An unified linear equation solvers interface for industrial softwares

Michel Belliard, Bernard Sécher – CEA/DEN

Numerical Platon

Context

– CEA strategy to promote massive parallel softwares

– A tool to help the development of new generation simulation codes

– To reach multiple available libraries of linear equation solvers from a single interface without any change in the code

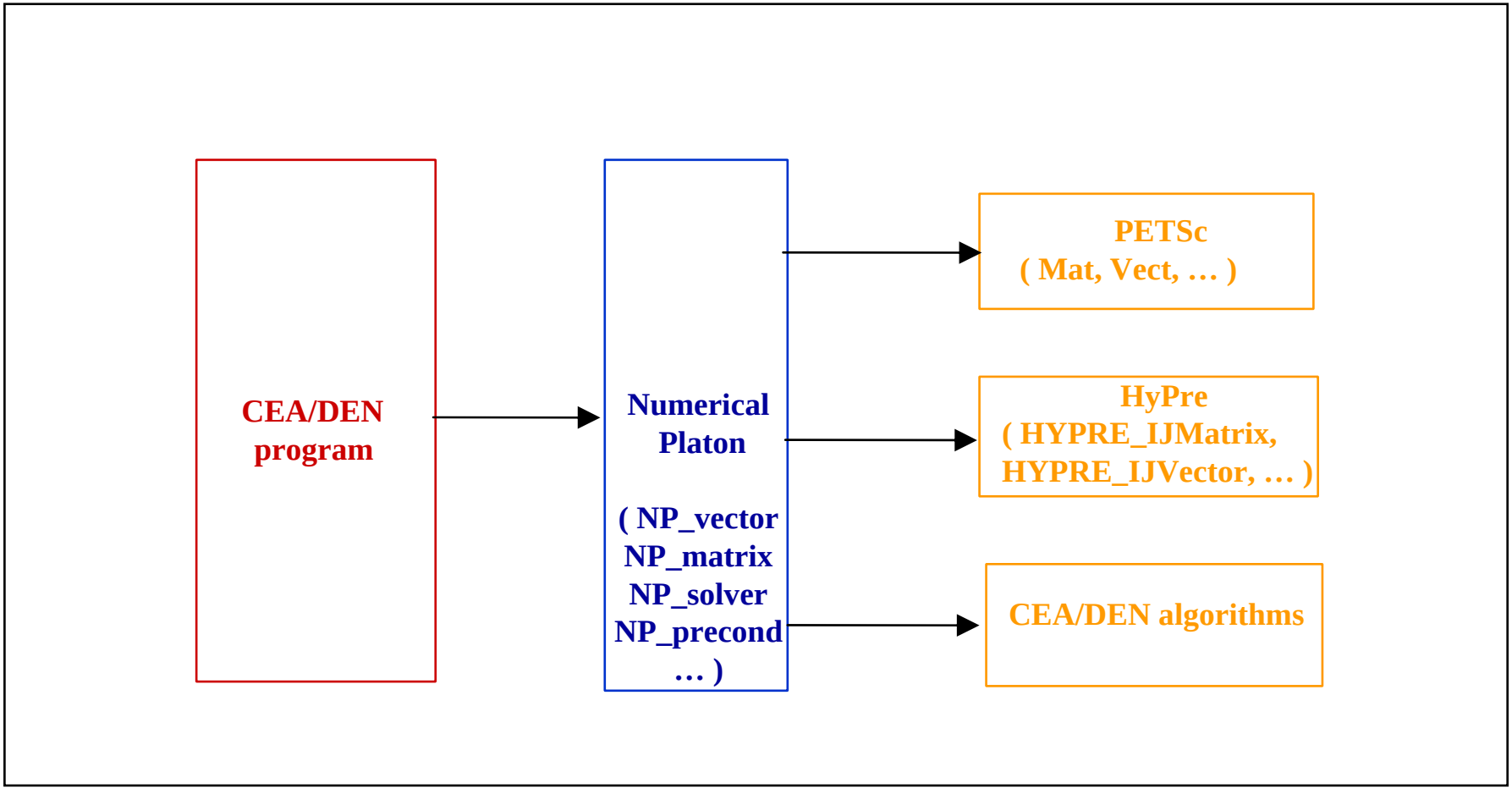
– To manage distributed and shared parallelism in a transparent way for the user

Numerical Platon

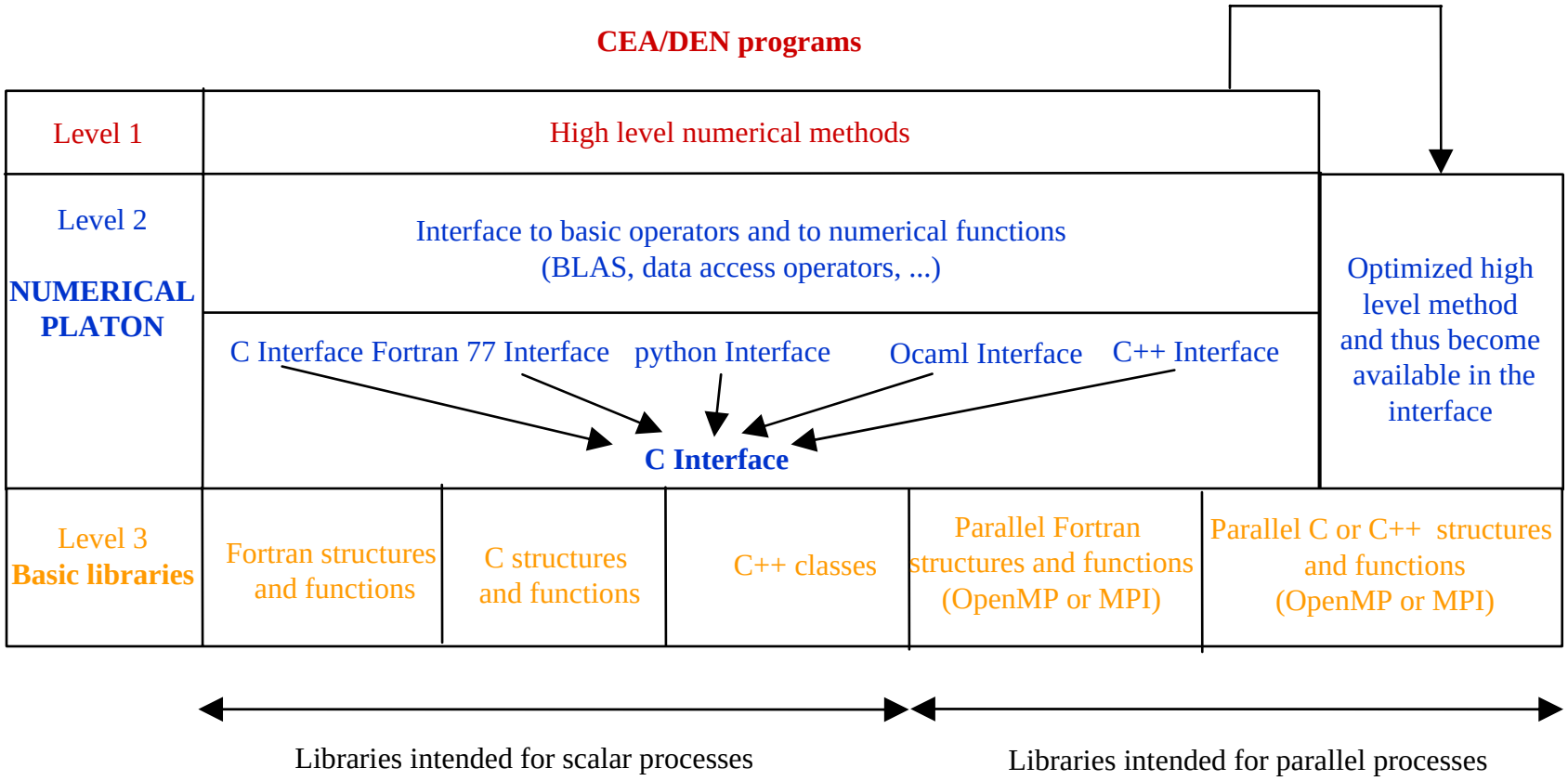
Goals

- Only one standardized interface available in various languages (C, C++, Fortran77, Ocaml, python)
- Using of the best algorithm in the available libraries
- Limited to vectors and matrices for linear equation solver: A x = b
- Promote code reusability, flexibility and portability
- To be evolutive
- To support data and processing parallelism, but optimal on scalar machines
- To offer primitives of read/write on file while masking the problems of the parallel accesses
- Porting and installation: IBM/aix, PC/linux, alpha/osf, SUN/solaris, SGI/irix, HP/UX

Numerical Platon



Numerical Platon



Numerical Platon

- Available libraires

PETSc	www-unix.mcs.anl.gov/petsc	Direct and iterative solvers for dense or sparse matrices	distributed
ScaLapack	www.netlib.org/scalapack	Solvers and basic linear algebra functions for dense or sparse matrices	distributed
Sparskit and pARMS	www.cs.umn.edu/~saad/software/home.html	a basic tool-kit for sparse matrix computations	distributed
NagMPI et NagSMP	www.nag.co.uk/numeric/numerical_libraries.asp	Commercial libraries: direct and iterative solvers for dense or sparse matrices	distributed and shared
IMSL	http://www.vni.com/products/imsl/index.html	Commercial libraries: BLAS and solvers for dense matrices	distributed
Aztec	http://www.cs.sandia.gov/CRF/aztec1.html	Iterative solvers	distributed
HyPre	http://www.llnl.gov/CASC/linear_solvers/	Iterative solvers for sparse matrices	distributed and shared
Compaq cxml library			shared

Numerical Platon

- Available solvers

  - Use of PETSc for distributed parallelism, associated with the following softwares: BlockSolve95, SPAI, SuperLU, LAPACK and BLAS
  - Use of HyPre for distributed parallelism
  - Development at CEA/DEN in OpenMP for shared parallelism
  - Use of multi-threads version of SuperLU for shared parallelism

  - Direct methods: Cholesky and LU factorizations with reorderings
  - Iterative methods: CG, GMRES, BICGSTAB, CGS, TFQMR, multigrids with preconditionning: diagonal, SSOR, ICC(k), ILU(k), PILUT, sparse approximate inverse, polynomial and additive Schwarz

Numerical Platon

- Use of NP at CEA: GENEPI, OVAP, APOLLO2, FLICA4, TRIO-U, Alliances

  – difficulty of integration in an existing sequential code (GENEPI, APOLLO2, FLICA4, OVAP)

    => sequential assembly of the matrix

    master/slave mode for the parallel resolution

    (ask an additional effort of reorganization of the code to obtain a code really SPMD)

  – facility of integration in a new code (prototype for Alliances project) or in an existing parallel code (TRIO-U)

    => assembly of the matrix and resolution in parallel (SPMD)

Numerical Platon

- Use of NP in TRIO-U(1):
  - CFD development platform
  - Specific applications in CFD for large simulations on complex geometries
  - Structured and unstructured meshes
  - Trio-U/PRICLES : LES
  - Trio-U/SND : DNS 2-phase flows
  - UML conception, C++ implementation
  - Parallel architecture
  - Portability from PC (linux) to supercomputers (HP/SC,VPP500,…)
  - NP is used first for HyPre solvers: CG with ILU(k), GMRES and BICGSTAB
  - Facility of use of several different solvers simultaneously (HyPre + PETSc...)

Numerical Platon

- Use of NP in TRIO-U(2):
  - NP is an alternative to intern Trio-U linear solvers (GC + SSOR or Cholesky)
  - Use in large scale of simulations: from 1 to 40 processors, from 100,000 to 5 millions unknown factors
  - Advantages of NP on intern Trio-U solvers:

  Sometimes intern Trio-U solvers don't converge, NP yes

  Intern Trio-U solvers good for symmetric matrices, NP good for symmetric or unsymmetric matrices

  Sometimes NP solvers are speeder than intern Trio-U solvers
  - Disadvantages of NP on intern Trio-U solvers:

  Sometimes NP solvers don't converge, intern Trio-U solvers yes

  Some pre-conditioning are expensive in time

  Sometimes NP solvers are slower than intern Trio-U solvers

Numerical Platon

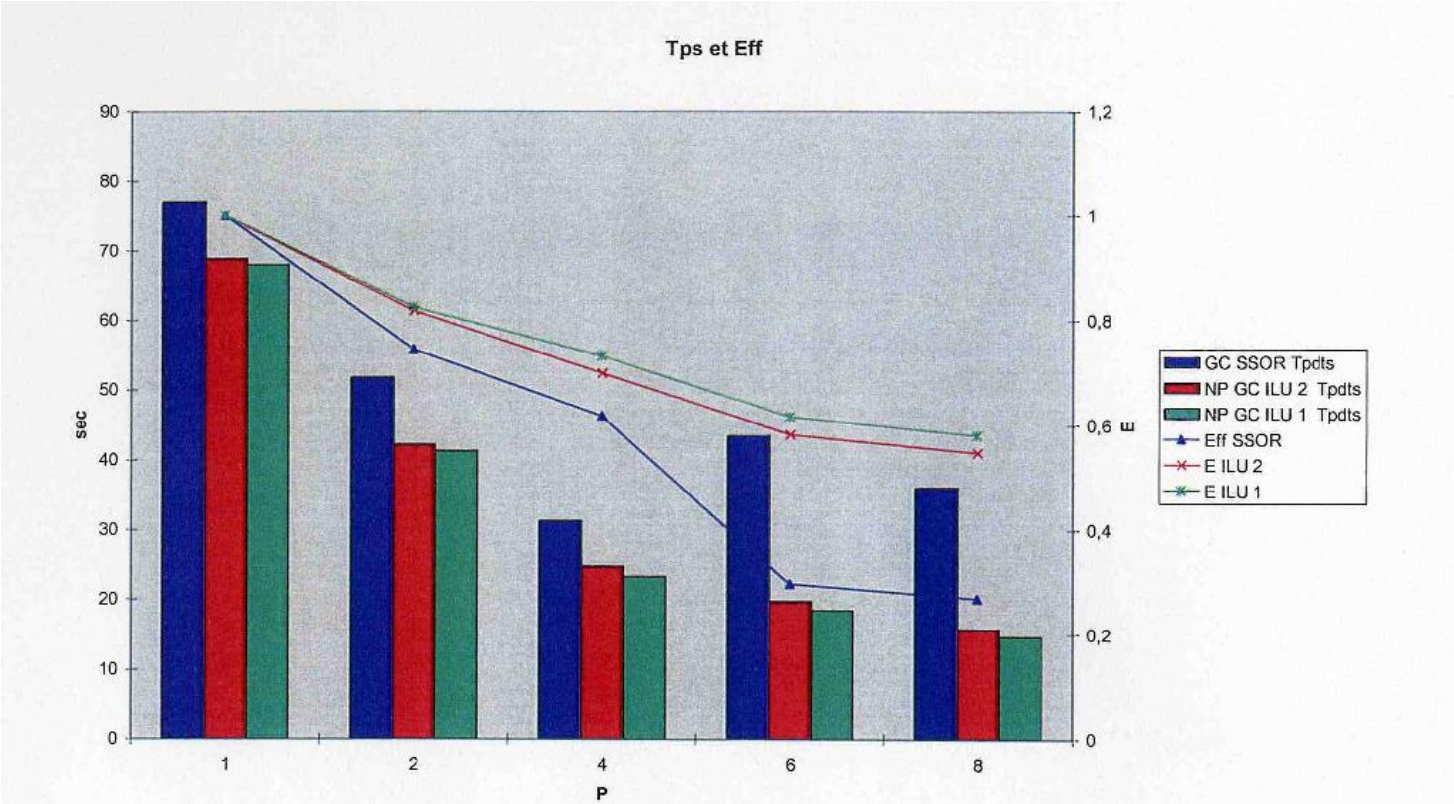


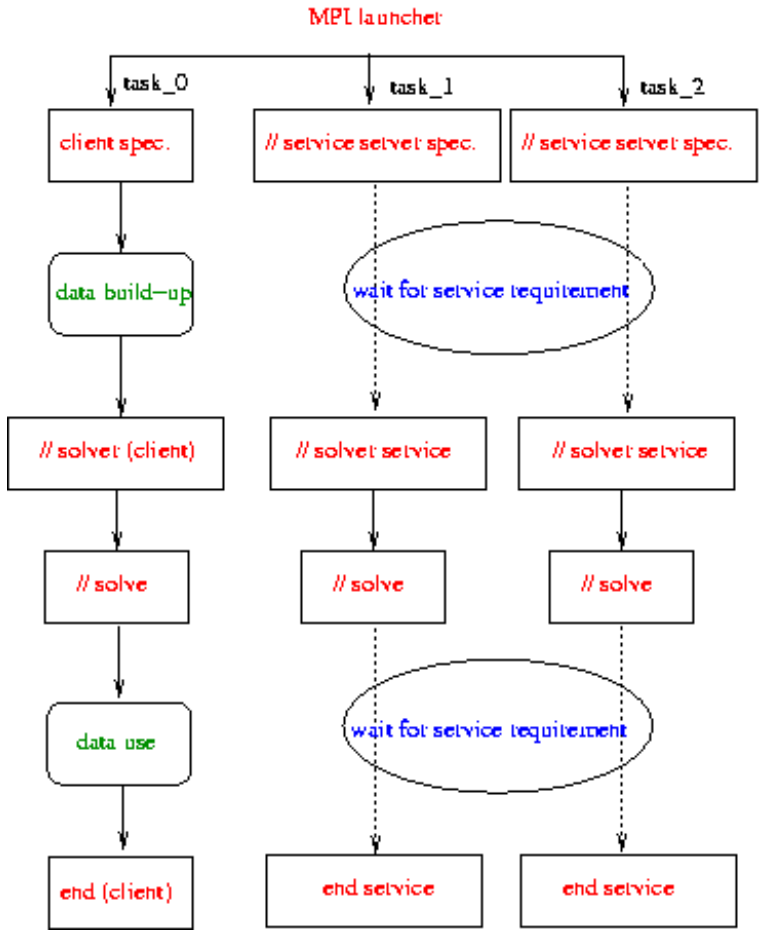
Figure 16 : Performances parallèles sur cluster de PC.

Numerical Platon

- Use of NP in GENEPI (1):
  - 3D two-phase flow simulations of steam generators
  - Easy re-engineering
  - Server of // services for distributed parallelism

  client: sequential ass. of data

  client + servers: // resolution
  - Useful solver toolkit for numerical tests whitout code change, even for sequential runs
  - Parallelization of home solvers using vectors and matrix NP routines



Numerical Platon

- **Use of NP in GENEPI (2):**
  - A example: the pressure solve by a projection scheme
    - Initial => LU solve; 150,000 cells; 1.9 Go
    - NP => PDCG solve (10^{-9}); 500,000 cells; 1.7 Go
    - CEA HP ES45 cluster: 4 proc. EV68 / 4 Go
    - 1 task / node => distributed memory
    - Sequential: 1 pressure solve => 90% of 1 time step CPU time

Proc. Number	CPU time (s)	Speed-up
1	1531	-
4	415	3.7
12	120	12.7
16	95	16.1

Development plan

- V2.6 version available (PETSc, BlockSolve95, SuperLU, SPAI, HyPre, CEA developments in OpenMP) with documentation: « Users guide and reference manual » with LGP license
- Next version interfacing with SLOOP (CEA/DAM) end of 2006
- Technological survey : new version of the existing libraries and new libraries