspectives et prévisions

Objectifs : faire le point sur l'existant, ses avantages et ses inconvénients ; présenter les progrès possibles et discuter des directions de recherche et de développement probables.

I - Introduction

Problèmes physiques	Modélisation	• Méthodes explicites
Electromagnétisme		• autres
CFD		• Méthodes implicites
Chimie		**Algèbre linéaire**
...		- $Ax = b$
		- $Au = \lambda u$
		- autres

Trouver la solution (avec une bonne précision) le plus rapidement possible (i.e. temps (en seconde) le plus petit)

Plusieurs pas de temps

$\Rightarrow$ Résoudre plusieurs fois un problème d'algèbre linéaire

De nombreuses bibliothèques d'algèbre linéaire existent, parallèles ou non.

Il existe aussi des bibliothèques parallèles plus spécialisées: mailleurs, PDE, CFD,... mais en bien plus petit nombre et moins portables en général.

Nous avons principalement regardé les bibliothèques d'algèbre linéaire.

II - Historique, de LINPACK à ScaLAPACK

A - Bibliothèques scalaires

LINear algebra PACKage **(LINPACK)** ; lancée en 1974.

Premiers BLAS (SAXPY, SDOT,...),

optimisés par les constructeurs ; et MACHEPS.

Principalement des méthodes directes.

Matrices denses ou bandes.

Dongarra, Bunch, Moler et Stewart (aide de Wilkinson).

EIgenvalues PACKage **(EISPACK)**

Calcul de vecteurs et valeurs propres de matrices denses.

Méthode QR comme base, pas de méthodes de Krylov.

Matrices denses ou bandes ; tailles moyennes ou petites

Garbow, Boyle, Dongarra, Moler (et al).

Linear Algebra PACKage **(LAPACK)**, ou **CLAPACK** et **LAPACK++**

LINPACK + EISPACK (approximativement)

Ajout de certaines méthodes des années 1970 et de *drivers*.

BLAS 1,2 (84-86) et 3 (87-88)

Beaucoup d'algorithmes *blocés*.

UTK et ONRL, Jack Dongarra et al.

ARnoldi PACKage **(ARPACK)**, récent.

basé sur la version implicite d'Arnoldi

Utilise BLAS 1,2 et 3 et LAPACK

Université de Rice, Danny Sorensen

ITerative methods PACKage **(ITPACK)**

Et IMSL, NAG, GAUSS, SMPAK (SCA) et bien d'autres...,

Autres Bibliothèques: CraySoft Libraries (seismic, chemistry, physics,..), Euler, NS, ...

<u>Cas des matrices creuses</u>

Yale Sparse Package, 1977

Bibliothèque pour matrices creuses.

Yale Sparse format (futur CSR) comme format de base.

Yale University, DCS, Eisenstat, Gursky, Schultz et Sherman.

ELLement finite PACKage **(ELLPACK)**

Stocke les matrices creuses dans un tableaux à 2 dimensions,

dont une compressée.

SPARSe KIT **(SPARSKIT)**

Bibliothèque pour matrices creuses.

format CSR comme format de base.

Outils de manipulation de matrices, de visualisation.

NASA puis Université de Minneapolis, Y. Saad.

La bibliothèque travaille uniquement avec le format CSR.

Il existe des primitives de conversion de CSR vers/de nombreux

autres formats (COO, CRC, ELLPACK, profil,....).

B - Bibliothèques vectorielles

Liste très partielle :

Version de **LINPACK** avec **BLAS** vectoriels :

> réécriture des **BLAS**, optimisés pour les machines vectorielles.

Version vectorielle de **LINPACK** :

> vectorisation explicite de certaines subroutine de **LINPACK**.

Bibliothèque **MAGEV** ; CDC, 1984.

> Utilise les primitives vectorielles Q8 du Fortran vectoriel de CDC, avec pagination optimisée ; pas de calcul de valeurs et vecteurs propres non symétriques.

Bibliothèque vectorielle **EIVPACK** pour le calcul de valeurs et vecteurs propres de matrices, y compris non Hermitiennes ; avec paginations optimisées.

> CDC France, 1987. Inclus déjà les méthodes d'Arnoldi.
> Versions vectorielles du QR (1984) sur CDC 205.

Calculs creux *vectoriels* à l'aide de Gather-Scatters, souvent avec une aide hardware (CRAY X-MP....).

C - Bibliothèques Parallèles

Première visite, (liste partielle)

version Parallèle de **LAPACK** : **PLAPACK**

Distribution des vecteurs, puis distribution induite des matrices.

Linear Algebra Object (LAObj).

Utilise les PBLAS.

University of texas at Austin, R. van de Geijn et al.

version Parallèle d'un sous-ensemble de **LAPACK** : **ScaLAPACK**

projet DARPA/ITO, objectif : réalisation d'une bibliothèque *scalable* pour l'algèbre linéaire dense ou creuse pour les machines massivement parallèles:

Avec **PBLAS**, **BLACS**, et **BLAS** ; PVM ou MPI, ou autres

Inclus **PARPAK** (D. Sorensen), utilise **BLACS** et MPI,

Inclus **PARPRE** (UCLA, T. Chan), utilise routine de **PETSc**,

Inclus **CAPSS** (Urbana, M. Heath), Méthodes directes, creux.

UTK et ONRL ; Dongarra et al.

CM Scientific Sotware Library : **CMSSL**

Bibliothèque data parallèle de TMC pour CM-(2,5)

Avec Arnoldi implicite.

Matrices creuses sous format COO-data-parallèle.

Avec compilation des communications possibles sur CM-2

Lennart Johnsson et al.

Portable, Extensible Toolkit for Scientific computation : **PETSc**

(version 2.0 annoncée le 11 octobre 1996)

Orienté objet, ciblé pour PDE.

Utilise **BLAS**, **LAPACK**, MPI

Possède aussi des *drivers*, *profiler* et outils de visualisation.

ANL, Bill Gropp et al.

TM Scientific Software Library : **TMSSL**

Adaptation de CMSSL, très incomplète pour l'instant

Cible Global Work de TMC.

Utilise Data-Frabric

Programmation local-global.

Scientific Computing Associate products **(SCA)**

PCGPACK, ...

En relation avec l'Université Yale, Martin Schultz et al.

Yale fut importante dans la génèse des bibliothèques,

avec le Yale Sparse Package. T. Chan, L. Johnsson, B. Gropp,

Y. Saad était alors à Yale, avec M.Schultz.

Parallel SPARSe LIBrary : **P-SPARSLIB**

Méthodes itératives, matrices creuses

Préconditionnements parallèles

Partitionnement de graphe, décomposition de domaines

Univ. Minneapolis, Youcef Saad et al.

NAG Parallel Library

ex NAG PVM library

Basée sur ScaLAPACK, PVM ou MPI

Parallel Utilities Library **(PUL)**

Basée sur le Common High Level Interface for Message Passing (CHIMP), porté sur MPI.

Load balancing, maillages non structurés.

Edinburgh (EPCC).

Parallel **(PCG++)**

Université du Texas à Austin

Écrit en C++

Parallel Iterative Methods **(PIM)**

Université du Rio Grande do sud, Br, et Université du Kent

IBM, Cray,.... : versions optimisées pour leurs machines de **ScaLAPACK**.

Machines japonaises?

Souvent basée sur MPI (avec parfois PVM ou outils constructeurs tel que CMMD pour TMC ou NX pour Intel, par exemple).

Un mailleur, pour illustrer un autre type de bibliothèque : **METIS**

Partitionneur de graphe et de matrices creuses.

Université de Minneapolis, G. Karypis et al.

Scalable Linear Algebra PACKage (ScaLAPACK) - 1 de 5 -

Programmation SPMD, utilise les BLAS, BLACS et les PBLAS 1,2 et 3

BLACS

Communications point à point, broadcast, combine, routines diverses.

Travaille selon une grille virtuelle 2D

2 classes principales de matrice :
- rectangulaire générale (N,M et LDA),
- trapézoïdale (N,M,LDA,UPLO et DIAG) ; les matrices triangulaires en sont une sous classe.

Scoped operations: ligne, colonne ou tous

Context: chaque grille virtuelle dans un *context* (communicateur MPI)

Création de grille : BLACS-GRIDINIT et BLACS-GRIDMAP (*context*)

Communications :
- non bloquantes : dès que message posté,
- localement bloquante (pour *send*): tampon libre pour autre message,
- globalement bloquante : confirmation ; *dead lock* possible

Deux classes de topologie pour les *broadcast*: anneau ou arbre (et hypercube)

Scalable Linear Algebra PACKage (ScaLAPACK) - 2 de 5 -

Exemple (extrait de la LAPACK working Note 94)

Déclarations

Initialisations

```
.../...
CALL BLACS-PINFO(IAM, NPROCS)
CALL BLACS-SETUP(IAM, NPROCS) ! la grille la plus proche d'un carré
.../...
! récupère le contexte et la grille définie
CALL BLACS-GET(0,0,CONTXT)
CALL BLACS-GRIDINIT(CONTXT,'ROW',NPROW,NPCOL)
CALL BLACS-GRIDINFO(CONTXT,NPROW,NPCOL,MYPROW,MYPCOL)
IF((MYPROW.GE.NPRO).OR.(MYPCOL.GE.NPCOL))GO TO 100 ! (pas dans la grille)
.../...
```

calcul avec BLAS, BLACS et PBLAS

```
.../..
CALL BLACS-PCOORD(CONTXT,ICALLER,HISROW,HISCOL)
.../...
100 CONTINUE
CALL BLACS-EXIT(0)
```

Scalable Linear Algebra PACKage (ScaLAPACK) - 3 de 5 -

PBLAS

Utilise les BLACS

Pxyyzz ; x = (S, D, C, Z), yy = (GE, HE, SY, TR,..), zz =(MM,MV,SV,..) + exceptions.

Matrices distribuées en *bloc cyclic* sur une grille virtuelle 2D.

Matrice 5 $\times$ 5 partitionnée en bloc de 2 $\times$ 2 :

$$\begin{array}{cc|cc|c}
a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\
a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} \\
\hline
a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & a_{3,5} \\
a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} & a_{4,5} \\
\hline
a_{5,1} & a_{5,2} & a_{5,3} & a_{5,4} & a_{5,5}
\end{array}$$

placée sur une grille de *process* 2 $\times$ 2 :

$$\begin{array}{|ccc|cc|}
\hline
a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\
a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} \\
a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & a_{3,5} \\
\hline
a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} & a_{4,5} \\
a_{5,1} & a_{5,2} & a_{5,3} & a_{5,4} & a_{5,5} \\
\hline
\end{array}$$

i.e. $(m, n) \rightarrow (p, q)$ et (i, j)

Descripteur : $(M, N, MB, NB, RSRC, CSRC, CTXT, LLD)$

Pour un vecteur 1D : $l \rightarrow (x, y)$, proc x et index local y.

Un *context* par matrice distribuée.

Scalable Linear Algebra PACKage (ScaLAPACK) - 3 de 5 -

Extrait de :

```
! -- PBLAS example code -- Written by Antoine Petitet, August 1995
! This program shows how to set the matrix descriptors and call
! the PBLAS routines.
```

Déclarations et initiations

```
.../...
CALL BLACS-PINFO( IAM, NPROCS )
CALL PDPBLASINFO( OUTFILE, NOUT, M, N, K, NB, NPROW, NPCOL, MEM, IAM, NPROCS
)
! Define process grid
CALL BLACS-GET( -1, 0, ICTXT )
CALL BLACS-GRIDINIT( ICTXT, 'Row-major', NPROW, NPCOL )
CALL BLACS-GRIDINFO( ICTXT, NPROW, NPCOL, MYROW, MYCOL )
IF( MYROW.GE.NPROW .OR. MYCOL.GE.NPCOL ) GO TO 20
.../...
! Initialize the array descriptor for the matrix A, B and C
CALL DESCINIT( DESCA, M, K, NB, NB, 0, 0, ICTXT, MAX( 1, MP ), INFO )
CALL DESCINIT( DESCB, K, N, NB, NB, 0, 0, ICTXT, MAX( 1, KP ), INFO )
CALL DESCINIT( DESCC, M, N, NB, NB, 0, 0, ICTXT, MAX( 1, MP ), INFO )
.../...
CALL PDNRM2( K, BNRM2, MEM( IPB ), 1, 1, DESCB, 1 )
.../...
CALL PDLAPRNT( M, K, MEM( IPA ), 1, 1, DESCA, 0, 0, 'A', NOUT, MEM( IPW ) )
.../...
CALL BLACS-GRIDEXIT( ICTXT )
20 CONTINUE
```

Scalable Linear Algebra PACKage (ScaLAPACK) - 4 de 5 -

Solution de systèmes linéaires, LU, autres factorisations, Cholesky, réduction sous forme d'Hessenberg, Problèmes de valeurs propres symétriques (bi-sections), moindres carrés, calcul de conditionnement de matrice,....

Prochaine version de ScaLAPACK ces mois ci, avec SVD,...

Trois types de routines :

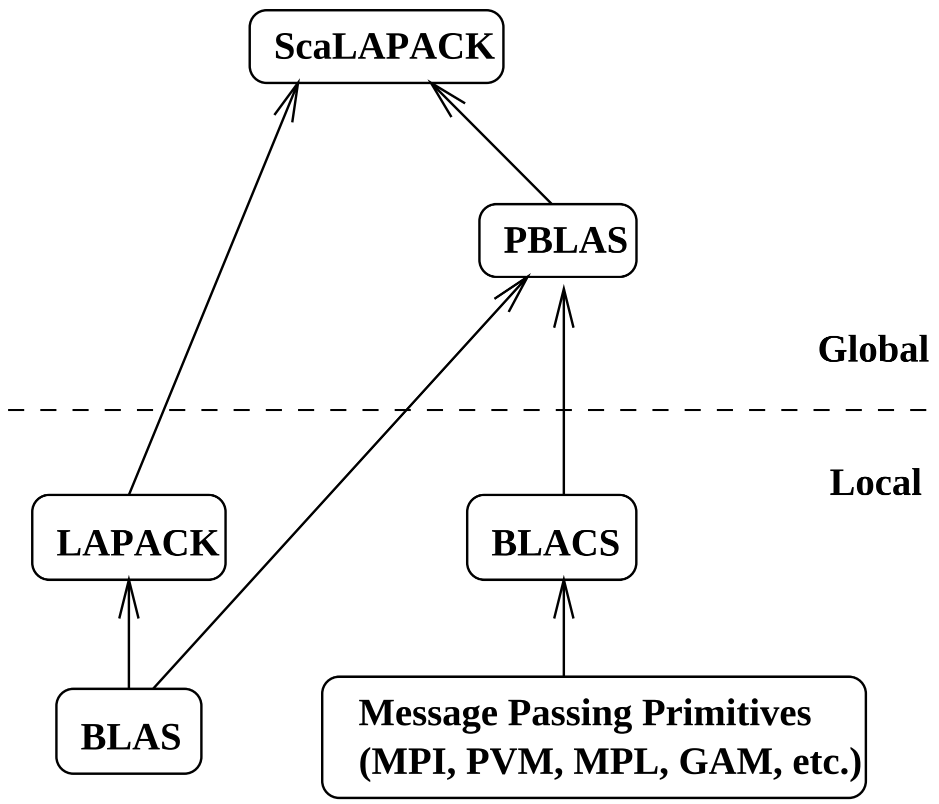
routines de calcul,

drivers : suite de routines de calcul,

routines auxiliaires : sous tâches.

Extrait du *draft* du manuel ScaLAPACK (merci à A. Petitet) :

ScaLAPACK Software Hierarchy



Scalable Linear Algebra PACKage (ScaLAPACK) - 5 de 5 -

```
SUBROUTINE PDSCAEXINFO( SUMMRY, NOUT, N, NRHS, NB, NPROW, NPCOL, WORK, IAM,
NPROCS )
```

`-- ScaLAPACK example code --`

`University of Tennessee, Knoxville, Oak Ridge National Laboratory,`
`and University of California, Berkeley.`

`Written by Antoine Petitet, August 1995 (petitet@cs.utk.edu)`

`This program solves a linear system by calling the ScaLAPACK`
`routine PDGESV. The input matrix and right-and-sides are`
`read from a file. The solution is written to a file.`

Déclarations, ...

`IF( IAM.EQ.0 ) THEN`

`Open file and skip data file header`

`Get matrix dimensions,...`

Appels effectués :

```
CALL BLACS-SETUP( IAM, NPROCS )

CALL BLACS-GET( -1, 0, ICTXT )

CALL BLACS-GRIDINIT( ICTXT, 'Row-major', 1, NPROCS )

CALL IGEBS2D( ICTXT, 'All', ' ', 5, 1, WORK, 5 )

CALL BLACS-SETUP( IAM, NPROCS )

CALL BLACS-GET( -1, 0, ICTXT )

CALL BLACS-GRIDINIT( ICTXT, 'Row-major', 1, NPROCS )

CALL IGEBR2D( ICTXT, 'All', ' ', 5, 1, WORK, 5, 0, 0 )

CALL BLACS-GRIDEXIT( ICTXT )

CALL BLACS-ABORT( ICTXT, 1 )
```

ScaLAPACK ; Parallel ARnoldi PACKage (PARPACK) - 1 de 2 -

Calculer les valeurs propres dominantes (et leurs vecteurs propres associés) de très grandes matrices creuses A non Hermitiennes.

Méthode Implicite d'Arnoldi Redémarrée (IRAM) de D. Sorensen :
synthèse entre la méthode d'Arnoldi et
la méthode QR avec shif implicite.

Utilise Fortran 77 et MPI ; programmation de type SPMD,
avec calculs redondants.

Les matrices, dont V, sont distribuées en bloc-lignes.

Trois parties : projection, calculs de Ritz, test d'arrêt et redémarrage

La phase de projection est composée d'une boucle sur j séquentielle comme suit, (avec v_0 donné et m fixé) :

$$v_j = Av_{j-1} - \sum_{i=0}^{j-1} h_{i,j-1} v_i, \tag{1}$$

$$\text{où } h_{i,j-1} = (Av_{j-1}, v_i); i = 0, j-1, \tag{2}$$

$$h_{j,j-1} = \|v_j\|_2, \tag{3}$$

$$v_j = v_j / \|v_j\|_2. \tag{4}$$

ScaLAPACK ; Parallel ARnoldi PACKage (PARPACK) - 2 de 2 -

Les opérations parallèles :

$w_j = Av_{j-1}$, **(PBLAS 2)**

$(w_j, v_i) = h_{i,j}$ pour $i = 1, j-1$, **(PBLAS 1), synchronisation**

$z_j = \sum_{i=1}^{j-1} h_{i,j} v_i$, **(PBLAS 1), synchronisation**

$v_j = w_j - z_j$, **PBLAS**

Deux synchronisations par itération.

Sur IBM-SP2, SGI Power Challenge, Cray T3E, ...

Gram Schmidt **classique**; avec techniques de réorthogonalisation.

Lanczos implicitement redémarré dans le cas Hermitien.

La seule partie de ScaLAPACK traitant vraiment des problèmes de valeurs propres non Hermitiennes.

ScaLAPACK ; PARallel PREconditioners

Très peu de docs disponibles.

Basée sur MPI et certaines routines de bas niveau de PETSc.

Préconditionnements parallèles :

Relaxation,

Bloc-SSOR, ILU,

Décomposition de domaines, ...

Algorithme de coloriage.

ScaLAPACK ; CAPSS

Très peu de docs disponibles (sans pages webs).

Résolution de $Ax = b$, A creuse, par méthodes directes.

Message passing.

CMSSL

Bibliothèque data parallèle de Thinking Machine (TM).

Systèmes linéaires (méthodes directes et itératives),

FFT,

valeurs propres : lanczos, IRAM,

Matrices denses,

Matrices creuses : format de base COO
et compilation des communications possibles sur CM-2(00).

Utilisable à partir de CM-Fortran et C*.

TMSSL,version Beta 1.0 - 1 de 2 -

Syntaxe basée sur les *tableaux*: TM/HPF, C*

Data Parallèle pour GlobalWork de TMC.

Local-Global pour C et F77 + Message passing.

Les *shapes* et les *lay out* sont encapsulés dans des descripteurs : les informations ne sont pas passées explicitement via la routine TMSSL.

TMSSL peut aussi être appelée à partir d'un programme orienté *message passing*

TMSSL,version Beta 1.0 - 1 de 2 -

Utilisation de **DataFrabric** ; bibliothèque local-global

Distribution *bloc cyclic.*

Un noyau de fonctions mathématiques appelables à partir de :

- TM-MPI
- C*
- DataFabric
- C ou F77 + Message passing avec global-local

Utilise BLAS, LAPACK et bibliothèques optimisées de chaque processeur.

Pour l'instant : PBLAS 1,2,3 ; LU; Inversion, FFT et tris de disponibles.

Outils pour le Global-Local:

Initiation

Initiation descripteur de tableaux

Redistribution

Accessoires : *local-to-global index conversion*, ...

Opérations sur tableaux.

Conversion CMSSL vers TMSSL : très limitée pour l'instant.

P-SPARSLIB

Partitionnement de graphe

Matrices creuses distribuées (Ax)

Localement CSR, Block Sparse Row ou dense

Ax :

i/ multiplication des blocs diagonaux B_{loc} par variables locales
ii/ prendre les variables externes nécessaires (vecteur distribué)
iii/ multiplier B_{ext} par ces variables locales et sommer

i/ et ii/ possible en parallèle ; B_{loc} et B_{ext} en CSR.

Algorithmes de sous espace de Krylov préconditionnés : FGMRES.
avec GM modifié.

Structure de P-SPARSLIB :

- GMRES, FGMRES, CG, etc...
- outils de préprocessing
- préconditionnements : Jacobi par blocs, SOR multicouleurs
- outils de passage de messages

PETSc - 1 de 2 -

Portable, Extensive Toolkit for Scientific computation

Méthodes implicites pour PDE.

Financé par DoE et NSF

Code en C, orienté objet, Utilisateurs Fortran, C et C++ ciblés

Avant d'installer PETSC : avoir BLAS, LAPACK, MPI

Versions pour DEC alpha, RS6000, Intel Paragon, IBM SP2,...

Permet de :

développer des codes pour problèmes de PDE,

prototyper rapidement des infrastructures d'algorithmes,

accéder à un environnement de programmation parallèle.

Encapsulation des données : code indépendant de la représentation des matrices creuses, par exemple.

PETSc - 2 de 2 -

Contient :

Noyau autour des méthodes de Krylov ; pour matrices creuses.

Assemblage des matrices

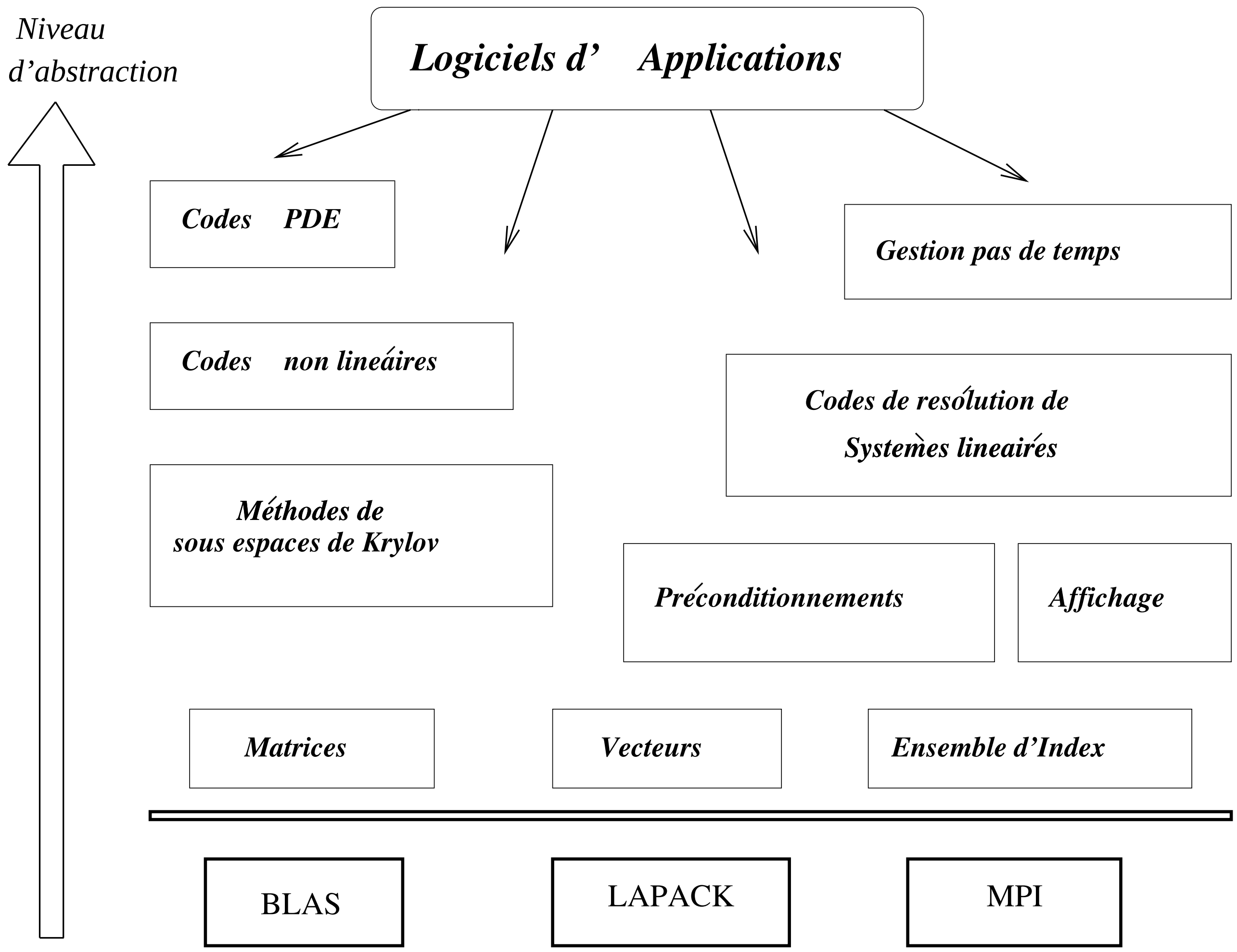
Préconditionnement

Profilling

Visualisation des calculs

Calculs avec matrices creuses (Yale Sparse Format : CSR)

Niveau d'abstraction, extrait (et traduit) de la doc PETSc :



Scientifique Computing Associate (SCA)

Lié étroitement avec le Département d'Informatique de l'Université Yale.
Martin Schultz, Andrew Sherman,...

Offre complète, logiciels parallèles pour matrices creuses, linda, etc...

Méthodes itératives et directes pour la résolution de
système linéaire *creux*: **PCGPAK**.

Matrices symétriques ou non.

Basée sur les méthodes de type Gradient-Conjugués

GMRES, préconditionnements

Version la plus récente: PCGPAK3,
est adaptable sur de nombreuses machines parallèles.

Pas de distributeur, pour le moment, en Europe.

Il existe de plus en plus de petites sociétés proposant des logiciels numériques avancés parallèles. Notons, par exemple, le développement de **Elegant Mathematics** créée par des numériciens venant principalement de l'Académie des sciences Russe (W. Yeremin,..) et domiciliée aux USA (Seattle) et qui propose dans le monde des codes parallèles pour certains problèmes difficiles.

III - Des bibliothèques vers les *toolkits* pour le calcul scientifique parallèle - 1 de 2 -

Beaucoup de bibliothèques parallèles orientées messages passing (MPI devenant la référence). Une seule bibliothèque complète est réellement data parallèle (CMSSL).

De nombreuses bibliothèques considèrent le cas des matrices creuses, mais avec des patrons statiques, en général.

Très peu proposent le calcul de problèmes de valeurs propres non Hermitiennes avec les dernières méthodes d'accélérations.

Peu de préconditionnements parallèles sont proposés, polynômiaux?

Les bibliothèques actuelles sont très bien adaptées pour les problèmes *in core*

Il n'existe pas néanmoins de résolution complète de grands problèmes de valeurs propres nom Hermitiennes

Chaque domaine scientifique a ses bibliothèques, ou codes, spécifiques.

III - Des bibliothèques vers les *toolkits* pour le calcul scientifique parallèle - 2 de 2 -

De plus en plus d'approches orientées objets.

Outils supplémentaires: *drivers*, visualisation, *profilling*, redistribution.

Pas encore vraiment MIMD et hétérogène.

Besoin d'une certaine forme d'interactivité, avec visu dynamique

Ajuster certains paramètres demande beaucoup d'expertise

Les tâches sont toujours séquentielles, alors que les processeurs peuvent être fortement vectoriels (BLAS vectoriels alors utilisés, mais quand on calcul un QR comme dans les méthodes de type Arnoldi?)

Problème du passage entre les modèles de parallélisme.

Problèmes des I/O parallèles

Choix des structures de données comme objet.

Répartitions des données entre unités parallèles et séquentielles

De bibliothèques classiques parallèles on se dirige vers des kit d'outils pour le calcul scientifique parallèle.

IV - Du *out of core* et autres problèmes actuelles

De plus en plus d'applications doivent "tourner" en *out of core.*

Cela demande une paramétrisation supplémentaire et des balayages de données optimisées. D'où la nécessité de PBLAS *paginables.*

Aucune bibliothèque ne propose cela ; certaines les envisagent dans les versions futures .

Problème de la répartition des données, régulières ou non, sur les disques, les bases de données, les *front-end.*

Choix de certains paramètres, comme la taille des sous-espaces pour les méthodes de type Krylov très utilisées dans les bibliothèques numériques parallèles.

Expression de l'hybridation et de la répartition des calculs selon les modèles de programmation.

Méthodes hybrides et bibliothèques.

Tests de convergences, arithmétiques différentes entre processeurs

Structures internes pour matrices creuses et problèmes des changements de formats parallèles.

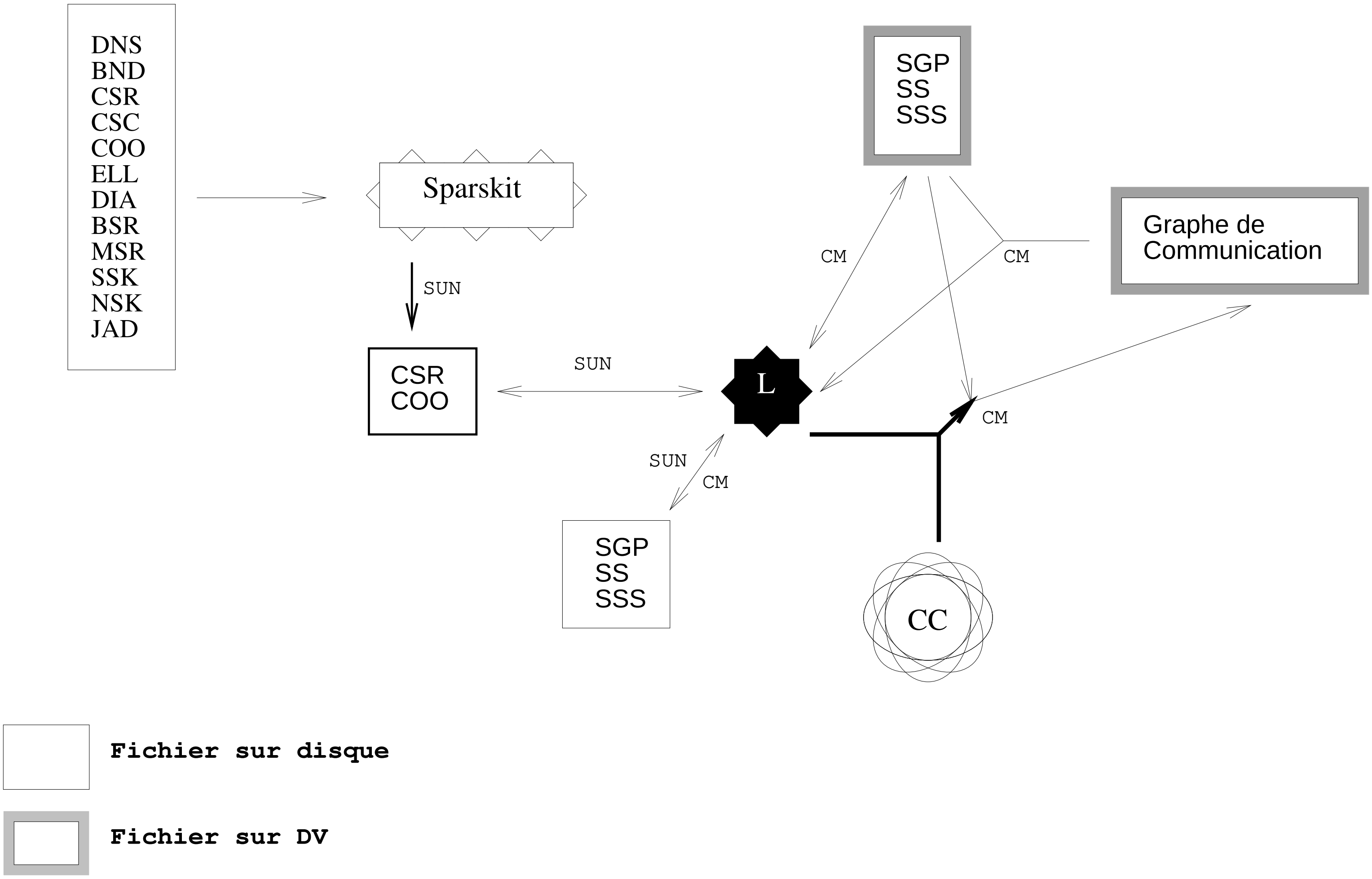


FIG. 1 - *Exemple : structure d'une version data parallèle du noyau de SPARSKIT, SEH, 1991*

Et, problèmes des structures de données irrégulières dynamiques, venant de remplissages locaux des matrices et/ou maillages adaptatifs, par exemple.

Certains paramètres de méthodes numériques itératives sont très difficiles à ajuster. L'utilisateur doit très bien connaître les méthodes présentes dans les bibliothèques.

Par exemple, la taille du sous-espace de Krylov dans une méthode d'Arnoldi avec redémarrage est très délicate à choisir :

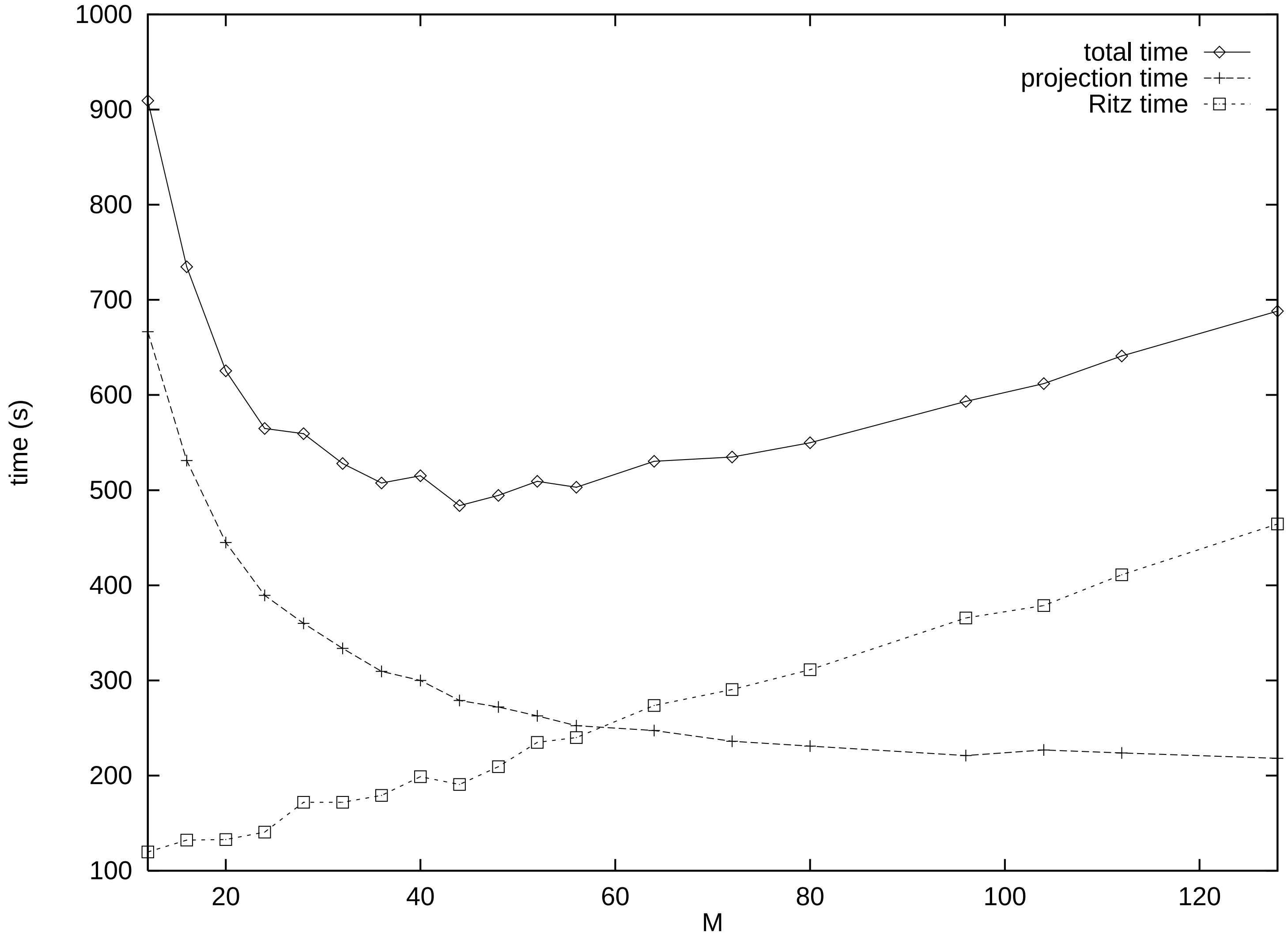


FIG. 2 - *Temps de calcul cumulé pour les diverses parties de la méthode d'Arnoldi jusqu'à convergence, exemple avec* $r = 1$

Il existe une taille optimale pour une matrice et une machine fixées.

Vers une offre globale **toolkit + expert on-line**

VI Conclusion

Il existe de nombreuses bibliothèques parallèles accessibles pour les applications bien conditionnées et de tailles *normales*; souvent basées sur ScaLAPACK + MPI, avec PBLAS 1,2,3 et BLACS optimisés par les constructeurs [IBM, CRAY, Intel,....) ou distributeurs des logiciels (NAG,...).

Elles ont été pensées dans les années 80-début 1990.

Les structures sont parfois rigides

Les plus nouvelles sont de plus en plus orienté objet et de plus en plus souvent écrites dans un autres langages que Fortran.

Elles sont surtout SPMD et avec des tâches séquentielles

Il y a des tentatives de passage de phases data parallèles à des phases messages passing dans la même bibliothèque.

VII Perspectives et prévisions

Vers des ateliers de programmation scientifique parallèle assistés d'experts.

Orienté objet.

Matrices creuses et maillages adaptatifs

Out of core

Modèle de programmation comme *objet*

Outils de visualisation, y compris interactifs.

Liens nécessaires entre développeurs des bibliothèques et les équipes de recherches (numériques, parallèles et objet).

Drivers parallèles

Vrai MIMD.

KrylovKit

Projet commun LIFL, Lille, et PRiSM, Versailles.

Bibliothèque basée sur les méthodes de Krylov : Arnoldi, Hybrid ARNOLDI(V,M,l,r), Arnoldi par blocs, FGMRES(v,m), QMR, BiCGStab, PCG,... Pour les problèmes de grandes tailles.

- Systèmes linéaire, **avec plusieurs seconds membres**
- Calcul de valeurs propres
- Accélération de la convergence
- Préconditionnement hybrides (polynômiaux)
- Plusieurs tests de convergences (Objet)
- Méthodes hybrides
- Calcul sur chaque processeurs data parallèles et/ou vectorielles ; et vrai MIMD
- Format de matrices creuses dynamiques
- Objet `modèle de programmation parallèle`
- Interface graphique (choix des objets) et documentation on line en html

KrylovKit, lancé en Avril 1996, est en phase de spécification. Une partie des codes parallèles existent déjà. Recherches de collaborations industrielles (Adulis,...)