5. 各会合での富士通からの情報提供資料

富士通からの情報提供として、以下を行った。

・Open Petascale Libraries (OPL) 活動状況報告

OPL 活動は，アプリケーション起点によるライブラリ開発を目指している．すなわち，各アプリケーション分野で要求される要素技術を分析し，需要の高い要素技術を高速ライブラリとして提供することを活動目的とする．要素技術の分析結果に基づき，以下の 4 分野について高速ライブラリをスーパーコンピュータ・PC クラスタ向けに開発・移植・高速化することを決定し，活動を続けている．

・密行列計算
- PLASMA/DPLASMA

・大規模偏微分方程式によるシミュレーション
- PETSc
- spBLAS
- PRAgMaTIc

・高速フーリエ変換 (FFT)
- FFTE
- 2DECOMP&FFT

・乱数生成
- MRG8 他

・OPL 活動で提供しているライブラリの実アプリケーションへの適用事例紹介

OPL 活動の中間成果として，開発・移植・高速化したライブラリを実際のアプリケーションに適用し，適用性・性能を評価した．1 つは平面波基底による密度汎関数法アプリケーション Quantum ESPRESSO への FFT ライブラリ適用である．FFT を利用する典型的なアプリケーションであり，FFT 計算部の計算負荷が高い．もう 1 つはフラグメント分子軌道法アプリケーション OpenFMO への密行列計算ライブラリ適用である．密行列計算を利用する典型的なアプリケーションであり，特に固有値計算部の計算負荷が高い．両者とも，より高速な計算が望まれている分野であり，高速ライブラリの適用は関心が高い．適用評価は，FFT 計算部を FFTE に，固有値計算部を PLASMA に置き換えることで実施した．それぞれ，ライブラリ固有の課題が見つかり，ライブラリの在り方そのものについても再認識することとなった．

・富士通の数学ライブラリ取組み紹介

富士通におけるライブラリ開発の立場から，現在の製品である SSL II について，その経緯，開発方針，スーパーコンピュータにおける性能，方針などについて紹介した．

また，現在の数値計算ライブラリを取り巻く環境として，OSS のライブラリの現状について調査した結果を紹介した．

次ページ以降に，以下の資料を掲載する．

No	実施日	タイトル	情報提供者（所属は当時）
1	2011/08/23	Open Petascale Libraries : Application Requirements	R. Nobes (欧州富士通研究所) J. Southern (同上) R. Saksena (同上)
2	2011/10/28	OPL 状況報告(2011 年 10 月)	金澤　宏幸（富士通）
3	2011/10/28	FFTE ライブラリ適用事例(Quantum ESPRESSO)	堀田　普介（富士通）
4	2012/03/01	大規模数値計算における数学ライブラリの取り組みについて	臼井　徹三（富士通）
5	2012/07/23	Open Petascale Libraries : Current Status	堀田　普介（富士通）
6	2013/02/22	Application of PLASMA to OpenFMO code	堀田　普介（富士通）

FUJITSU
shaping tomorrow with you

Open Petascale Libraries: Application Requirements

Fujitsu Laboratories of Europe

Hamamatsucho, 23 August 2011

Outline

FUJITSU

1. Introduction to Fujitsu Laboratories of Europe
2. The Open Petascale Libraries Project
   - Progress and Plans
3. Application Requirements
4. Discussion

FUJITSU

Fujitsu Laboratories of Europe

- Introduction

FUJITSU

Open Petascale Libraries Project

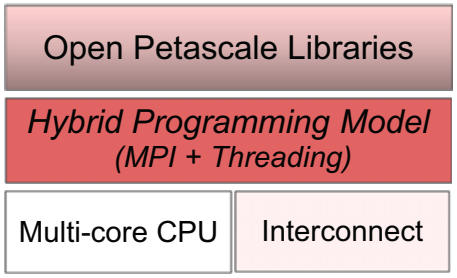
- Introduction

Open Petascale Libraries

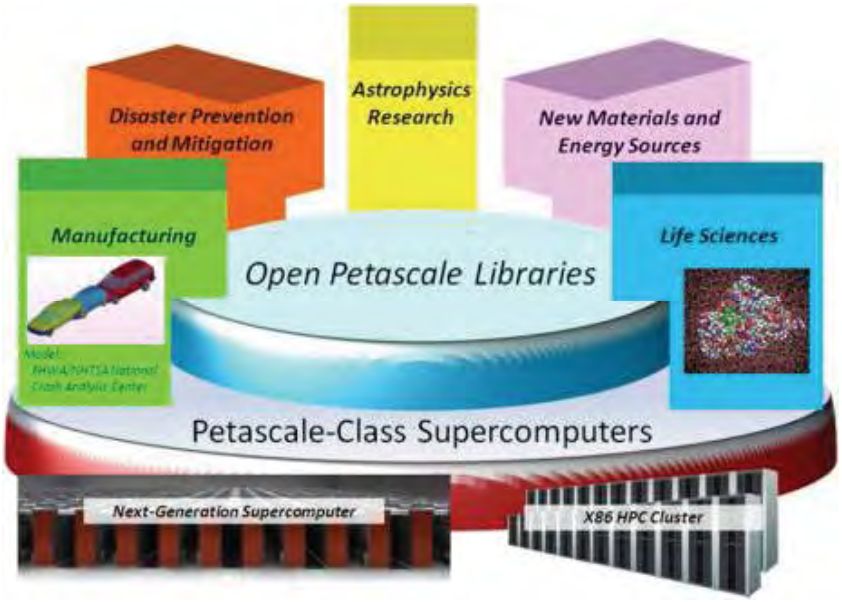
FUJITSU

Global collaboration to develop advanced numerical software for supercomputing

- Dedicated forum to promote the open exchange of ideas and the collaborative development of general-purpose and application-specific heterogeneous numerical libraries
- Targeted at parallel computers built from multi-core processors
- All output available as open-source software



openpetascale LIBRARIES



Providing a software platform to accelerate applications running on massively parallel multicore supercomputers

OPL Members

FUJITSU

ANU THE AUSTRALIAN NATIONAL UNIVERSITY

MANCHESTER 1824

理化学研究所 RIKEN

Imperial College London

NII

ICL

nag

Scientific Systems

Institute of High Performance Computing A*STAR

oerc Oxford e-Research Centre

Science & Technology Facilities Council

JAIST JAPAN ADVANCED INSTITUTE OF SCIENCE AND TECHNOLOGY 1990

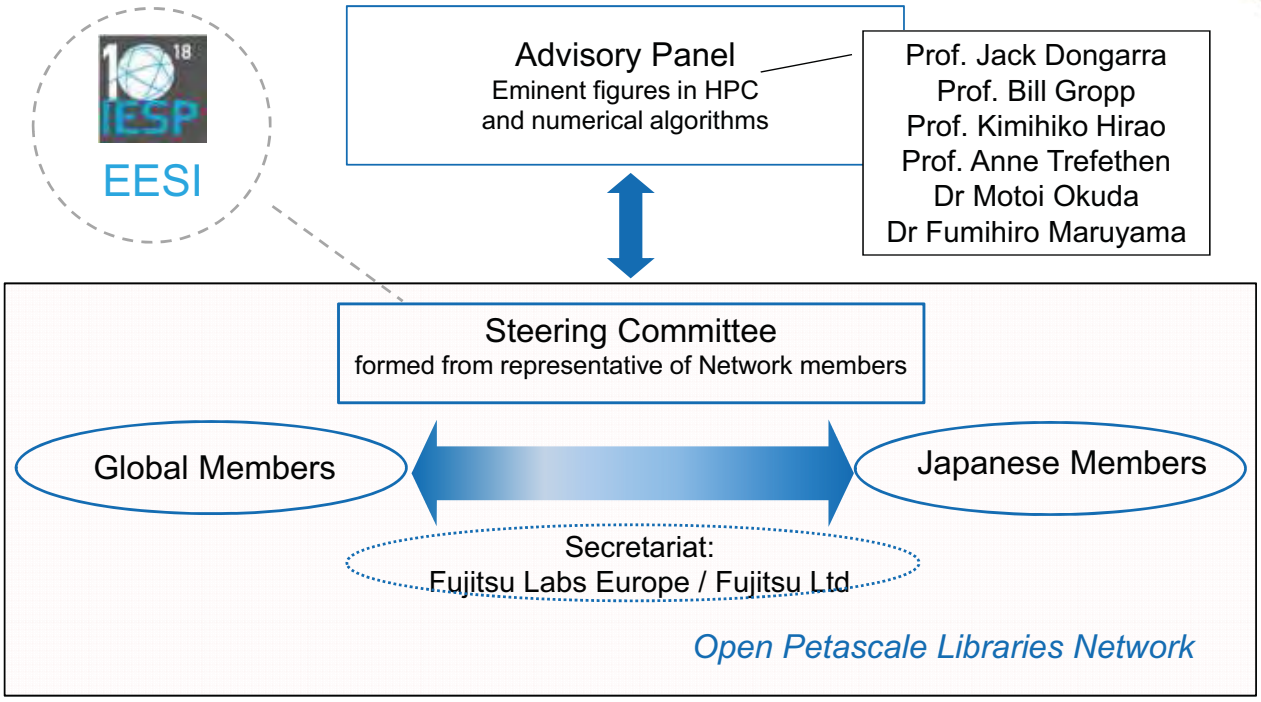
FUJITSU

UCL

OPL Membership

FUJITSU

OPL: Organisation



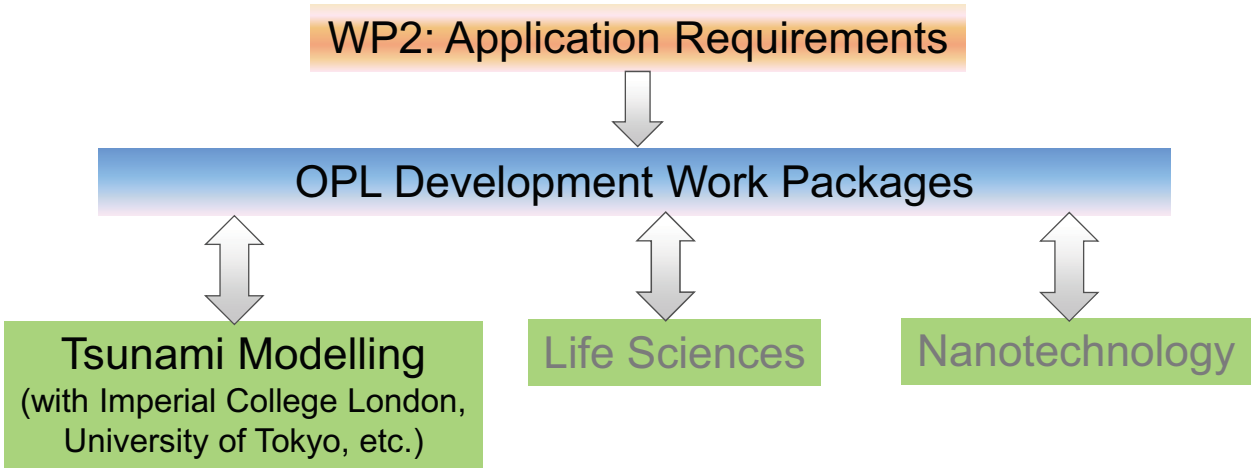
Current Projects

- WP3: Dense Linear Algebra
  - PLASMA/DPLASMA (based on directed acyclic graphs, with Professor Jack Dongarra)
- WP4: Large-Scale PDE-Based Simulations
  - Hybrid PETSc (with Imperial College and Argonne National Laboratory)
  - SpBLAS (threaded Level 3 sparse BLAS, in-house)
  - PRAgMaTIc (hybrid mesh adaptation library, in-house)
- WP5: Fast Fourier Transforms
  - FFTE (from Dr Takahashi, Tsukuba)
  - 2DECOMP&FFT (from Numerical Algorithms Group, NAG)
- WP6: Random Number Generators
  - Led by Professor Kenichi Miura, NII

Application-Driven Approach

Application Driven Development

- Advisory Panel (Hirao, Dongarra, Gropp, Trefethen): “it is important that the OPL work be driven by the application requirements”



Application ‘Co-Design Vehicles’

Application Requirements

FUJITSU

- Key goal of OPL: to provide applications with the libraries they need to run effectively on the emerging generation of supercomputers
- Vital that effort within OPL is focused on libraries that are of highest importance to users
  - Success of an open source project dependent of uptake from user community – who may become developers
  - High impact libraries will increase size of potential user/developer base

Sources of Information

FUJITSU

- Four sources of information on typical library requirements for petascale applications have been used within OPL:
  - Hirata-sensei's presentation at the HPC in Asia workshop at ISC'11 on requirements for K computer nanotechnology applications (Japan)
  - Summary of requirements for K computer applications in the life sciences provided by Himeno-sensei (Japan)
  - IESP roadmap and workshops (USA)
  - Report by STFC, based on examination of DEISA and PRACE benchmark suite (Europe)

Core Algorithms

FUJITSU

- PDEs: equilibrium (implicit)
- PDEs: evolution (explicit)
- Fast Fourier Transforms (FFT)
- Fast Multipole Method (FMM)
- Particle pushing
- Sampling (multicanonical ensemble, etc.)
- Adaptive mesh refinement
- Sparse and dense linear algebra
- ODE integrators

Sources: Applications Breakout, IESP workshop, Maui, Hawaii, October 2010.
F. Hirata, HPC in Asia Workshop, ISC'11.

Minimal Library Dependencies(?)

FUJITSU

- MPI, GlobalArrays, GasNet
- ParMeTiS
- BLAS, ScaLAPACK
- UMFPACK, SuperLU, MUMPS
- PETSc, Hypre, Trilinos, SUNDIALS
- Chombo, SAMRAI
- FFTW
- GraphLib
- VisIt, VTK
- Triangle
- PALM
- SILO, ADIOS, HDF5
- BOOST

Source: Applications Breakout, IESP workshop, Maui, Hawaii, October 2010

Minimal Library Dependencies(?)

FUJITSU

- **MPI**, GlobalArrays, GasNet, SPHERE
- ParMeTiS
- **BLAS**, **ScaLAPACK**
- UMFPACK, SuperLU, MUMPS
- **PETSc**, Hypre, Trilinos, **SUNDIALS**
- Chombo, SAMRAI
- **FFTW**
- GraphLib
- VisIt, VTK
- Triangle
- PALM
- SILO, ADIOS, **HDF5**, NetCDF
- **BOOST**, GSL, R

(Repeated libraries in **bold**)

Source: K computer software list for ISLiM, 2011

Minimal Library Dependencies(?)

FUJITSU

- **MPI**, GlobalArrays, GasNet, SPHERE
- ParMeTiS
- **BLAS**, **ScaLAPACK**
- UMFPACK, SuperLU, MUMPS
- **PETSc**, Hypre, Trilinos, **SUNDIALS**
- Chombo, SAMRAI
- **FFTW**, FFTE
- GraphLib
- VisIt, VTK
- Triangle
- PALM
- SILO, ADIOS, **HDF5**, NetCDF
- **BOOST**, GSL, R

(Repeated libraries in **bold**)

Source: F. Hirata, HPC in Asia Workshop, ISC'11

Minimal Library Dependencies(?)

FUJITSU

- **MPI**, GlobalArrays, GasNet, SPHERE
- ParMeTiS
- **BLAS**, **ScaLAPACK**
- UMFPACK, SuperLU, MUMPS
- **PETSc**, **Hypre**, Trilinos, SUNDIALS, WSMP, SLEPSc
- Chombo, SAMRAI
- **FFTW**, FFTE
- GraphLib
- VisIt, VTK
- Triangle
- PALM
- SILO, ADIOS, HDF5, NetCDF, EAS3
- **BOOST**, **GSL**, R

(Repeated libraries in **bold**)

Source: STFC tech. report, August 2011

Summary

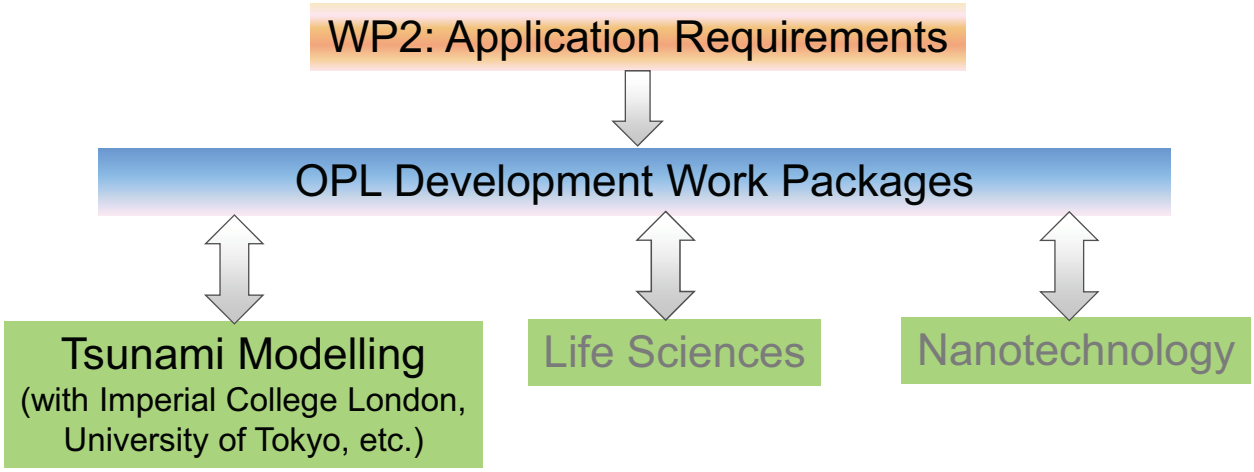
FUJITSU

- There is a large overlap between the library requirements identified by the three reports
- Libraries that are very widely used across all three regions include:
  - BLAS/ScaLAPACK
  - PETSc
  - FFTW
- So, these are natural targets for OPL
  - Or, alternatives with equivalent functionality where licensing of initial target is inappropriate

Discussion

FUJITSU

- Which life science and nanotechnology applications should be used as CDVs for the OPL numerical library development?



Application 'Co-Design Vehicles'

FUJITSU

shaping tomorrow with you

FUJITSU

OPL状況報告 (2011年10月)

- ライブラリ初版の公開
- SC11 BoFセッション

ライブラリ初版の公開（準備中）

FUJITSU

- 2DECOMP&FFT
  a library for 2D pencil decomposition and parallel fast Fourier transform
- FFTE
  a package to compute discrete Fourier transforms in 1, 2 and 3 dimensions
- spBLAS
  a threaded level 3 (matrix-matrix) sparse BLAS package
- PLASMA
  a dense linear algebra package optimized for multicore processors
- PETSc
  data structures and routines for the solution of partial differential equations

http://www.openpetascale.org/

SC11 BoFセッション

FUJITSU

- BoF (Birds of a Feather) セッション
  - セッション名：Open Petascale Libraries
  - 日時：11月16日(水) 17:30~19:00
  - 場所：TCC LL2

SC11 BoFのアジェンダ（仮）

FUJITSU

1. Gerard Gorman - Introduction, architectural trends, programming models (MPI, MPI+threads, PGAS), arguments for MPI+threads
2. Motoi Okuda - Architectures designed for MPI+threads (K computer and others)
3. Jack Dongarra - Dense linear algebra (PLASMA and DPLASMA)
4. Gerard Gorman - Sparse solvers and adaptive meshing (PETSc and PRAGMATIC)
5. Daisuke Takahashi - Fast Fourier transforms (FFTE)
6. Kenichi Miura - NRGs and Monte Carlo methods
7. Ross Nobes - Structure of OPL Project
8. Discussion

FUJITSU
shaping tomorrow with you

FFTEライブラリ適用事例（Quantum ESPRESSO）

富士通株式会社
計算科学ソリューション統括部
堀田 普介

Agenda

FUJITSU

- QE outline
- QE cost profile
- Application of FFTE to QE
- Performance comparison on PFX1
- Conclusions & Comments

What is Quantum ESPRESSO ?

FUJITSU

- An integrated suite of computer codes for electronic-structure calculations and materials modeling at the nanoscale
- Based on density-functional theory, plane waves, and pseudopotentials
- Standing for *opEn Source Package for Research in Electronic Structure, Simulation, and Optimization*
- Freely available to researchers around the world under the terms of the GPL (http://www.quantum-espresso.org/)
- There are two core solvers (PWscf and CP), and several packages and utilities
- PWscf was ported onto K (PFX1)

New version of PWscf (4.3.2)

FUJITSU

- Program codes (PWscf part only)

kind	# of files	# of lines
Fortran	475	296,176
C	12	29,473
Header	8	2,261
Total	495	327,910

- Parallelisation
  - Basically parallelized by MPI
  - More OpenMP codes than the previous version
- FFT
  - FFTW V1.2 is implemented and tuned by OpenMP
  - Could be linked with the external FFTW V3 instead of the internal FFTW
- Modification
  - Slightly tuned by directive (!ocl) effective for the hybrid run

Machine environment

FUJITSU

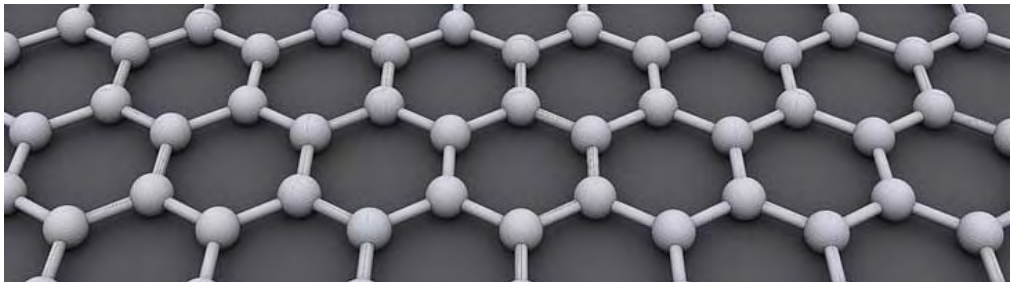
The machine used was as follows.

- PFX1 + Fujitsu compilers
  - Fujitsu's proprietary machine, each node has one SPARC64VIIIfx CPU (SPARC64VIIIfx is 8-core chip whose clock signal frequency is 2.0GHz)
  - PWscf runs in parallel with up to 84 nodes (672 cores)

Input data

FUJITSU

- Name : graphene
- Calculation : SCF
- # of atoms : 128
- # of electrons : 512
- Pseudopotential : C.pbe-rrkjus.UPF
- # of K points : 1
- FFT dimensions : (240,240,192), (150,150,120)



Cost profile

- Below are the results of profiler.
  (All processes and threads are accumulated)
  - FFT costs >16.5% (64 processes x 8 threads)

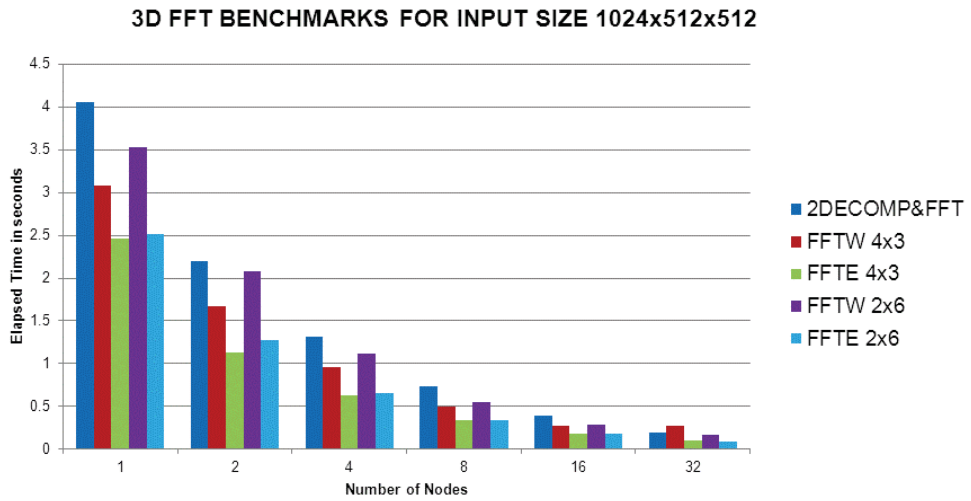
```
*******************************************************************************
    Application - procedures
*******************************************************************************
```

Cost	%	Barrier	%	Start	End	
3863547	100.0000	645500	16.7074	--	--	Application
664215	17.1918	0	0.0000	--	--	mca_btl_tofu_component_progress
394368	10.2074	366760	92.9994	315	347	fft_scalar.cft_2xy._OMP_3_
374985	9.7057	0	0.0000	116	166	dfunct.newq._OMP_1_
368763	9.5447	0	0.0000	--	--	__jwe_phb_sync
257502	6.6649	0	0.0000	--	--	ptlib_read_mrq
244872	6.3380	241316	98.5478	353	380	fft_scalar.cft_2xy._OMP_4_
219600	5.6839	0	0.0000	--	--	opal_progress
145565	3.7677	0	0.0000	--	--	dl_gmwn1_base_16x4_
139775	3.6178	0	0.0000	112	117	addusdens_g._PRL_3_
111002	2.8731	0	0.0000	--	--	dl_gmwn1_general_16x4_

FFTE performance (FYI)

Multi-Dimensional FFT

Performance Benchmarks on BX900 with Open MPI and Fujitsu Compilers



Note: (a) 4x3 refers to 4 tasks and 3 threads (b) 2x6 refers to 2 tasks and 6 threads

Comparison of FFTs (FYI)

FUJITSU

	FFTE	FFTW
Language	Fortran	C with Fortran Interface
Parallelization	OpenMP + MPI	OpenMP + MPI
Input/Output	complex to complex	complex to complex real to complex complex to real
Radix	2,3,5 (only $2^p3^q5^r$ are allowed) (*)	arbitrary
Dimension	1D, 2D, 3D	1D to nD
Precision	double	single and double
# of plans	1	n
Normalization	yes (1/N for inverse)	no
Scaling factor	no	no

(*) PWscf looks satisfying this condition fortunately.

Application of FFTE to QE (1)

FUJITSU

- # of plans
  - FFTE can deal with only one plan (*)
  - PWscf needs at least 9 plans (3 plans for x,y,z axes)

⇨ Made 9 copies
(e.g. `zfft1d_x1`, `fft235_y1` etc).

(*) transform matrix for FFT

Application of FFTE to QE (2)

FUJITSU

- Normalization factor
  - FFTE has normalization factors 1 for forward transform and $1/N$ for inverse transform, where N is # of points
  - FFTW doesn't have normalization factor (i.e., $F^{-1} \circ F = F \circ F^{-1} = N$)
  - PWscf claims $1/N$ for forward and 1 for inverse

⇨ FFTE needed modifying.

cf. In some style, (including FFTE)

Forward

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi \frac{k}{N} n} \quad (k = 0, \ldots, N-1)$$

Inverse

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \cdot b e^{+i2\pi \frac{k}{N} n} \quad (n = 0, \ldots, N-1)$$

Coefficient ($1/N$ in this style) and sign (- for forward and + for inverse in this style) depend on style of definition

Application of FFTE to QE (3)

FUJITSU

- OpenMP overhead
  - As FFTE is highly customized for FFT with large # of points, even loops composed of 1 line are parallelized by OpenMP
  - Loop length (150 – 240 in case of graphene) is too small

⇨ Cancelled OpenMP directives for loops composed of 1 line.

Proposal to FFTE developer

FUJITSU

Through the application, I thought some features need implementing, which make porting programs easier.

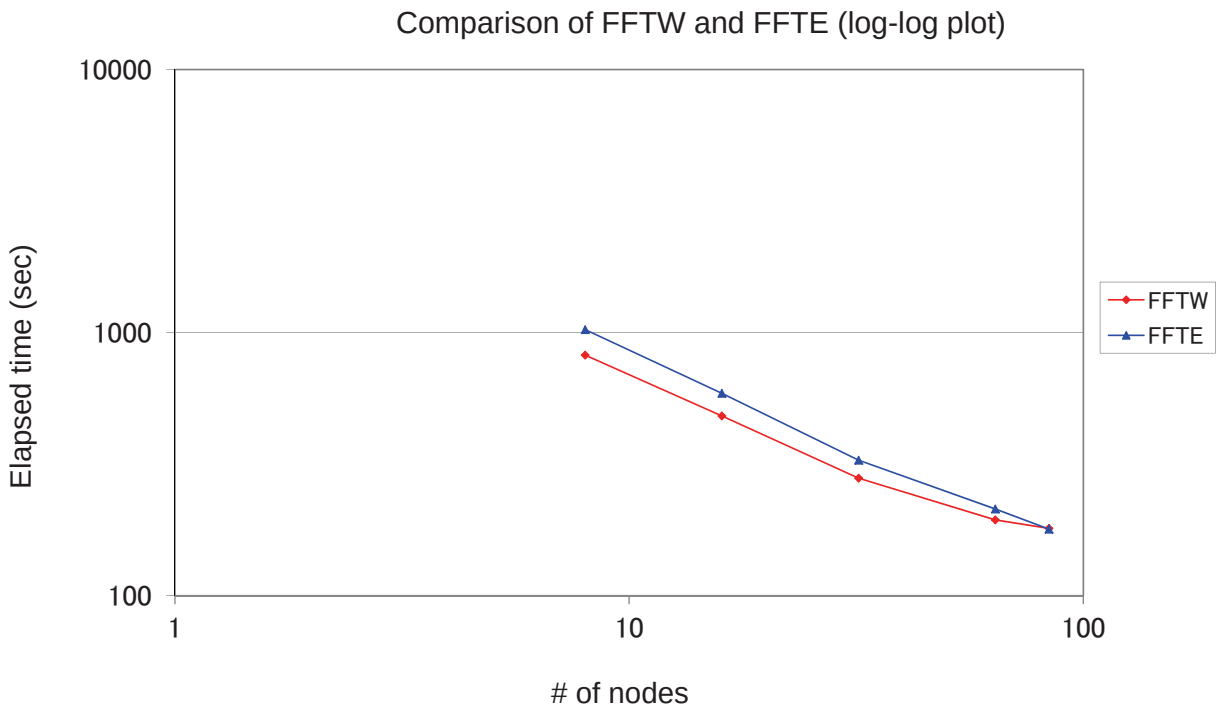
- Support of multiple plans
- Support of scaling factor instead of normalization
- Support of better performance even with small # of points

Reply from Dr. Takahashi :
The requests you suggested seems to contain important ideas for FFTE.
I will consider supporting your requests in a future release of FFTE.

Results of FFTE

FUJITSU

Using tuned version with graphene input data, comparison up to 84 nodes (672 cores on PFX1).



Conclusions & Comments

- Replacement of FFTW by FFTE was done, but with small data such as graphene, the less effect was seen than expected while FFTE is customized for PFX1.

- FFTE is easy-to-use and simple FFT library, but less applicable for general applications.

FUJITSU

shaping tomorrow with you

FUJITSU
shaping tomorrow with you

大規模数値計算における数学ライブラリの取り組みについて

富士通株式会社 ミドルウェア事業本部
アプリMW事業部 第四開発部 数学ライブラリGroup

2012.3.1 SS研ペタスケールLib-WG

FUJITSU

報告内容

- 当社数学ライブラリ紹介
  - 富士通数学ライブラリ体系
  - SSL II系ライブラリ開発の経緯について
  - 現在の取り組み
  - 方針について
- 大規模数値計算向けOSSライブラリ概観
- まとめ

紹介(1):富士通数学ライブラリ体系

FUJITSU

- 逐次版ライブラリ(スレッドセーフ)
  - SSL II …約300種ルーチン。Fortranで記述。単・倍精度。
  - C-SSL II …SSL IIへのCインタフェース。主に倍精度。
  - BLAS/LAPACK …米国で開発されNetlibで公開されているデファクトスタンダードな線形計算ライブラリ。BLAS約80種、LAPACK約400種ルーチン。各精度あわせて約2000ルーチン。
- スレッド並列版ライブラリ
  - SSL IIスレッド並列機能 …重要機能約80ルーチン。混在使用できるように逐次SSL IIと別インタフェース。倍精度。
  - C-SSL IIスレッド並列機能 …SSL IIスレッド並列機能へのCインタフェース。
  - BLAS/LAPACK …逐次向けと同一インタフェース。LAPACK層でスレッド並列化したルーチンもあり。
- MPI並列版ライブラリ
  - SSL II/MPI …3次元FFT機能3ルーチン。
  - ScaLAPACK …PBLASとBLACSレイヤーを含む構成。約200種ルーチン、各精度あわせて約700ルーチン。

紹介(2):SSL II系ライブラリについて

FUJITSU

■機能一覧

線形計算	:連立一次方程式、逆行列、最小二乗解、特異値分解
固有値問題	:固有値・固有ベクトル
非線形方程式	:代数方程式、超越方程式、連立非線形方程式
極値問題	:関数の極小化、線形計画問題、非線形計画問題
補間・近似	:補間式/補間値、近似式、平滑化式/平滑値、級数展開
変換	:フーリエ変換、ラプラス変換、ウェーブレット変換
数値微分・積分	:離散点/関数入力、有限区間/無限領域、一次元/二次元
微分方程式	:連立一階常微分方程式、連立一階スティフ常微分方程式
特殊関数	:ベッセル関数、楕円積分、指数積分、正弦・余弦積分
擬似乱数	:乱数生成(一様/正規/指数/ポアソン/二項)、乱数検定

■開発の歴史

■SSL II 基本機能:
国内の大学・研究機関の専門家に協力いただいて共同開発。(1970年代～1980年代)

■SSL II/VP拡張機能, SSL II/VPP:
ANU(オーストラリア国立大学)と共同開発 (1990年代)

■C-SSLII/VP:
SSL II/VPへのCインタフェース(1997～, FECIT)

■BLAS/VP, LAPACK/VP, ScaLAPACK
ベクトルマシン向け

SMPマシンへのプラットフォーム移行(1998年頃)

■SSL II 逐次機能:
スレッドセーフ化、スカラ機向けチューニング

■SSL IIスレッド並列機能:
重要機能約80ルーチン

■SSL II/XPF, SSL II/MPI:
分散メモリ並列向け

■BLAS, LAPACK, ScaLAPACK
スカラマシン向けチューニング版、スレッド並列化

紹介(3):分散メモリ型数学ライブラリ開発経緯

FUJITSU

- SSL II/VPP （1992～1999年）
  - VPP FORTRANのサブルーチンとして記述。
  - 1999年時点で52ルーチン。
  - ANU(Australian National University)と共同開発。
  - 機能範囲は密行列問題・スパース行列反復解法・FFT・乱数生成など。
  - 基本的には配列の1つの次元を等分割するデータ分散方式が多い。
- SSL II/HPF （2000年頃）
  - SSL II/VPPをベースにインターフェース部分を開発。24ルーチン。
- SSL II/XPF （2002年～）
  - SSL II/VPPと同じインターフェース、同機能範囲で提供。
  - 並行して、重要技術はSSL IIスレッド並列機能に移行。
- SSL II/MPI （2005年～）
  - 現時点では実・複素の3次元FFT(1軸分散)の2ルーチンのみ。
  - 次版でVolumetric FFT(3軸共に分割してデータ分散)を追加予定。
  - ノード内スレッド並列のHybrid並列。
- ScaLAPACK
  - 今後も継続的にサポートしていく方針。

現在の取り組み - 概要 -

FUJITSU

- SSL II スレッド並列機能ライブラリの機能エンハンス
  - スパース行列連立一次方程式の直接解法(正値対称行列, Left-looking)
  - スパース行列連立一次方程式の反復解法(IDR法,COCR法,近似逆行列前処理)
  - スパース行列の固有値問題解法(Jacobi-Davidson法)
  - 常微分方程式解法(RADAU5)
- SSL II/MPIライブラリの機能エンハンス
  - 分散並列3次元FFTルーチン開発(Volumetric decomposition)
- マシンアーキ毎のチューニング
  - 命令列スケジューリングの改善・浮動小数点レジスタの有効利用
  - SIMD命令の利用
    - アセンブラコーディング(逐次BLAS重要ルーチン)
    - SIMD命令生成組み込み関数利用(FFT系)
    - コンパイラによるSIMD化促進
  - prefetchのチューニング
  - キャッシュ有効利用
  - スレッド並列化（OpenMP記述）

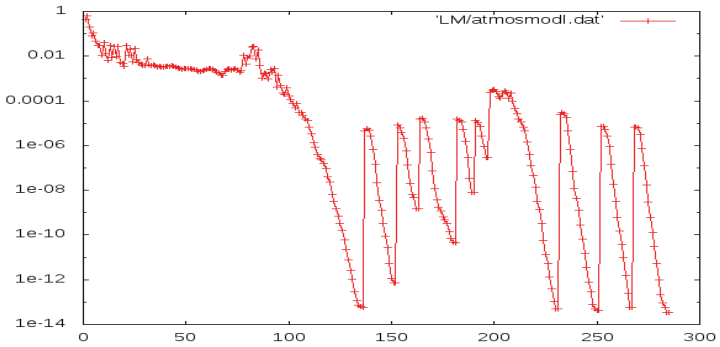
現在の取り組み(1) - スパース行列の固有値問題 -

FUJITSU

■Jacobi-Davidson法によるスパース行列固有値問題解法ルーチンを新規開発(エルミートおよび非対称)

- ■アルゴリズムが複雑で、OSSでは試験的実装に留まっているものが多いことから開発必要性有と判断。
- ■内部反復として連立一次方程式反復解法の数ステップを使用する(GMRES,BiCGStab(L)に対応)。
- ■Compressed Row Storage格納方式に対応。固有値選択はVal指定,LM,SM,LR,SR,LI,SIに対応。

- 測定マシン:
  - CPU : SandyBridge 2.7Ghz
  - 2CPU×8core で16並列
- 設定パラメタ条件:
  - 絶対値最大固有値を10個まで
  - 反復回数上限300回
  - 内部反復回数上限30回,GMRES使用
  - 副空間最大次元:80
  - リスタート時の縮小次元:50



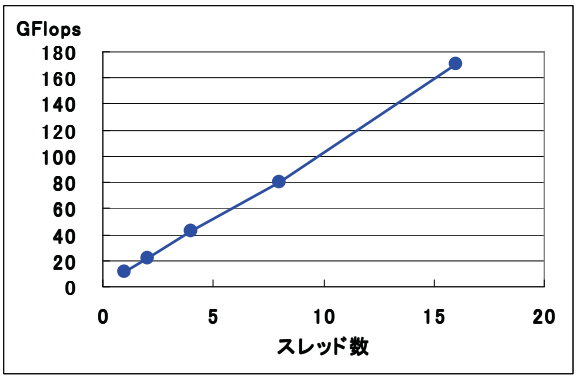
MatrixName	Type	Dim.	nonzero	Iter.	num	LM eigen	rel.err	Time
3dspectralwave	エルミート meterials	680943	17165766 (下三角部)	169	10	69.06480786603274	4.0E-14	444.30
fem_hifreq_circuit	複素対称 electromag.	491100	20239237	121	10	(24542.9075344564, -1.6043623581E-12)	8.8E-09	109.35
tmt_unsym	非対称実 electromag.	917825	4584801	228	10	(7.99999998833645, -3.2237076408E-10)	3.2E-07	493.91
Hamrle3	非対称実 circuit sim.	1447360	5514242	300	8	(-1.3574077330799, -1.7110687085144)	1.0E-13	848.59
atmosmodl	非対称実 fluid dynamics	1489752	10319760	285	10	(-620445.25170863, -3.9563019527E-11)	5.9E-14	992.30 second

現在の取り組み(3) - LAPACKスレッド並列 -

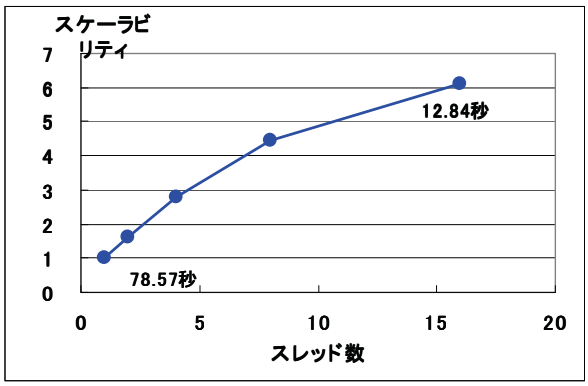
FUJITSU

- 計測マシン
  - ■FX10 (SPARC64IXfx 1.848GHz 16コア)

- DGESV
  - LU分解による連立一次方程式
  - 元数(N): 20,000
  - LAPACK層で並列化



- DSYEVD
  - 対称行列の固有値問題
  - 元数(N): 5,000
  - 全固有値・全固有ベクトル求解
  - スレッド並列化効果を出しにくい問題



富士通 数学ライブラリ方針

FUJITSU

- 高速化：継続的に取り組み
  - 命令レベルチューニング(命令スケジューリング改善, SIMD化, prefetch, …)。
  - キャッシュ有効利用(ブロック化)。
  - スレッド並列化チューニング。
- 機能的エンハンス：スレッド並列機能を重点的に
  - 顧客やフィールドからの要望に対して、優先順位を決めて対応。
  - 技術動向調査に基づき計画
    - 技術進展により陳腐化した既存ルーチン機能があれば刷新。
    - 問題大規模化により特有のアルゴリズムが必要になれば取り込み。
- プロセス並列版：状況ウォッチ
  - データ分散方法が共通でなくライブラリ化が難しい。
  - 経験的に、独自インターフェースでのプロセス並列機能はアプリから許容されない。
  - 多数ユーザを獲得した標準的新仕様が現れた場合には追従していく。

FUJITSU

報告内容

- 当社数学ライブラリ紹介
- 大規模数値計算向けOSSライブラリ概観
  - 大規模密行列問題
  - 分散並列FFT
  - スパース行列系の問題
- まとめ

OSS調査(1) -大規模密行列問題-

FUJITSU

- ScaLAPACK
  - LAPACKの分散メモリ並列版ライブラリ。
  - 分散並列の密行列問題（連立一次方程式・最小二乗解・固有値問題・特異値問題など）のソルバとしてデファクトスタンダード。
  - 行列を二次元サイクリック分割するデータ分散方式。
  - HPFインタフェースのプロトタイプ版も存在した。
  - LAPACKのMany-core版として開発中のPLASMAに対応するものとして、DPLASMAも開発進行中。

  - 分散メモリ型の新並列言語が普及するためには、wrapperを介すなどしてScaLAPACKを利用できるのが望ましいと思われる。
    - XcalableMPではMPI並列ライブラリとのインタフェース検討中？
    - CrayはCo-array versionのScaLAPACKを計画中？

OSS調査(2) -分散並列FFT-

FUJITSU

- FFTW
  - 2011.7末の最新版(V3.3)でMPI並列機能も加わったが、1軸分散相当であり、大規模での並列効果は期待できない。
  - 自動チューニング機構を持っていることが特徴的。
  - 普及する傾向だが、一方で内部挙動が複雑で扱いにくい面もある。
  - FX10向けには、SPARC64IXfxのSIMD命令を利用するための修正方法準備。
- FFTE
  - 非常に長いFFT長でのMPI並列1次元FFTルーチンを持っていることが特徴。
  - MPI並列1次元FFTはHPCCのGlobal-FFTで用いられている。
  - 汎用性には疑問もあり（使用ノード数とFFT長さに関する制約が強いなど）。
  - 大規模システムでの多次元FFTに関する機能や性能は未調査だが期待薄。
- FFTSS
  - FFT長さは2冪のみで実FFTは無。MPI並列は2次元FFTのみ。
  - CREST成果物。その後の活動状況不明。
- 2DECOMP&FFT
  - 3次元FFTの2軸分散（柱状データ分割）。

※ 1次元FFTカーネル処理だけライブラリを利用して、MPI部分はユーザーが自作してしまうことも多いもよう。

OSS調査(3) -スパース行列系ライブラリ-

FUJITSU

- 連立一次方程式反復解法など
  - PETSc …複素数対応。直接解法なども含。object指向。
    行列成分を格納しないMatrix-free方式も可能。
  - Trilinos …各種package集合。object指向。
  - Hypre …マルチグリッド前処理。実行列のみ。
  - Lis …内部的にDD方式4倍精度演算有。CREST成果物。
- 連立一次方程式直接解法
  - SuperLU(_DIST)…CSR/CSC格納方式に対応。少しobject指向的。
  - UMFPACK …マルチフロンタル法。CSC格納方式(転置も可)。MPI並列無。
  - MUMPS …マルチフロンタル法。オーダリング含（AMD,QAMD,AMF）。
  - (PARDISO,WSMP …非OSS)

  → スパース行列を圧縮格納方式で渡す（データ分散方式は未調査）。
- 固有値問題
  - (P)ARPACK
    - RCI(Reverse Communication Interface)を用いており、行列ベクトル積はユーザー側の処理にゆだねられている。明に行列成分を格納する必要がない点で汎用的であるが、Modern Programming Modelとは言い難い。
  - 他多数(JDQZ,LOBPCG,BLZPACK,JADAMILU,Anasazi,SLEPc,…)
    - 大規模計算に耐えない試験的実装にとどまっているものが多い印象。

まとめ：問題タイプ分類案と分散並列ライブラリ状況

FUJITSU

- 大規模密行列タイプ

  → ScaLAPACKの利用が現実的。
- 空間領域単純分割タイプ
  - 3次元FFT機能

    → 使いやすいOSSライブラリが整備されているとは言えない状況。
    SSL II/MPI次版で3次元FFT機能の新ルーチンを追加。
  - 例えば2DECOMP（general-purpose 2D pencil decomposition library）
- 複雑分割およびスパース行列タイプ

  → 混迷状態。新しいProgramming Modelを利用した統一的指針が必要か。
- その他
  - 乱数生成
  - 行列成分オーダリング(ParMETIS)
  - グラフ分割
  - メッシュ生成（PRAgMaTIc）
  - 積分計算の分散処理（SS法とか）
  - パラメタスイープ
  - 連成計算
  - Common Data Form(NetCDF,HDF5)
  - 可視化
  - …

FUJITSU

shaping tomorrow with you

FUJITSU
shaping tomorrow with you

Open Petascale Libraries: Current Status

Technical Computing Research Division
Fujitsu Laboratories of Europe

20 July 2012

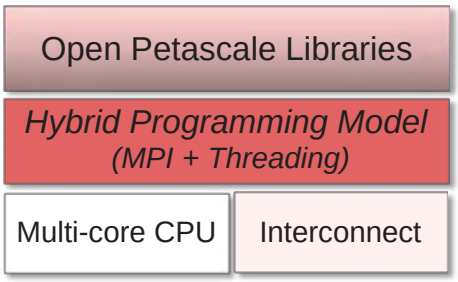
FLE-TCR-PSL-12-010

Open Petascale Libraries

FUJITSU

Global collaboration to develop advanced numerical software for supercomputing.

- Dedicated forum to promote the open exchange of ideas and the collaborative development of general-purpose and application-specific numerical libraries.
- Targeted initially at parallel computers built from multi-core processors.



openpetascale LIBRARIES

OPL Members

FUJITSU

Open Petascale Libraries

FUJITSU

- Six currently supported libraries:
  - 2DECOMP&FFT, version 1.4.682 (original developers: NAG Ltd).
  - FFTE, version 5.0 (University of Tsukuba).
  - spBLAS, version 1.01.672 (Fujitsu Laboratories of Europe).
  - PLASMA, version 2.4.5 (ICL, University of Tennessee).
  - PETSc, version 3.3-p1 (Argonne National Laboratory).
  - PRAgMaTIc, version 1.0.2 (Imperial College London).
- www.openpetascale.org/index.php/public/page/download.
- Related libraries with no current OPL release:
  - OpenMP-enabled version of PETSc under development within OPL.
  - DPLASMA: distributed memory version of PLASMA.
  - Plan for future OPL Random Number Generator (RNG) library.

Status of Each Supported Library

- Features
- Status of FX10 Porting and Optimisation

2DECOMP&FFT 1.4.682

- 2D pencil decomposition and parallel fast Fourier transforms.

●		2D pencil decomposition FFTs.
●		3D FFTs.
●		Arbitrary input sizes.
●		Complex-to-complex transforms.
●		Complex-to-real transforms.
●		Real-to-complex transforms.
●		Interface with external FFT libraries.

Legend :
● : Ported onto FX10.
▲ : Optimised for FX10.
No marks : There are features with the original version.
Grey coloured : Under development (for the original version).

FFTE 5.0

FUJITSU

- A package to compute discrete Fourier transforms in 1, 2 and 3 dimensions.
  - K computer optimisation of FFTE carried out by University of Tsukuba as part of work towards 2011 HPC Challenge Benchmarks.

●	▲	1D, 2D and 3D FFTs.
●	▲	Inputs of sizes $2^p3^q5^r$.
●	▲	Complex-to-complex transforms.
●	▲	Complex-to-real transforms.
●	▲	Hybrid parallelism using OpenMP and MPI.

Legend :
● : Ported onto FX10.
▲ : Optimised for FX10.
No marks : There are features with the original version.
Grey coloured : Under development (for the original version).

spBLAS 1.01

FUJITSU

- A threaded level 3 (matrix-matrix) sparse BLAS package.

●		Compressed Sparse Row Matrices.
●		Compressed Sparse Column Matrices.
●		Coordinate Format Matrices.
●		Block Compressed Sparse Row Matrices.

Legend :
● : Ported onto FX10.
▲ : Optimised for FX10.
No marks : There are features with the original version.
Grey coloured : Under development (for the original version).

PLASMA 2.4.5

FUJITSU

- A dense linear algebra package optimized for multi-core processors.

●		Symmetric Positive Definite Systems of Linear Equations.
●		Cholesky factorization (PLASMA_dpotrf).
●		General Systems of Linear Equations.
●		LU factorization (PLASMA_dgetrf).
●		QR factorization (PLASMA_dgeqrf).
●		Standard and Generalized Dense Symmetric Eigenproblems.
●		Compute eigenvalues (PLASMA_dsyev).
		Compute eigenvectors (research topic for developers).
●		Dense Singular Value Decomposition.
●		Compute singular values (PLASMA_dgesvd).
		Compute singular vectors (research topic for developers).
●		Tiled BLAS Level 3.

Legend :
● : Ported onto FX10.
▲ : Optimised for FX10.
No marks : Features exist in original version (not ported).
Grey text : Under development (in the original version).

PETSc 3.3-p1

FUJITSU

- Data structures and routines for the solution of partial differential equations.

●		Vector data structure.
●		Matrix data structures.
●		Dense.
●		Compressed sparse row (CSR).
●		Block CSR.
●		Symmetric block CSR.
●		Preconditioners.
●		Krylov subspace methods (sparse iterative solvers).
●		Newton-based nonlinear solvers.
●		ODE solvers.

Legend :
● : Ported onto FX10.
▲ : Optimised for FX10.
No marks : Features exist in original version (not ported).
Grey text : Under development (in the original version).

PRAgMaTIc

- 2D and 3D anisotropic mesh adaptivity for meshes of simplexes.
  - Currently not supported on PRIMEHPC FX10.
  - Port scheduled for completion by end of September.

		Distributed mesh data structure.
		Mesh operations.
		Mesh smooth.
		Mesh refine.
		Mesh coarsen.
		Face/edge swapping.

Legend :
● : Ported onto FX10.
▲ : Optimised for FX10.
No marks : There are features with the original version.
Grey coloured : Under development (for the original version).

Status of Related Libraries

- Features
- Status of FX10 Porting and Optimisation

PETSc with OpenMP

FUJITSU

- Extend PETSc by adding OpenMP support (within OPL).
  - Code is currently not ready for public release.

●		Vector data structure.
●		Matrix data structures.
●		Dense.
●		Compressed sparse row (CSR).
		Block CSR.
		Symmetric block CSR.
●		Preconditioners.
●		Krylov subspace methods (sparse iterative solvers).
		Newton-based nonlinear solvers.
		ODE solvers.

Legend :
● : Ported onto FX10.
▲ : Optimised for FX10.
No marks : Features exist in original version (not ported).
Grey text : Under development (in the original version).

Distributed PLASMA

FUJITSU

- Initial release candidate of DAGuE/DPLASMA available since December 2011 from University of Tennessee.
  - DPLASMA is the multi-node version of PLASMA.
  - DAGuE is the framework that distributes the DAG scheduler over multiple nodes.
  - OPL development (e.g. FX10 port) awaiting a full release.

		Cholesky (factorization and solve).
		QR and LQ (factorization and generation of Q).
		LU (factorization and solve).
		Matrix-matrix and triangular solve operations.

Legend :
● : Ported onto FX10.
▲ : Optimised for FX10.
No marks : Features exist in original version (not ported).
Grey text : Under development (in the original version).

PRIMEHPC FX10 Optimisation

- PLASMA
- PETSc

PLASMA Optimisation

- Targets for PRIMEHPC FX10 optimisation:
  - Cholesky factorisation.
  - LU and QR factorisation.
  - Computation of eigenvalues and singular values.
- These have been highlighted by the developers as:
  - Important functionality.
  - Benefiting from the Directed Acyclic Graph approach (instead of fork-join OpenMP).
- Benchmark performance of optimised version against:
  - PLASMA with no optimisations.
  - Equivalent SSL II routines.

PETSc Optimisation

FUJITSU

- Optimisation of PETSc is focused on the OpenMP version of the code.
  - Currently unreleased.
  - Identical to non-OpenMP version if “-Kopenmp” is not specified.
  - Uses SSL II for low-level BLAS routines.
- Initial experiments with using a threaded version of spBLAS to achieve a hybrid version of PETSc were carried out.
  - Good scaling (at least for a small number of threads), but introduced a performance overhead when running with a single thread.
  - PETSc developers at Argonne National Laboratory advised against this strategy and recommended adding OpenMP to the PETSc source code instead.

PETSc Optimisation

FUJITSU

- Four initial target matrices for optimisation (all from CFD):
  - Backward-facing step (turbulent fluid flow).
  - Lock exchange (mixing of two fluids with different densities).
  - Salt fingering (mixing of two fluids with different densities).
  - Flue pressure (dispersion of a momentum driven flue emission in a cross wind under neutral atmospheric conditions).
- Between 4 million and 750 million nonzero entries.
- Benchmark performance of optimised version against:
  - Standard version of PETSc (no OpenMP, no optimisations).
  - Initial OpenMP version of PETSc (no optimisations).
  - Pthreads version of PETSc (available from PETSc developers).

Future Plans

Future Plans

- By 30 September 2012:
  - Release initial OpenMP-enabled version of PETSc.
  - Complete PRIMEHPC FX10 optimisations for PETSc and PLASMA.
  - Complete port of PRAgMaTIc to PRIMEHPC FX10.
- By 31 March 2013:
  - Extend functionality of OpenMP-enabled version of PETSc (e.g. to include block CSR matrices).
  - Identify and provide a supported release for a RNG library.
  - PRIMEHPC FX10 optimisations for PRAgMaTIc and spBLAS.
  - Consider inclusion of supported DPLASMA release in OPL.

FUJITSU

shaping tomorrow with you

FUJITSU
shaping tomorrow with you

Application of PLASMA to OpenFMO code

James Southern, Paul Caheny, Michael Li
Fujitsu Laboratories of Europe

20 February 2012

FLE-TCR-PSL-13-004

Overview

FUJITSU

- **Aim:** Investigate the use of PLASMA in the OpenFMO application.
  - **OpenFMO** is a fragment molecular orbital (FMO) software package for computational chemistry.
  - The FMO method requires **many eigenvector calculations**.
  - Typically these calculations use LAPACK.
  - **PLASMA** is the OPL library identified as a replacement for LAPACK on multi-core systems.
  - What is the prospect for PLASMA to improve OpenFMO performance compared to vendor LAPACK?
    - On PRIMEHPC FX10 (compared to Fujitsu SSL II).
    - On PRIMERGY cluster (compared to Intel MKL).

OpenFMO

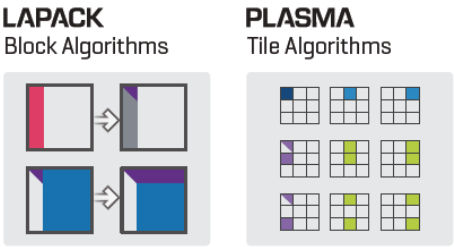
FUJITSU

- Kyushu University's implementation of the FMO method.
- The FMO method decomposes the molecule to be computed into many small fragments.
  - Quantum mechanical calculations on fragments and their dimers.
  - Coulomb field calculation for entire system.
  - Leads to **highly efficient parallelization** as calculation for individual fragments can be farmed out to processors.
    - Each process constructs a relatively **small matrix** for its fragment and then **computes** its **eigenvectors**.
    - Global broadcast of new energy field at end of iteration.
  - A larger molecule or more basis functions means more small fragments to farm out.
    - **Matrix size for each fragment remains small**.

PLASMA

FUJITSU

- PLASMA is a linear algebra package aimed for use on multi-threaded CPUs.
  - Replaces LAPACK's block algorithms with **tile algorithms** more suitable for fine-grained parallelism.



  - Uses Directed Acyclic Graph (DAG) scheduling rather than fork-join parallelism to minimize idle time.
  - These methods provide significant benefits for **large matrices** (lots of work) on a **large number of cores** (when the overhead of fork-join becomes very large).
  - For **small problems** tile algorithms and DAG scheduling do not perform so well → PLASMA is not designed for these problems.

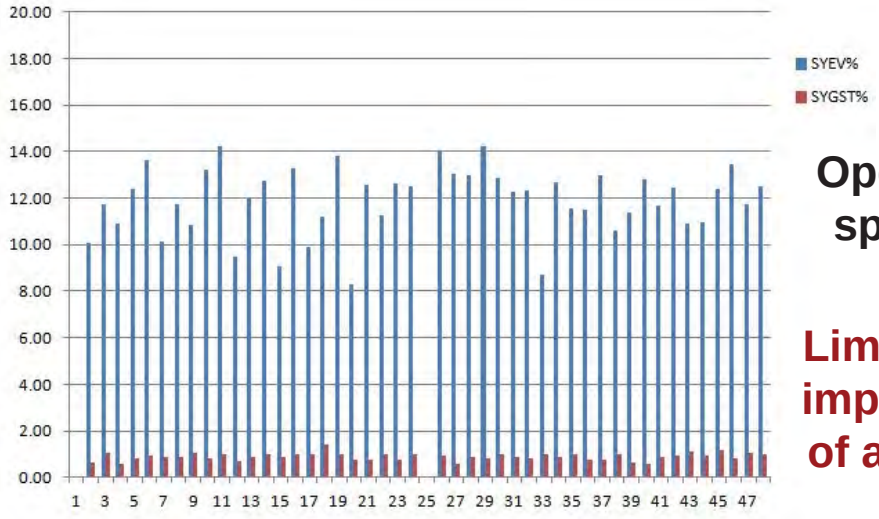
Small matrix sizes in OpenFMO mean that prospects for improving its performance using PLASMA are not good

OpenFMO Update

- Runtime and compilation problems for PRIMEHPC FX10 reported at last teleconference resolved.
  - Compiling with –Xg resolves SIGBUS in ofmo_input()
  - New openfmo tar received Feb 5th resolves problem compiling without –DDEBUG.
  - OpenFMO profiled with SSL2.
  - Small proportion of runtime spent in LAPACK functionality and matrix sizes are small.
  - Issues integrating PLASMA with OpenMP threaded application codes on PRIMEHPC FX10 → PLASMA much slower than SSL II.
- OpenFMO also run on PRIMERGY cluster.
  - No issues integrating PLASMA with OpenMP application.
  - PLASMA performance with OpenFMO is much better than on PRIMEHPC FX10 (but still not as good as MKL).

OpenFMO Profile (FX10)

- Profiled OpenFMO with "Small" problem data set (1ejg-sto3) running on 48 nodes of Varuna GE (48 MPI x 16 OpenMP) with single threaded SSL2 (runtime 5m15s).



<15% of total OpenFMO run-time spent in LAPACK routines

Limited prospect for improvement by use of any alternative to LAPACK

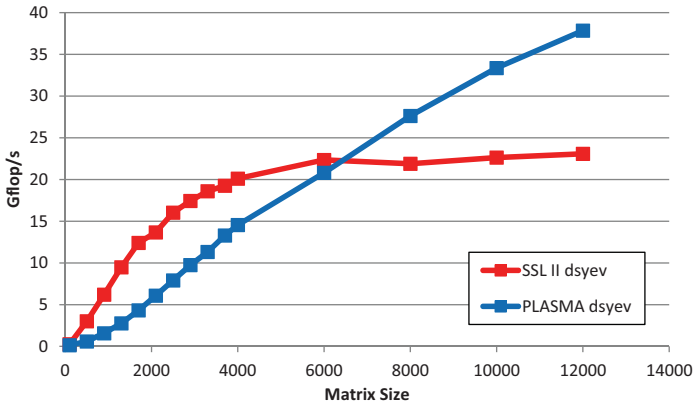
Percentage of Thread 0 runtime spent in SYEV & SYGST per process.

OpenFMO and Matrix Size

- SYGST & SYEV not themselves threaded in SSLBLAMP but make use of multi-threaded BLAS.
  - Impact of linking with SSL2BLAMP instead of SSL2 minimal ~ 1% in overall application runtime.
  - Possibly due to **small matrix sizes** in calls to SYEV & SYGST.
- Average N in SYEV calls from OpenFMO is small. (Ave N ~ 100 – 400 depending on input data set. More detail in following slides.)

PLASMA needs much **larger** problem sizes to beat SSL II (due to overhead of DAGs and tile algorithms).

On SPARC64 IXfx, PLASMA requires a **matrix size over 6000** before it beats SSL II.



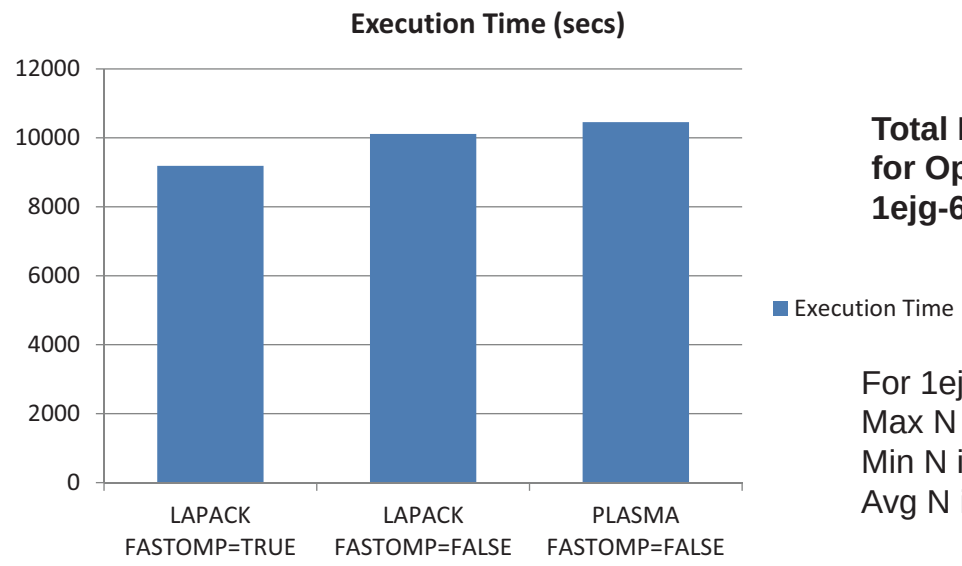
PLASMA and OpenMP

- PLASMA must be linked with single threaded BLAS/LAPACK and have OMP_NUM_THREADS set to 1 when initialised and called by application code.
  - This is a problem for multi-threaded interoperability with OpenMP application code.
  - Current solution (from the PLASMA forum) is:

```
omp_set_num_threads(1) //set OpenMP number of threads to 1
PLASMA_init(16)        //set PLASMA number of threads to 16
PLASMA_dsyev(....)     // call PLASMA functionality
PLASMA_Finalize()      // Finalize PLASMA
omp_set_num_threads(16)// Set OpenMP number of threads back to 16 for
                       // application code that follows.
```

- This workaround does not have a negative impact on performance on PRIMERGY cluster.
- On FX10, using omp_set_num_threads requires setting FLIB_FASTOMP=FALSE.
  - Negative impact of FLIB_FASTOMP=FALSE is significant on performance of OpenFMO application code.

OpenFMO on FX10



Total Execution time for OpenFMO with 1ejg-631s data set.

For 1ejg-631s data set
Max N in SYEV = 680,
Min N in SYEV = 81
Avg N in SYEV = 337

PRIMEHPC FX10, 8 MPI x 16 OpenMP Threads, 1ejg-631s data set.

- Running OpenFMO with PLASMA is **1.13x slower** than using SSL II.
- Largely due to setting FLIB_FASTOMP=FALSE.

OpenFMO on PRIMERGY

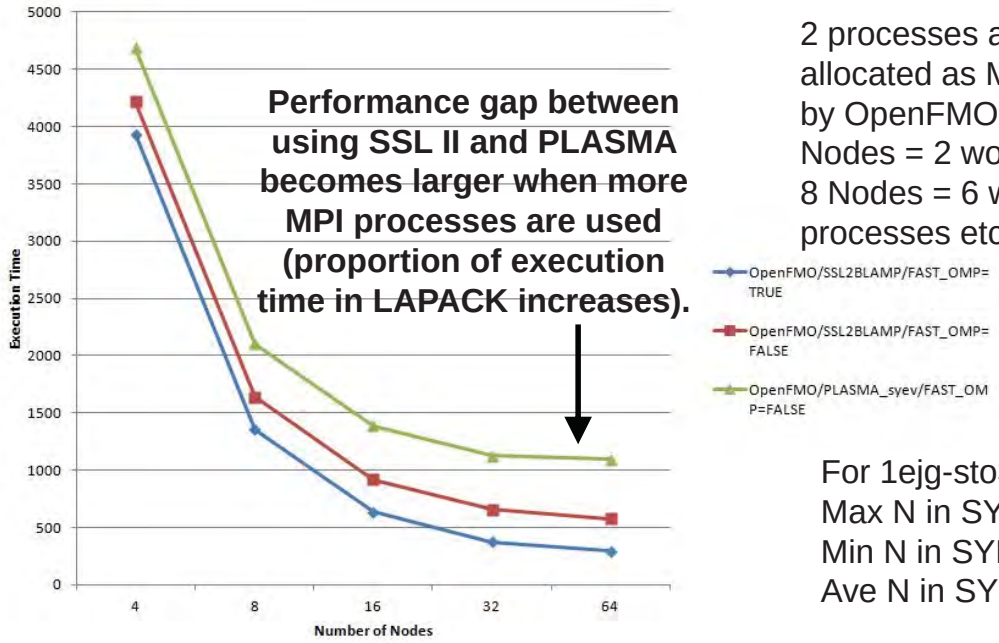
Total Execution time for OpenFMO with 1ejg-631s data set.

For 1ejg-631s data set
Max N in SYEV = 680,
Min N in SYEV = 81
Avg N in SYEV = 337

PRIMERGY BX900 Cluster, 8 MPI x 12 OpenMP Threads, 1ejg-631s data set.

- Running OpenFMO with PLASMA is **1.03x slower** than using Intel MKL.
- Same as on PRIMEHPC FX10 when FLIB_FASTOMP=FALSE for SSL II.

Scaling



2 processes are always allocated as Master and I/O by OpenFMO. Therefore 4 Nodes = 2 worker processes, 8 Nodes = 6 worker processes etc.

For 1ejg-sto3 data set
Max N in SYEV = 240,
Min N in SYEV = 28
Ave N in SYEV = 123.4

Total Execution time for OpenFMO with 1ejg-sto3 data set, 16 threads per node on PRIMEHPC FX10.

Summary

- The use of PLASMA as an alternative to SSL II (or Intel MKL) LAPACK within OpenFMO has been investigated on PRIMEHPC FX10 and PRIMERGY cluster.
  - The **FMO method is not well suited for use with PLASMA**: the size of the **matrices** are **too small** to justify the overheads incurred when using tile algorithms and DAGs.
    - On PRIMEHPC FX10, 1.13x slowdown when using PLASMA (1.10x is due to setting FLIB_FASTOMP=FALSE).
    - On PRIMERGY, FLIB_FASTOMP is not available, so slowdown is less (1.03x) → comparable to PRIMEHPC FX10.
  - Little prospect of overcoming these issues for an FMO code.
  - For a standard quantum chemistry code (that does not use the FMO method) the size of the **matrices** will be **much larger**: PLASMA is much more likely to work well with this type of application.

FUJITSU

shaping tomorrow with you