Coupling Multi-level Component Interfaces for Parallel Sparse Linear System Solvers

CCA Background:

Create a standard component architecture for High-Performance Computing. Well-designed interface is the key for the CCA model.

Many different universities and government laboratories are involved.

Includes Babel, a language interoperability tool, and Bocca, a component development tool.

SPARSKIT and pARMS

SPARSKIT: Sparse Matrix library written by Y. Saad(University of Minnesota in the 90s).

Provides a range of functionality and iterative techniques. Loses popularity due to outdated design and Fortran77 implementations.

pARMs: Parallel Algebraic Recursive Multilevel Solvers.
Provides a library of parallel iterative solvers for distributed sparse linear systems of equations. Offers a large selection of preconditioners for distributed sparse linear systems and several accelerators.

Builds on SPARSKIT kernels.

Motivation: Provide hierarchical levels of usability

HI->MI: Provide a high-level view to MI components.
- Package MI components to implement HI interface.
- Allow application easy to access MI components.

MI->HI: Allow HI components to access state-of-the-art MI iterative method components.
- Make HI component use MI components (matvec, preconditioner or accelerator) via call-back.
- Extend HI components' functionality through MI components.

SPARSKIT Toolkit:

Developed according to CCA Toolkit specifications. Includes:
- Three accelerator components.
- Five preconditioner components.
- Three arms last level preconditioner components.
- Three matrix vector multiply components
- Shared matrix data component.
- General matrix solver driver.
- Matrix generation component.
- BLASSM components.

Download available at:
http://www.scl.ameslab.gov/Projects/Sparskit-CCA

Parallel SPARSKIT Toolkit

Designed to extend sequential MI Interfaces.

-**GenericMatrixFormat** serves as matrix storage and converter into the formats acceptable by the chosen preconditioner components.

-**GenericDistributedMatrixFormat** extends the GenericMatrixFormat interface with a "set/get" methods for the (global) row distribution.

-**GenericPartitioner** partitions matrix data for efficient distributed matrix-vector multiplication and overall solution.

Separation of concerns in medium-level design

-Component GenericMatrixFormat encapsulates matrix format, transformations, inquiries, and matrix storage.
- Since MI is designed a priori for expert MI users
- Wiring diagrams now resemble star topology.

Benefits
- Increased component reuse.
- Easy addition of new matrix formats.
- Simplification of MI design (more light-weight components).

Solver Interface Granularity Levels:

Low-level (LI): Refers to operating on objects such as matrices directly. One way to accomplish is via function overloading.

Example:
```
matrix1->type = "CSR"
matrix2->type = "CSR"
blassm.amub(matrix1,matrix2)
```

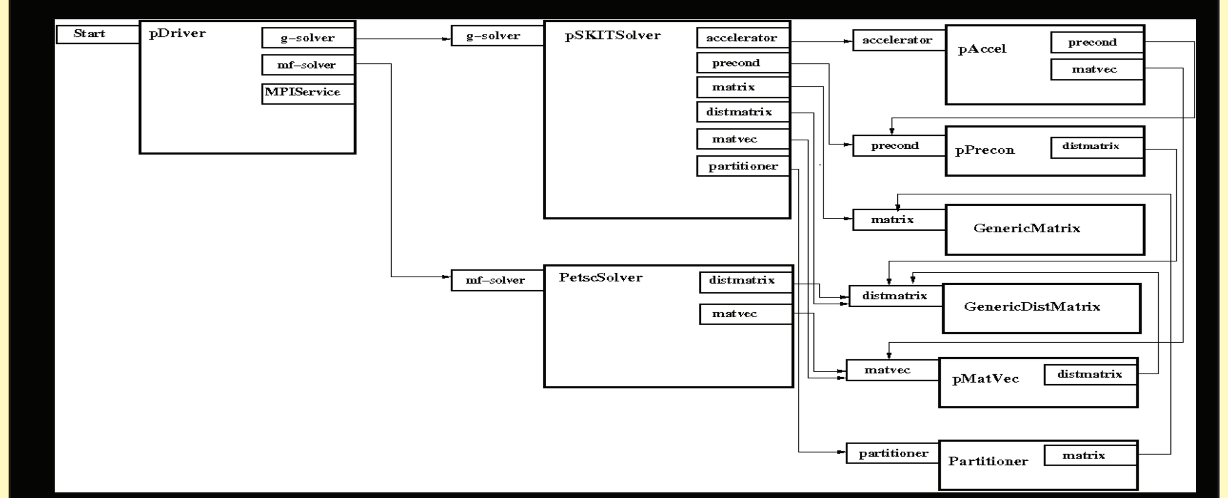
Medium-level(MI): More abstract than low-level. Major steps of an iterative solution procedure are encapsulated as components. These steps include accelerator, preconditioner, and matvec operations as well as a shared data component for holding the matrix.

Example:
```
accelerator.apply(<RHS Arguments>)
preconditioner.create(<ilu arguments>)
matrix.create(<format type and data arrays>)
```

High-level(HI): Most abstract design choice. Treated as black box and all steps of the iterative solution procedure are included in one component. CCA-LISI from Indiana University TOPS interfaces (SciDAC-2 project).

Example:
```
solver.setupMatrix(<Matrix arguments>)
solver.setupRHS(<RHS arguments>)
solver.set("solver","GMRES")
solver.set("preconditioner","ilu")
solver.solve(<Solution arguments>)
```

Multi-Level Interface Coupling Architectures



Resources:

pARMS	*http://www-users.cs.umn.edu/~saad/software/pARMS/*
SPARKSIT	*http://www.users.cs.umn.edu/~saad/software/SPARSKIT/sparskit.html*
Babel	*https://computation.llnl.gov/casc/components/babel.html*
Bocca	*https://www.cca-forum.org/wiki/tiki-index.php?page=Bocca*
CCA	*http://www.cca-forum.org*
CCA-LISI	*http://www.cs.indiana.edu/~fangliu/lsi/intro.html*
Ccaffeine	*http://cca-forum.org/ccafe*
TOPS	*http://www.scidac.gov/math/TOPS.html*

(Serial Experiment) Easy Switching of Preconditioner Components:

PETSc using SKIT components

Matrix	its	Mf-Func	Mf-MI
sherman	100	0.17	0.17
parmobil	22	0.44	0.44
bcsstk16	76	0.71	0.71

SKIT medium and high level on Parmobil

Precon	its	SKIT	SKIT-MI	SkitSolver
VBILUT	*	6.39	6.44	6.43
VBILUK	23	78.19	77.79	77.57
ILUC	*	2.54	2.6	2.61
ARMS+ILUTP	31	0.82	0.87	0.86
ARMS+ILUTD	32	0.87	0.87	0.86
ARMS+ILUC	31	0.83	0.85	0.85

(Parallel Experiment) Parallel MI Components and HI encapsulation:

pArms medium and high level on Borges

Matrix	#proc	its	pArms-MI	LISI-pArms
mat100x100	1	23	0.33	0.34
mat100x100	4	129	0.30	0.31
mat100x100	16	209	0.41	0.41
sherman5	1	17	0.083	0.091
sherman5	4	38	0.075	0.083
sherman5	16	114	0.145	0.168

PETSc using pArms components
on mat100x100 on matrix-free method

#proc	its	Mf-MI
1	113	0.29
4	189	0.74

Contact Information: Fang Liu (fangliu@scl.ameslab.gov), Masha Sosonkina (masha@scl.ameslab.gov)

National Institute of Standards and Technology – Mathematical and Computational Sciences Division