### 2.12 FMO プログラム ABINIT-MP の富士通系マシンでの性能評価

立教大学 理学部 化学科 望月祐志

【序】 巨大分子の"分割統合"系の種々の量子化学計算法[1]の中で、フラグメント分子軌道 (FMO)法[2-4]は方法論的な発展の多様性と応用事例の豊かさの点で、最も成功している手法の一 つです。手法上の大きな特徴は、対象系をフラグメント単位に分割し、環境静電ポテンシャル(ESP の印加条件でフラグメント単体(モノマー:1体)、フラグメント対(ダイマー:2体)…の計算を並 列実行することで、生来的に高い並列処理親和性を有しています。FMO 法では、系全体の分極や 電子の非局在化などの量子論的な効果が取り込める点で古典(MM)や量子/古典混成)(QM/MM)と は異なります。また、全系をフルに QM で扱う最近の手法(いわゆるオーダーN 系など)とも違い、 フラグメント間相互作用エネルギー(IFIE や PIE と呼ばれる)などの「物理化学的な内部情報」が 得られるので、FMO 法は計算対象に対する解析ツールとしての機能も併せ持つことにもなります。

FMO 法のプログラムとしては、米国の Iowa 州立大の Gordon 研/産総研の D.G.Fedorov 氏ら が手掛けている GAMESS-US[1,3]が機能豊富でメジャーな存在ではありますが、私たちが自主開 発してきました ABINIT-MP[2,4,5]、長崎大の石川岳志氏による PAICS[6]もあります。これまで FMO 計算は、上述のフラグメント間相互作用エネルギーをベースにタンパク質・リガンド複合系 などを対象に創薬や生物物理のバイオ系分野で主に使われてきました[2-4]。しかし最近では、ナ ノバイオ・非バイオ系のものづくり分野への適用など応用範囲が広がってきています。特に ABINIT-MP は、ポスト「京」のプロジェクト(フラッグシップ 2020)では、ナフィオンなどのイ オン交換性高分子と水の界面の相互作用解析などに粗視化シミュレーションと連携して利用され ることになっています。

これ以下、本稿では ABINIT-MP の富士通系マシン(「京」、FX-10、FX-100)での ABINIT-MP による FMO 計算の性能評価について概要をまとめます。詳細なタイミングデータやグラフなど は、別提出の資料をご参照いただければと思います。

【FMO 計算の流れ】 FMO 計算は多数のステップから構成されており、各々で演算のタイプも 異なります[2-4]。ここでは、ジョブとして最もよく利用される2体フラグメント展開による2次 摂動論(FMO-MP2: FMO2-MP2として2体を強調する場合もあり)の流れについて概説します。

先ずフラグメントのモノマー計算では、各フラグメントに対して ESP を印加した条件で分子軌 道の最適化をハートリーフォック(HF)レベルで行います。各フラグメントは複数のコアで並列計 算され、フラグメントモノマーのリストもまた並列化されますので並列処理としては 2 階層とな ります。フラット MPI でも出来ますが、「京」や FX-10 などの富士通系マシンではフラグメント 内の処理を OpenMP でスレッド並列し、フラグメントのリストを MPI でプロセス並列する混成 型で実行することが普通です。各フラグメントの HF 計算では添字を 4 つ有する基底関数添字の 2 電子積分を生成し、密度行列と縮約して Fock 行列を構築するステップがコストを決しており、 フラグメント内で自己無撞着場(SCF)条件が成立するまで構築と対角化が反復されます。また、 Fock 行列要素に加算される ESP 項の計算も時間がかかりますので、幾つかの近似を要求精度に 応じて導入しています。各フラグメントの電子分布は ESP を通じて互いに依存しているため、系 全体として自己無撞着な電荷場(SCC)を満たすようにサイクルとして繰り返す必要があり、ここ で分極効果が取り込まれます。ただ、モノマーはノード内でのみ OpenMP で並列化されるので、 系が小さい・フラグメント数が少ない場合には多数のノードが「遊ぶ」状態となることには注意 が要ります。一旦 SCC 条件が満たされれば、各フラグメントで MP2 計算のステップに進みます が、処理の本質は基底関数添字から分子軌道添字への4 段階の線形変換処理ですので DGEMM に 代表される BLAS で効率よく処理が出来ます(HF と違って繰り返しの必要は無)。

フラグメントモノマーのステージが終われば、ダイマーの計算に進みます。ここでは、ESP は モノマー段階で決まったものを使うので全体での繰り返しは不要で、近接モノマーの組から成る ダイマーのタスクリストに応じて HF/MP2 計算が進行し、電子の非局在化が取り込まれます。ダ イマーのリストは組み合わせとなりますので、ノードを有効利用することが出来ます。モノマー と同様、ダイマーも OpenMP でノード内で並列処理されます(コストはモノマーよりも大)。また、 離れたモノマーから成るダイマーは HF を省いて静電近似で済ませて高速化を図っています。 FMO2-MP2 ジョブのコストの増加は、対象分子系の形状にも依りますが、実質的には系の大き さに対して 2 乗より低い依存性となり、FMO 計算の実用性・汎用性の高さに直結しています。

【ABINIT-MPの現況】 東大生産研のサイト[5]では、2014年の7月末から Ver.7の Intel Xeon 用の ABINIT-MP[4]のバイナリがフリーでダウンロード可能です。また、「京」では Ver.6+が 2014 年度末に共用ライブラリとして登録され、創薬コンソーシアムなどで利用されています[8]。

ABINIT-MP では、フラグメントのトリマーとテトラマーまで含める4体展開によるFMO4法 [7]に一つの特徴があり、高い空間解像度を持つ相互作用解析や固体(バンドギャップは有)のモデ リングなどが可能です。種々の工夫により、FMO4-MP2のFMO2-MP2に対する相対的なコスト は最大で10倍程度で済むようになっています。もう一つ特徴的なのは、2電子積分のコレスキー 分解(CD[9])によって計算時間の短縮が図れることです。CDでは、4添字のテンソルと見なせる 2電子積分を3添字のテンソルの積和の形に分解して、BLASでの処理を積極的に活用すること で加速を得ます。富士通系ではDGEMMが特に高速であるため、CDの効果は大きいと期待され ます。ただ、CDでは分解補助基底が基底関数の積に取られるため、最適化された補助関数を別 途用意する恒等分解(RI)[10]に比すとコスト高になりがちで、計算精度は若干落ちますが1中心 (1C)制限などを課すことも多々あります。

ABINIT-MP Ver.7 のその他の機能では、3 次摂動(MP3)エネルギー[11]、MP2 レベルのエネル

ギー微分を利用した部分構造最適化 [12]や直接分子動力学(FMO-MD)[13]、 軌道レベルでの相互作用解析などがあ ります。また、専用 GUI の BioStation Viewer も提供されていますので、フラ グメント情報などの入力データのセッ トアップ、ならびに出力データとして得 られる相互作用エネルギー(IFIE)のリ



ストの可視的な理解が容易に行えます。上図は、タンパク質の FMO 計算による解析のワークフ ローを例示したものですが、アミノ酸残基数が数十よりも多い場合、結果の把握には GUI の利用 は不可欠となってきます。もちろん、BioStation Viewer は非バイオ系の固体や分子性結晶に対し ても使うことが出来ます[4]。

【「京」でのベンチマーク結果】「京」での実行テストは、2012年の夏から、それまで ABINIT-MP がプラットフォームとしてきた Intel x64 系のサーバや NEC の SX の実行結果と数値的に一致す るかどうかの検証から始めました。Ver.6 を使い、OpenMP/MPI 混成並列で本格的にベンチマー クを行ったのは 2014 年度ですが、d 関数に関する積分生成ルーチン群を Opt3 でコンパイルする と値が不正となる障害があり、当該箇所のみ Opt2 適用とする暫定措置でデータを取得しました(d 関数は 6-31G\*や cc-pVDZ 基底で含まれます)。

先す、24ノードで測定した	Basis set	TEI treatment	Time (Min)	Eff. (%)	Time (Min)	Eff. (%)
小タンパク質の FMO2-MP2			TrpCage	(20 frag.)	Tr	p23His
ジョブのタイミングを右表に	6-31G	Regular	12.6	2.3	36.8	2.53
		CD/MP2	8.2	4.4	19.5	7.13
示します。TrpCage は 20 残		CD/MP2&HF	5.7	15.6	14.6	18.35
基で主鎖分割して 20 フラグ		1C-CD/MP2	7.6	2.5	16.7	2.97
		1C-CD/MP2&HF	3.0	8.8	8.2	9.66
メント、粒度を掴えたモテル	6-31G*	Regular	38.2	3.0	136.9	3.43
タンパク質の Trp23His は 24		CD/MP2	28.0	6.6	96.0	8.87
フラガメントでナ(FMO の結		CD/MP2&HF	18.9	29.0	44.6	31.89
フラウブラド Cy(FMO の)桁		1C-CD/MP2	25.9	3.4	84.7	3.88
合切断の都合から 24 番目の		1C-CD/MP2&HF	8.3	18.8	22.0	21.06
残基は小さ目でバランスさせ	cc-pVDZ	Regular	76.7	2.9	267.8	3.52
		CD/MP2	55.8	6.1	188.6	8.06
ます)。2電子積分を CD で扱		CD/MP2&HF	34.7	26.3	81.0	30.43
い、HF ステップも BLAS で		1C-CD/MP2	52.1	3.2	170.1	3.77
加理する古が直効率とわり		1C-CD/MP2&HF	16.6	14.9	39.1	18.86
爬生り ロカル 同効平となり、						

6-31G\*基底では良好な効率を出していることが見てとれます。また、1C 化で効率は低下します がジョブ時間はさらに短くなっています。

続けて、実タンパク質として HIV-1 プロテアーゼを使った FMO2-MP2 計算の結果を下表に示 します。残基数は 198、中心部に嵌るリガンドであるロピナビルを 4 分割、さらに水分子を含め た総数で203フラグメントの系を204ノードで計 TEI treatment Time (Min) Eff. (%) Basis set HIV-1 Protease + Lopinavir (204 frag.) @ 204 nodes 算しました。実行効率は、やはり CD で HF と MP2 6-31G Regular 31.9 1.85 を共に処理すると高くなり、計算時間はほぼ半分 CD/MP2&HF 18.8 7.78 1C-CD/MP2&HF 3.74 14.8 となります。通常積分での 6-31G のコストと 6-31G\* Regular 74.5 2.67 1C-CD での 6-31G\*のコストがほぼ同じであり、d CD/MP2&HF 44.0 17.54 分極関数を含み信頼性に勝る 6-31G\*の利用が容 1C-CD/MP2&HF 29.4 8.30 cc-pVDZ Regular 2.69 161.3 易になっています(最近の FMO 応用計算では 17.47 CD/MP2&HF 80.4 6-31G\*が標準基底です)。一方で、cc-pVDZ は原 1C-CD/MP2&HF 58.3 8.02 始関数の多さから 6-31G\*に比べても相対的に時間がかかっています。

FMO4-MP2 ジョブでは、組合わせ的に多くなるテトラマーの処理、特に繰り返し計算を伴う HF ステップが占めてしまいます。メモリ要求の関係からテストでは 6-31G 基底とし、43 残基の クランビンと HIV-1 プロテアーゼ(共に主鎖と側 鎖を分割で総数で各々94、363 フラグメント)でタ イミングデータを取りました。右表より、通常 2 電子積分に比べて CD では約 1/3 の時間で計算出 来ることが分かります。

FMO4 計算でノード数使用最多の例は、インフ ルエンザウィルスのノイラミニダーゼ(386 残基)

Basis set	TEI treatment	Time (Min)	Eff. (%)
Crambin (9-	4 frag.) @ 960 nodes	5	
6-31G	Regular	23.7	2.06
	CD/MP2&HF	9.1	12.06
	1C-CD/MP2&HF	6.7	6.38
HIV-1 Prot	ease + Lopinavir (36	3 frag.) @ 6	000 nodes
6-31G	Regular	144.3	2.79
	CD/MP2&HF	128.0	11.85
	1C-CD/MP2&HF	49.5	6.88

とタミフルの複合系(総数では706フラグメント)の FMO4-MP2/6-31G(HFとMP2を共に1C-CD で計算)の12504 ノードでのテストジョブで、計算自体は1.1 時間で完走しています(計算結果の ファイルリスティングまで含めると1.5 時間)[4]。計算時間全体での効率は6%ですが、全体の半 分以上のコストがSCC条件を課すフラグメントモノマーのHFステップとなってしまっています。 FMO4 では、このあたりに抜本的な改善を図る必要がありそうです。

【FX-10とFX-100での結果】 RIST 神戸のFX-10は、共同研究により2014年度から利用させていただいています。2014年12月のSS研の第一回の報告時には前述のコンパイラの問題があり、d 関数積分ルーチンをOpt2としてコンパイルしたものでジョブタイミングを紹介し、PAによる解析で2電子積分のCD化によってDGEMM処理比率が大きく改善されることをお示ししました(通常積分で10%がCDで74%、1C-CDで63%)。コンパイラの件は、その後すぐ富士通さんにご対応をいただき、"Technical Computing Suite V1.0L30"以降のコンパイラ環境では正常オブジェクトが生成されることが確認されましたので、2015年からはデフォールトのkfastオプションでコンパイルしています。幸い、幾つかの系でテストしたところ、6-31G\*を使う場合でもジョブの計算時間短縮は数%しか変わりませんでした(d 関数が1項展開で相対コストが小さいため)。そこで、FX-10のデータは基本的に第一回報告のままとしました(ノード内は16スレッドのOpenMP並列とし、ノード数を増加させてスケーリングに注目)。JAXAのFX-100でのタイミング測定は、コードと入力データなどー式を高木亮治先生に委託して行っていただきました(2015年秋)。これ以下、注記しない限りはFMO2-MP2/6-31G\*ジョブでの結果を示していきます(CDの場合はHFもMP2もCDで実行)。また、"Conv"は通常2電子積分の使用を意味します。

FX-	10		Chignolim				Trp9His		
Method	#nodes	Monomer	Dimer	Total	Acc	Monomer	Dimer	Total	Acc
Conv-MP2	1	2111.0	5394.2	7505.2	1	3835.7	29524.3	33360.1	1
	2	1076.7	2789.2	3865.9	1.94	1941.1	15101.3	17042.4	1.96
	5	520.0	1257.4	1777.5	4.22	809.3	5905.1	6714.4	4.97
	10	457.5	834.5	1292.0	5.81	413.8	3297.4	3711.2	8.99
Full-CD	1	1390.1	1728.1	3118.2	1	4164.3	7943.6	12107.9	1
	2	710.1	875.6	1585.7	1.97	2117.2	4051.3	6168.5	1.96
	5	501.3	394.1	895.5	3.48	888.5	1619.0	2507.5	4.83
	10	500.1	230.5	730.6	4.27	473.1	890.5	1363.5	8.88
1C-CD	1	568.6	762.9	1331.5	1	1635.4	3398.0	5033.4	1
	2	294.9	387.6	682.6	1.95	832.5	1730.5	2563.0	1.96
	5	177.8	171.2	349.0	3.82	347.0	695.5	1042.5	4.83
	10	176.9	99.9	276.8	4.81	184.4	384.1	568.5	8.85

最初に、10 残基の Chignolin、 粒度を揃えた Trp9His の FX-10 での結果を下表にまとめます。

ノード数増に対する加速ですが、大小のアミノ酸残基を複数種含む Chignolin では 10 ノードでは 完全にドロップしてしまっており、CD では 2 電子積分関係のコストが下がるために劣化が目立 つ形になっています(特にモノマーの部分)。一方、ジョブ時間が 4 倍ほどかかることからも明ら かなように、Trp9His では Trp が処理的に重く、かつ数が揃っているために良好なスケーリング を示しています。時間短縮としては、1C 化の効果が確かにあります。

続いて FX-100 での結果ですが、こちらはノード内で 16 スレッドで 2 プロセスまで立てられるので、ノードを跨ぐ場合/跨がない場合の 2 通りを試していますが、大きな差はありませんでした。

FX-	100		Chignol	im			Trp9H	is	
Method	#nodes (n-p-t)	Monomer	Dimer	Total	Acc	Monomer	Dimer	Total	Acc
Conv-MP2	1-1-16	614.3	3132.2	3746.6	1	1924.2	15887.8	17811.9	1
	1-2-16	326.7	1622.7	1949.4	1.92	976.1	8138.7	9114.8	1.95
	2-2-16	325.8	1620.3	1946.1	1.93	977.5	8215.3	9192.8	1.94
	5-5-16	237.8	838.4	1076.2	3.48	403.7	3554.1	3957.9	4.50
	5-10-16	230.7	564.3	795.0	4.71	208.2	2077.5	2285.6	7.79
	10-10-16	230.1	563.1	793.2	4.72	208.5	2090.5	2299.0	7.75
Full-CD	1-1-16	713.5	783.6	1497.1	1	2081.6	3593.6	5675.2	1
	1-2-16	347.8	394.6	742.4	2.02	1058.8	1833.7	2892.5	1.96
	2-2-16	349.8	395.9	745.7	2.01	1061.2	1831.5	2892.7	1.96
	5-5-16	244.4	180.5	424.8	3.52	439.1	740.6	1179.8	4.81
	5-10-16	244.8	106.4	351.2	4.26	238.0	407.2	645.2	8.80
	10-10-16	245.3	106.4	351.7	4.26	239.2	407.9	647.1	8.77
1C-CD	1-1-16	291.0	363.4	654.4	1	826.0	1712.2	2538.2	1
	1-2-16	151.8	184.8	336.6	1.94	418.4	872.1	1290.5	1.97
	2-2-16	151.8	184.7	336.5	1.94	419.6	871.3	1290.9	1.97
	5-5-16	85.7	85.2	170.9	3.83	173.9	355.1	529.1	4.80
	5-10-16	87.8	51.9	139.7	4.68	92.7	195.7	288.4	8.80
	10-10-16	88.8	51.9	140.6	4.65	93.2	195.6	288.8	8.79

これは、FX-100のネットワークの高速性の証左とも言えるでしょうか。FX-10とのジョブ時間の 比較では、約2倍速くなっていることが分かります。スケーリングの傾向はFX-10と同様ですが、 CPU コアの速度が速い分、ノード数が多い場合は僅かですが低下が見られます。

次に、FX-100 による HIV-1 プロ テアーゼのタイミングを右に示しま Mす。ノード内で 16 スレッドを 32 に 倍化させてみたのですが、1.4 倍程 度の加速に留まりました。また、ス レッドを 8 に半減してプロセス数を 倍化したテストも行い、若干の加速 を確認しました。こうした実タンパ ク質の応用計算で、条件出しの試行 錯誤はあまり現実的ではありません

FX-	100	HIV-1 protease					
Method	#nodes	Monomer	Dimer	Total	Acc		
	(n-p-t)		JIOHEI DIHEI IOAI				
Conv-MP2	36-36-16	853.3	5093.1	5946.5	1		
	36-36-32	549.8	3571.4	4121.2	1.44		
	72-72-8	916.2	4458.8	5375.1	1.11		
Full-CD	36-36-16	913.8	2613.3	3527.1	1		
	36-36-32	658.1	1950.9	2608.9	1.35		
	72-72-8	883.3	2072.8	2956.1	1.19		
1C-CD	36-36-16	637.1	2146.4	2783.5	1		
	36-36-32	437.8	1622.8	2060.6	1.35		
	72-72-8	529.8	1648.4	2178.2	1.28		

ので、ノード内を8ないし16スレッドとして、プロセス-ノード数を積んでいく方が実用的かと 思われます。比較対象のFX-10ですが、ノード内で16スレッド、総数で36ノードでのタイミン グは、"Conv"、"Full-CD"、"1C-CD"の順に9552.5秒、4883.5秒、3296.7秒です。FX-100では CPU コアの速度が速い分、少ないノード数でも FX-10 よりも速くジョブを終わらせることが出 来ますので、ノード資源を上手く分割して複数のサンプルを同時並行的に(capacity computing) 流す方が有利と考えられます。

最後に、2015 年の秋にアルゴリズムを改良して再実装した CD ベースの FMO2-MP3/6-31G\* 計算の FX-10 でのベンチマーク結果を示します。MP3 ではメモリ要求が MP2 よりも大きくなる ため、Ala9Gly を選びました。MP2 の結果も最新のコンパイラを使ってのバイナリでのものです。

FX-10		FMO2-MP2 - Ala9Gly			FMO2-MP3 - Ala9Gly				
Integral	#nodes	Monomer	Dimer	Total	Acc	Monomer	Dimer	Total	Acc
Conv	1 815.4 982.1		982.1	1797.5	1	852.1	3024.4	3876.5	1
	2	470.1	490.5	960.6	1.87	487.8	1524.9	2012.7	1.93
	5	228.5	202.5	430.9	4.17	235.7	648.7	884.4	4.38
	10	100.6	103.0	203.6	8.83	111.7	427.5	539.2	7.19
Full-CD	1	312.6	317.6	630.2	1	379.0	6604.1	6983.0	1
	2	163.2	158.7	321.9	1.96	202.7	3388.5	3591.2	1.94
	5	70.9	65.2	136.2	4.63	93.2	1480.7	1573.9	4.44
	10	57.6	33.0	90.6	6.96	80.2	875.7	955.9	7.31
1C-CD	1	133.2	147.7	280.9	1	163.6	3107.2	3270.7	1
	2	69.2	74.6	143.8	1.95	87.3	1602.3	1689.6	1.94
	5	30.2	30.8	61.0	4.60	41.0	697.5	738.5	4.43
	10	25.3	15.9	41.2	6.82	35.7	420.3	456.0	7.17

比較のために通常 2 電子積分での計算も行いましたが、MP3 エンジン[11]が DGEMM を多用し つつ、無駄な演算を巧妙に省くように高度に最適化されているため、1C 化してない Full CD で はむしろ計算時間が増えてしまっており、1C でも時間短縮の効果は僅かです。主な理由は、CD では既述のように分解基底を基底関数自身の積とするので Full の場合、そこで自乗の依存性が伴 ってしまい、MP3 ではテンソル積和のオーダーが MP2 よりも本質的に 2 だけ多いこともあり、 演算数が実効的に増加して DGEMM 処理による加速でもカバーし切れないためと考えられます。 1C 制限では分解基底の数が減りますのでコストは下がりますが、CD の代わりに RI[10]を導入す ることがより根本的な改善策になると思われます。スケーリングについては、このテスト系では MP3 と MP2 は同程度のようですが、今後は他の系でもテストをしてみたいところです。

【ABINIT-MP の今後】 ABINIT-MP には、東大の生産研サイト[5]からバイナリでダウンロー ド可能な版とは別に、主に開発経緯的な事由から「ローカル版」が存在しています[4]。そちらで は、励起エネルギー[14]や動的分極率の算定[15]、さらに結合クラスター展開による高精度エネル ギー計算[16]などが利用出来ます。こうした機能に関心を持たれる方も居られますし、手持ちの マシン環境によっては再コンパイルやチューニングのためにソースを所望される場合もあります。

現在、産官学を交えたコンソーシアム的な組織で ABINIT-MP のソース共有を行い、創薬系だ けでなくものづくり分野への本格的な対応など、継続的なコード開発と保守を図っていくための 準備を進めています。これに併せて、版管理も Open シリーズに移行します。Ver.7 に準拠した最 初の Open Ver.1 については、2015 年度内のリリースを目指しています。また、2016 年度に整備 する Open Ver.2 では、密度汎関数(DFT)のモジュールを分子研の石村和也氏の SMASH[17]から、 また上記で懸案の RI 関係のモジュール(C 言語で記述)を PAICS[6]から移植する他、「ローカル版」 の幾つかの機能を含める予定です。また、ホットスポットの洗い直しやメモリ割り当ての見直し なども行い、パフォーマンスの総合的な向上も図ります。Open Ver.3 以降も機能充実を順次進め ていきます。

ソース共有とは別に Open シリーズでも従来のバイナリでの提供も続けるつもりですが、これ までの Intel Xeon 系と富士通系だけでなく、NEC のベクトル系、さらにパソコン用に Windows 64 ビット系を提供しようと考えています。なお、ソース共有コンソーシアムのポータルサイトは 「計算工学ナビ」[18]を予定しています。

Open シリーズのベンチマークは、2016 年度も引き続いて RIST の FX-10 を使って行っていき ますが、FX-100 もチャンスがあれば是非利用したいところです。他方、GAMESS では Xeon Phi (KNC)[19]や ARM-32/-64[20]での試行が既に報告されています。小研究室でも SMASH を使った Phi でのテストを始めており、2016 年度は ABINIT-MP でも行うことになっています。Phi や ARM などの省エネルギーに優れる加速器的な技術は今後は重要性が増す可能性も高く、継続して ウォッチしていく必要を感じているところです。

【謝辞】 ABINIT-MP の研究開発は、様々な資金援助を頂戴してここまで続けてくることが出 来ました。本稿に記載した内容に関連しては、文科省の「HPCI 戦略プログラム 分野 4」(代表者: 東大 加藤千幸教授)と「フラッグシップ 2020 プロジェクト 課題6」(代表者:東大 吉村忍教授) から支援を受けています。後者はポスト「京」に関する研究活動で、ものづくり文脈を明確にし つつ ABINIT-MP - Open シリーズの改良・機能拡張が 2019 年度まで継続されていく予定です。

個人としましては、SS 研の本 WG への参加をお誘いくださいました JAXA の高木亮治先生に 先ず感謝したいと思います。「京」での実行は、沖山佳生氏と渡邉千鶴氏(共に元東大:現理研)に 依ります。FX-10 でのベンチマークは、RIST 神戸の山木大輔、野口孝明、小久保達信、吉澤香 奈子の諸氏のご支援の下、小研究室の齊藤天菜君と坂口正貴君が行ってくれました。また、 ABINIT-MP のコアメンバである神戸大の田中成典先生、日大の福澤薫先生、NEC の坂倉耕太氏 にも謝意を表します。

【文献】 [1] M. S. Gordon et al., *Chem. Rev.* **112** (2012) 632. [2] "The Fragment Molecular Orbital Method: Practical Applications to Large Molecular Systems", (2009, CRC). [3] D. G. Fedorov et al., *Phys. Chem. Chem. Phys.* **14** (2012) 7562. [4] S. Tanaka, et al., *Phys. Chem. Chem. Phys.* **16** (2014) 10310. [5] <a href="http://www.ciss.iis.u-tokyo.ac.jp/software/">http://www.ciss.iis.u-tokyo.ac.jp/software/</a>. [6] T. Ishikawa et al. *J. Comp. Chem.* **30** (2009) 2594. [7] T. Nakano et al., *Chem. Phys. Lett.* **523** (2012) 128. [8] 「京算百景」, Vol. 9, 2014 年 3 月 10 日. [9] Y. Okiyama et al., *Chem. Phys. Lett.* **490** (2010) 84. [10] T. B. Pedersen, *Theor. Chem. Acc.* **124** (2009) 1. [11] Y. Mochizuki et al., *Chem. Phys. Lett.* **493** (2010) 346. [12] T. Tsukamoto et al., *Chem. Phys. Lett.* **535** (2012) 157. [13] Y. Komeiji et al., *Intech J.* <a href="http://bit.ly/Hkefpl">http://bit.ly/Hkefpl</a>. [14] Y. Mochizuki et al., *Theor. Chem. Acc.* **117** (2007) 541. [15] Y. Mochizuki et al., *Chem. Phys. Lett.* **418** (2006) 418. [16] Y. Mochizuki et al., *Theor. Chem. Acc.* **130** (2011) 515. [17] K. Ishimura, <a href="http://smash-qc.sourceforge.net/">http://smash-qc.sourceforge.net/</a>. [18] <a href="http://www.cenav.org/">http://www.cenav.org/</a>. [19] V. Mironov et al., *ISC High Performance 2015*, LNCS **9137** (2015) 113. [20] K. Keipert et al., *J. Chem. Theor. Comp.* **11** (2015) 5055.



### FMOプログラムABINIT-MPの富士通系マシンでの性能評価



◇巨大分子系 生体高分子や凝集系では一般的 ⇒ タンパク質、DNA (水和状態)

数千~数万原子、数千~数十万軌道

### ◇分割&統合系のアプローチの一つ

北浦らが15年程前に2体展開で提案



【HIVプロテアーゼとロピナビル】

- ⇒ フラグメントとその対で系のエネルギーを評価 (FMO2)
- ⇒ 環境静電ポテンシャル (ESP)、直接結合切断 (BDA)
- ⇒ 階層的な並列処理(フラグメントリスト&内部処理)
- ⇒ 電子相関の導入、構造最適化も可能
- フラグメント間の相互作用エネルギー(IFIE)
  - ⇒ 計算対象の解析ツール
  - ⇒ 創薬設計には好適
  - ⇒ ナノ系の問題でも利活用可能

$$E = \sum_{I>J} E_{IJ} - (N_f - 2) \sum_{I} E_{I}$$

$$E = \sum_{I>J} \Delta E_{IJ} + \sum_{I} E'_{I}$$

### 基本のHF計算

$$\mathbf{F}^{x} \mathbf{C}^{x} = \mathbf{S}^{x} \mathbf{C}^{x} \mathbf{\varepsilon}^{x} \qquad \mathbf{F}^{x} = \mathbf{H}^{x} + \mathbf{G}^{x} \quad \Leftrightarrow \quad \mathsf{HFO} - \mathbf{@} \iota \square \mathbf{fa} \mathbf{fa}$$

### 高速化のための幾つかの工夫

 $V_{\mu\nu}^{L} \cong \sum_{\lambda \in L} (\mathbf{P}^{L} \mathbf{S}^{L})_{\lambda\lambda}(\mu\nu, \lambda\lambda)$  for  $R_{\min}(X, L) \ge L_{aoc}$  ⇔ ESP-AOC近似 (実用精度高し)  $V_{\mu\nu}^{L} \cong \sum_{A=L} \left\langle \mu \left| \left( Q_{A} / |\mathbf{r} - \mathbf{A}| \right) | \nu \right\rangle \quad \text{for } R_{\min}(X, L) \ge L_{ptc} \qquad Q_{A} = \sum_{\lambda = L} (\mathbf{P}^{L} \mathbf{S}^{L})_{\lambda\lambda} \quad \Leftrightarrow \quad \text{ESP-PTC 近似} \quad (遠い)$ 

 $E'_{IJ} \cong E'_{I} + E'_{J} + \operatorname{Tr}(\mathbf{P}^{I}\mathbf{u}^{J}) + \operatorname{Tr}(\mathbf{P}^{J}\mathbf{u}^{I}) + \sum_{\mu\nu \in I} \sum_{\lambda\sigma \in J} \mathbf{P}_{\mu\nu}^{I} \mathbf{P}_{\lambda\sigma}^{J}(\mu\nu, \lambda\sigma) \quad \Leftrightarrow \quad \operatorname{Dimer-ES}_{\mathcal{I}} \mathcal{U} \quad (\mathsf{HF}_{\mathsf{I}})$ 

Ref.; Szabo & Ostlund, "Modern Quantum Chemistry".



Refs.; Mochizuki et al., Chem. Phys. Lett. 493 (2010) 396. & Mochizuki et al., Theor. Chem. Acc. 130 (2011) 515. MPI/OpenMP混成並列化 フラグメント間をMPI、フラグメント内をOpenMPで並列化 ⇒ 混成並列化 Target system Fragment momer Parallelized by fragment (or dimer) indice MPI並列→ OpenMP並列→ Parallelized by integral Processor group indices 作業配列をスレッド間で共有してメモリを節約 ABINIT-MPではES2用に開発、メモリの少ない京では必然 従来の並列化(MPI) SMP並列化(OpenMP) CPU0 CPU1 CPU2 CPU3 メモリ CPU0 CPU1 CPU2 CPU3 メモリ MPI诵信 5 2016/3/1

Recent ABINIT-MP Ref.; Tanaka et al., Chem. Phys. Phys. Chem. 16 (2014) 10310



## MP2計算の基本アルゴリズム2種:形式的にN<sup>5</sup>

		スピンを積分した実装式
Loop over <i>i</i> -batch [parallelizable when needed] Loop over <i>s</i> [to be parallelized for worker processes] Loop over <i>r</i>	$E_{_{\rm I}}$	$_{\text{MP2}} = \sum_{ijab} \frac{(ia, jb) [2(ia, jb) - (ib, ja)]}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b}$
Generate $(pq,r[s])$ list [canonical relation for $pq$ ] Do 1/4 transformation of $p \rightarrow i$ [screening & DAXPY]		添字の変換: 4№コスト
Do 2/4 transformation of $q \rightarrow a$ [DDOT] Do 3/4 transformation of $r \rightarrow j$ [screening & DAXPY] End of loop over r	( <i>ia</i> ,	$jb) = \sum_{s} c_{sb} \left( \sum_{r} c_{rj} \left( \sum_{q} C_{qa} \left( \sum_{p} c_{pi}(pq, rs) \right) \right) \right)$
Do 4/4 transformation for $s \rightarrow b$ [screening & DAXPY] End of loop over s ["all-reduce" must be done for $(ia,jb)$ ]		
Calculate partial MP2 energy End of loop over <i>i</i> -batch		【第一版】 ・DAXPYとDDOTを使い、
Loop over <i>ij</i> -batch ! size depending on available memory Loop over <i>s</i> ! to be parallelized Loop over <i>r</i>		<ul><li> 閾値判断を優先 </li><li> 実効的演算数を下げる方針、 </li><li> 10年前のチップでは有効 </li></ul>
Preparing $(pq,rs)$ ! for canonical $pq$ -pair Forming $(iq,rs)$ ! DGEMM, fixed $rs$ , running over $p$ Forming $(ia,rs)$ ! DGEMM, fixed $rs$ , running over $q$ Forming $(ia,is)$ ! DGEMM, fixed $s$ , direct-product for fix	ed r	【第二版】 ・DGEMMの高性能に期待
End of loop over $r$ Forming $(ia,jb)$ ! DGEMM, direct-product for fixed $s$ End of loop over $s$ ! all-reduce operation as barrier Calculate partial MP2 energy with respect to <i>ii</i> -batch		【現在:混成型】 ・スカラ型のCPUでもステップ2 とステップ4はDGFMMで
End of loop over <i>ij</i> -batch		

Ref.; Okiyama et al., Chem. Phys. Lett. 490 (2010) 84. / general CD Ref.; Pedersen et al., Theor. Chem. Acc. 124 (2009) 1.

-

コレスキー分解(CD)による近似: テンソル次致の低減  
MP2 correlation energy (E<sup>MP2</sup>)  
Loop over I # Parallelized  

$$Q_{l} = \sum_{r_{s}} L_{l,rs} P_{rs}$$
 # DDOT  
 $\tilde{J}_{pq} = \tilde{J}_{pq} + L_{l,pq} Q_{l}$  # DAXPY  
End of loop over I  
ALLREDUCE ( $\tilde{J}_{pq}$ )  
Exchange matrix (K)  
 $\tilde{K}_{pq} = 0$   
Loop over I # Parallelized  
 $X_{l,qi} = \sum_{s} L_{l,qs} C_{si}$  # DGEMM  
 $\tilde{K}_{pq} = \delta$   
Loop over I # Parallelized  
 $X_{l,qi} = \sum_{s} L_{l,qs} C_{si}$  # DGEMM  
 $\tilde{K}_{pq} = \tilde{K}_{pq} + \sum_{i} X_{l,pi} X_{l,qi}$  # DGEMM  
End of loop over I  
 $\tilde{K}_{pq} = \tilde{K}_{pq} + \sum_{i} X_{l,pi} X_{l,qi}$  # DGEMM  
End of loop over I  
 $\tilde{K}_{pq} = \tilde{K}_{pq} + \sum_{i} X_{l,pi} X_{l,qi}$  # DGEMM  
End of loop over I  
 $\tilde{K}_{pq} = \tilde{K}_{pq} + \sum_{i} X_{l,pi} X_{l,qi}$  # DGEMM  
 $\tilde{K}_{pq} = \tilde{K}_{pq} + \sum_{i} X_{l,pi} X_{l,qi}$  # DGEMM  
 $(pq, rs) \approx \sum_{i} L_{l,pq} L_{l,rs}$   
 $\cdot$  CDでは分解基底は軌道積に取る: 展開長MitN<sup>2</sup>T -  $\mathcal{F}$ -  
 $\cdot$  CD 1中心(1C)化は同一原子内に限る: MitNic\_{LV}  
 $\cdot$  Ri近似では最適化基底を別途調製: 展開長はほぼ)

### 小型タンパク質での「京」での性能評価の例

FMO2-MP2ジョブ @ 24ノード									
Basis set	TEI treatment	Time (Min)	Eff. (%)	Time (Min)	Eff. (%)				
		TrpCage	(20 frag.)	Tr	p23His				
6-31G	Regular	12.6	2.3	36.8	2.53				
	CD/MP2	8.2	4.4	19.5	7.13				
	CD/MP2&HF	5.7	15.6	14.6	18.35				
	1C-CD/MP2	7.6	2.5	16.7	2.97				
	1C-CD/MP2&HF	3.0	8.8	8.2	9.66				
6-31G*	Regular	38.2	3.0	136.9	3.43				
	CD/MP2	28.0	6.6	96.0	8.87				
	CD/MP2&HF	18.9	29.0	44.6	31.89				
	1C-CD/MP2	25.9	3.4	84.7	3.88				
	1C-CD/MP2&HF	8.3	18.8	22.0	21.06				
cc-pVDZ	Regular	76.7	2.9	267.8	3.52				
	CD/MP2	55.8	6.1	188.6	8.06				
	CD/MP2&HF	34.7	26.3	81.0	30.43				
	1C-CD/MP2	52.1	3.2	170.1	3.77				
	1C-CD/MP2&HF	16.6	14.9	39.1	18.86				

NLYIQWLKDGGPSSGRPPPS



[TrpCage]

$$X_{I,qi} = \sum_{s} L_{I,qs} C_{si}$$

$$\widetilde{K}_{pq} = \widetilde{K}_{pq} + \sum_{i} X_{I,pi} X_{I,qi}$$

I

9

CDではHF計算の交換項(K)をDGEMMで処理

・モデル系では最高で30%を超える効率を達成

・トータルでの時間短縮は1Cでは最高6倍程度

2016/3/1

#### Regularの速さは「実効演算数」の少なさも効いている

### 実タンパク質での「京」での性能評価の例 FM04-MP2ジョブ

<b>FMO2-MP2ジ</b> ョブ									
Basis set	TEI treatment	Time (Min)	Eff. (%)						
HIV-1 Pro	HIV-1 Protease + Lopinavir (204 frag.) @ 204 nodes								
6-31G	Regular	31.9	1.85						
	CD/MP2&HF	18.8	7.78						
	1C-CD/MP2&HF	14.8	3.74						
6-31G*	Regular	74.5	2.67						
	CD/MP2&HF	44.0	17.54						
	1C-CD/MP2&HF	29.4	8.30						
cc-pVDZ	Regular	161.3	2.69						
	CD/MP2&HF	80.4	17.47						
	1C-CD/MP2&HF	58.3	8.02						
Infulenza H	IA1 + Fab (911 frag.	) @ 912 node	s						
6-31G	Regular	123.2	1.01						
	CD/MP2&HF	110.7	2.27						
	1C-CD/MP2&HF	106.1	1.14						
6-31G*	Regular	212.1	1.62						
	CD/MP2&HF	166.5	6.47						
	1C-CD/MP2&HF	146.5	2.66						
cc-pVDZ	Regular	400.5	2.32						
	CD/MP2&HF	296.6	8.14						
	1C-CD/MP2&HF	269.9	3.37						

Basis set	TEI treatment	Time (Min)	Eff. (%)				
Crambin (94 frag.) @ 960 nodes							
6-31G	Regular	23.7	2.06				
	CD/MP2&HF	9.1	12.06				
	1C-CD/MP2&HF	6.7	6.38				
HIV-1 Pro	otease + Lopinavir (3	63 frag.) @ 60	00 nodes				
6-31G	Regular	144.3	2.79				
	CD/MP2&HF	128.0	11.85				
	1C-CD/MP2&HF	49.5	6.88				
		1 200	Jo				

### ▶ FMO2計算

- → 数百残基の計算は容易
   → 6-31G\*基底の実用利用
   → 実行効率は及第レベル
   → 1中心CDAMで時間短縮
- → 「中心CDAMで時间短縮 (実用性には問題無し)

#### ➤ FMO4計算

- → 多ノード利用のメリット
- → 1中心化の効果は顕著
- → 詳細解析の用途を推奨
- → HFのコスト低減は課題





### FX-10、-100共に最新のコンパイラ環境でkfastでビルド(d積分ルーチン問題は解消済) FMO2-MP2タイミング #1 chignolin (10残基)

FX	-10	0	RIST
		6	

OLD											
		Method	Number of Proc	Elapsed time [secor	nd]						
Chignolin				Monomer SCF	Monomer MP2	Monomer (Total)	Dimer ES	Dimer SCF	Dimer MP2	Dimer (Total)	FMO (Total)
(10 res)	点線	Conv-MP2	1	1958.7	143.0	2111.0	10.2	1643.1	3608.9	5394.2	7505.2
			2	994.6	77.0	1076.7	6.4	777.9	1942.4	2789.2	3865.9
			5	471.1	46.4	520.0	5.6	424.0	802.3	1257.4	1777.5
			10	408.8	46.0	457.5	5.5	286.3	535.3	834.5	1292.0
	実線	Full-CD	1	1342.3	38.4	1390.1	10.2	1282.3	435.3	1728.1	3118.2
		(HF & MP2)	2	685.1	19.9	710.1	6.4	639.0	230.0	875.6	1585.7
			5	483.7	15.0	501.3	5.6	287.1	101.5	394.1	895.5
			10	482.5	14.9	500.1	5.5	161.4	63.6	230.5	730.6
	破線	1C-CD	1	547.8	11.5	568.6	10.2	539.7	212.6	762.9	1331.5
		(HF & MP2)	2	284.1	5.7	294.9	6.4	268.7	112.2	387.6	682.6
			5	171.0	4.2	177.8	5.6	116.6	49.0	171.2	349.0
			10	170.1	4.2	176.9	5.4	63.6	30.8	99.9	276.8

#### FX-100 @ JAXA

#### FX-10に比して約2倍の加速 ノード跨ぎの性能低下は少ない

FAIUU		(n-p-t)=(node-pro	cess-thread)								
		Method	Number of Proc	Elapsed time [second	nd]						
Chignolin			(n-p-t)	Monomer SCF	Monomer MP2	Monomer (Total)	Dimer ES	Dimer SCF	Dimer MP2	Dimer (Total)	FMO (Total)
(10 res)	点線	Conv-MP2	1-1-16	543.1	64.5	614.3	14.1	1194.9	1922.2	3132.2	3746.6
			1-2-16	287.6	35.1	326.7	9.5	602.7	1010.4	1622.7	1949.4
			2-2-16	286.7	35.2	325.8	9.3	601.7	1009.2	1620.3	1946.1
			5-5-16	208.2	27.8	237.8	8.4	360.1	469.8	838.4	1076.2
			5-10-16	201.2	27.6	230.7	8.5	262.0	293.7	564.3	795.0
			10-10-16	200.6	27.6	230.1	8.5	261.5	293.0	563.1	793.2
	実線	Full-CD	1-1-16	687.6	18.8	713.5	14.1	601.3	168.2	783.6	1497.1
		(HF & MP2)	1-2-16	334.8	9.5	347.8	9.4	297.9	87.3	394.6	742.4
			2-2-16	336.7	9.6	349.8	9.3	299.1	87.4	395.9	745.7
			5-5-16	235.5	7.0	244.4	8.5	133.0	38.8	180.5	424.8
			5-10-16	236.0	7.0	244.8	8.5	74.1	23.9	106.4	351.2
			10-10-16	236.5	7.0	245.3	8.5	74.0	23.8	106.4	351.7
	破線	IC-CD	1-1-16	279.2	5.3	291.0	14.2	263.6	85.7	363.4	654.4
		(HF & MP2)	1-2-16	145.5	2.7	151.8	9.5	130.5	44.7	184.8	336.6
			2-2-16	145.5	2.7	151.8	9.3	130.8	44.7	184.7	336.5
			5-5-16	82.0	1.9	85.7	8.5	57.0	19.7	85.2	170.9
			5-10-16	84.2	1.9	87.8	8.5	31.3	12.1	51.9	139.7
			10-10-16	85.1	1.9	88.8	8.5	31.3	12.1	51.9	140.6
-											

### FMO2-MP2タイミング #1 chignolin (10残基)

Gly1-Tyr2-Asp3-Pro4-Glu5-Thr6-Gly7-Thr8-Trp9-Gly10



# FX-100のグラフはノード数=プロセス数のものをプロット(スレッドは16) FMO2-MP2タイミング #1 chignolin (10残基)



### FMO2-MP2タイミング #2 Trp9His (10残基)

#### **FX-10 @ RIST**

OLD											
		Method	Number of Proc	Elapsed time [second	nd]						
Trp9His				Monomer SCF	Monomer MP2	Monomer (Total)	Dimer ES	Dimer SCF	Dimer MP2	Dimer (Total)	FMO (Total)
(10 res)	点線	Conv-MP2	1	3389.6	410.8	3835.7	104.5	7950.5	21336.9	29524.3	33360.1
			2	1713.1	210.2	1941.1	53.3	3643.3	10288.0	15101.3	17042.4
			5	715.6	86.1	809.3	22.5	1708.7	4147.5	5905.1	6714.4
			10	366.5	43.4	413.8	12.4	696.7	1895.2	3297.4	3711.2
	実線	Full-CD	1	3994.6	134.4	4164.3	104.5	5483.0	2355.9	7943.6	12107.9
		(HF & MP2)	2	2030.6	68.7	2117.2	52.9	2677.3	1158.2	4051.3	6168.5
			5	852.1	28.8	888.5	22.4	1133.1	463.5	1619.0	2507.5
			10	454.0	14.9	473.1	12.7	507.2	211.5	890.5	1363.5
	破線	IC-CD	1	1561.9	38.2	1635.4	104.5	2189.2	1104.2	3398.0	5033.4
		(HF & MP2)	2	794.9	19.7	832.5	53.2	1068.3	543.7	1730.5	2563.0
			5	331.3	8.1	347.0	22.5	454.6	218.4	695.5	1042.5
			10	176.4	4.0	184.4	12.6	202.9	100.5	384.1	568.5

#### FX-100 @ JAXA

#### Trp9Hisはモデル系で計算は重め 加速はchignolinより若干向上

ベンチ用に粒度を揃えた効果が見える

FX100		(n-p-t)=(node-pro	cess-thread)								
		Method	Number of Proc	Elapsed time [second	nd]						
Trp9His			(n-p-t)	Monomer SCF	Monomer MP2	Monomer (Total)	Dimer ES	Dimer SCF	Dimer MP2	Dimer (Total)	FMO (Total)
(10 res)	点線	Conv-MP2	1-1-16	1655.4	243.8	1924.2	173.2	6634.1	9080.3	15887.8	17811.9
			1-2-16	839.1	124.4	976.1	87.0	3186.8	4364.2	8138.7	9114.8
			2-2-16	838.6	125.2	977.5	87.2	3184.5	4401.3	8215.3	9192.8
			5-5-16	347.1	51.2	403.7	39.1	1553.9	1961.1	3554.1	3957.9
			5-10-16	179.6	25.8	208.2	21.5	640.4	825.9	2077.5	2285.6
			10-10-16	179.8	25.8	208.5	21.5	640.1	802.5	2090.5	2299.0
						_					
	実線	Full-CD	1-1-16	1992.2	64.1	2081.6	172.5	2503.4	917.5	3593.6	5675.2
		(HF & MP2)	1-2-16	1013.2	32.8	1058.8	86.9	1222.4	450.4	1833.7	2892.5
			2-2-16	1015.6	32.9	1061.2	87.0	1224.1	450.4	1831.5	2892.7
			5-5-16	420.4	13.5	439.1	38.8	519.0	182.8	740.6	1179.8
			5-10-16	228.0	7.2	238.0	21.5	230.9	82.6	407.2	645.2
			10-10-16	228.7	7.2	239.2	21.7	230.9	82.7	407.9	647.1
	破總	IC-CD	P 1-1-16	783.7	17.3	826.0	172.5	1069.4	470.3	1712.2	2538.2
	HIX WA	(HE & MD2)	1 2 16	206.7	80	418.4	97.2	521.7	220.7	872.1	1200.5
		(III' & WI 2)	2-2-16	398.0	8.9	410.4	87.0	521.7	229.1	871.3	1290.5
			P 5-5-16	165.0	3.7	173.0	30.1	222.7	03.3	355.1	529.1
		-	5-10-16	88.2	1.8	92.7	21.4	99.0	42.3	195.7	288.4
			10-10-16	88.4	1.0	93.2	21.5	99.1	42.3	195.6	288.8
			10 10 10	00.1	1.7	75.2	21.0	//.1	12.5	175.0	200.0
2016/3/	1										

OI D

15

# FMO2-MP2タイミング #2 Trp9His (10残基)



### FMO2-MP2タイミング #2 Trp9His (10残基)

FX-10比で約2倍の速さ



### 1ノード、16スレッドの条件で実行 Trp9His のPAでの解析(FX-10) - 通常2電子積分

\*\*\*\*\* Process 0 - performance monitors

Elapsed(s)	MFLOPS MFLO	OPS/PEAK (%)	MIPS	MIPS/PEAK(%)		
32768. 6626	7527.0432	3. 5639	24434.6821	23.1389	Process	0
32768.6626	465. 5478	3. 5269	1451.3261	21.9898	Thread	0
32768.3581	498.2169	3.7744	1781.9925	26.9999	Thread	13
32768.3581	475.7742	3.6044	1696.1898	25.6998	Thread	15
32768.3581	462.5215	3.5040	1537.1240	23.2898	Thread	2
32768.3581	460.4899	3.4886	1553.7291	23.5413	Thread	10
32768.3581	482.4262	3.6547	1543.6502	23.3886	Thread	6
32768.3581	458.8652	3.4763	1532.2277	23.2156	Thread	4
32768.3581	457.2214	3.4638	1527.3619	23.1418	Thread	8
32768.3581	467.7318	3.5434	1524.9341	23.1051	Thread	9
32768.3581	460.0703	3.4854	1515.9977	22.9697	Thread	3
32768.3581	468.2977	3.5477	1509.0344	22.8642	Thread	7
32768.3581	482.9977	3.6591	1493.0931	22.6226	Thread	5
32768.3581	477.7212	3.6191	1471.0861	22.2892	Thread	11
32768.3581	476.3939	3.6090	1468.9008	22.2561	Thread	12
32768.3581	469.9461	3.5602	1434. 1813	21.7300	Thread	1
32768.3581	462.8869	3.5067	1394.0669	21.1222	Thread	14

Application - procedures

Cost	%	Operation (S)	Barrier	%	Start	End	
 32714564	100.0000	327210. 6875	10514044	32. 1387			Application
 10733725	32.8102	107358. 5938	9787448	91.1841	453	638	dmp2_ene_ijOMP_1_
3413289	10.4335	34139.6758	6	0.0002	67	375	direct_scf_gmatOMP_1_
3348662	10.2360	33493.2773	2848	0.0850	1	12	dgemx_
2356285	7.2026	23567.5352	26183	1.1112	1	250	get_tei_pq_fix_
1443213	4.4115	14435.0000	0	0.0000	1	213	get_tei_rs_fix_
1092117	3. 3383	10923.3418	607526	55.6283	338	822	dmp2_ene_ij_
532148	1.6266	5322.5381	0	0.0000	16	77	sub_ssss_
495632	1.5150	4957.3057	0	0.0000	16	93	sub_spss_
380148	1.1620	3802.2361	0	0.0000	16	92	sub sssp
364106	1.1130	3641.7839	0	0.0000	16	90	sub ssps

Dgemxlt wrapした DGEMMで 10.2%

17

# Trp9His のPAでの解析(FX-10) - Full CD

 Process
 0 - performance monitors

Elapsed(s)	MFLOPS	MFLOPS/PEAK(%)	MIPS	MIPS/PEAK(%)		
12423. 5145	63398.9059	30.0184	32088. 5892	30. 3869	Process	0
12423. 5145	4053.8133	30. 7107	2194.8465	33. 2552	Thread	0
12423.4433	3970. 5250	30.0797	2007.2082	30.4122	Thread	6
12423.4433	3943.1042	29.8720	1994.2075	30.2153	Thread	11
12423.4433	3943.8243	29.8775	1993.6070	30.2062	Thread	9
12423.4433	3975.1896	30.1151	1997.4153	30.2639	Thread	4
12423.4433	3968.2847	30.0628	2004.1674	30.3662	Thread	7
12423.4433	3979.6754	30.1491	2002.3862	30.3392	Thread	2
12423.4433	3938.1847	29.8347	1986.7419	30.1022	Thread	15
12423.4433	3937.4160	29.8289	1986.0516	30.0917	Thread	14
12423.4433	3972.8625	30.0974	1994.5367	30.2203	Thread	5
12423.4433	3938.3377	29.8359	1986.1901	30.0938	Thread	13
12423.4433	3975.9605	30.1209	1995.8162	30.2396	Thread	3
12423, 4433	3939, 2306	29,8427	1984, 7733	30,0723	Thread	10
12423, 4433	3949,8235	29, 9229	1982, 9750	30,0451	Thread	8
12423.4433	3936.7083	29.8235	1983.3389	30.0506	Thread	12
12423. 4433	3976.3059	30. 1235	1994.4987	30.2197	Thread	1

\*\*\*\*\* Application - procedures \*\*\*\*\* 

Cost	%	Operation (S)	Barrier	%	Start	End		
 15642706	100.0000	156648.0000	95132	0.6082			Application	DGEMMは
 11624815	74.3146	116412.3359	13524	0.1163	1	12	dgemx_	74.3% と
857421	5.4813	8586.3203	27325	3.1869	86	196	direct_scf_gmat_cdOMP_1_	十キノト見
362373	2.3166	3628.8481	0	0.0000	95	247	coulomb_metricOMP_1_	ᆺᇰᆞᆂ升
202263	1.2930	2025.4867	0	0.0000	1	203	get_tei_pq_fix_cd_	
133570	0.8539	1337.5865	0	0.0000	16	77	sub_ssss_	*
127694	0.8163	1278.7435	0	0.0000	16	93	sub_spss_	性能向上
109860	0.7023	1100.1516	0	0.0000	16	93	sub_psss_	
81559	0.5214	816.7419	0	0.0000	16	115	sub_ppss_	
68158	0.4357	682.5427	0	0.0000	1	61	fmt_	
58923	0.3767	590, 0623	0	0.0000	119	304	monomer esp electron direct. O	MP 1

2016/3/1

Trp9His のPAでの解析(FX-10) - 1C CD

Process 0 - performance monitors

Elapsed(s)	MFLOPS MFL	.0PS/PEAK (%)	MIPS	MIPS/PEAK(%)		
5191.0808	48008. 0999	22.7311	26812.1375	25.3903	Process	0
5191.0808	3185. 3280	24.1313	1977.6491	29.9644	Thread	0
5191.0054	3019.4644	22.8747	1674.3123	25.3684	Thread	2
5191.0054	3004.6186	22.7623	1672.0772	25.3345	Thread	6
5191.0054	2973.7315	22.5283	1657.3430	25.1113	Thread	9
5191.0054	2972.3038	22.5175	1652.5802	25.0391	Thread	10
5191.0054	2997.8868	22.7113	1662.7503	25.1932	Thread	7
5191.0054	3009.6369	22.8003	1662.7523	25.1932	Thread	4
5191.0054	2969.3276	22.4949	1651.0246	25.0155	Thread	12
5191.0054	2970.0616	22.5005	1646.8130	24.9517	Thread	13
5191.0054	2965.5583	22.4664	1647.0634	24.9555	Thread	15
5191.0054	3007.3903	22.7833	1659.3869	25.1422	Thread	5
5191.0054	2969.7733	22.4983	1647.3608	24.9600	Thread	11
5191.0054	2979.9758	22.5756	1640.9404	24.8627	Thread	8
5191.0054	3011.4207	22.8138	1659.7717	25.1481	Thread	3
5191.0054	2961.2015	22.4333	1640.2363	24.8521	Thread	14
5191.0054	3011.0716	22.8111	1660.4365	25.1581	Thread	1

\*\*\*\*\*\* Application - procedures \*\*\*\*

*****	*****	*****	************	*******

Cost	%	Operation (S)	Barrier	%	Start	End	
 6019941	100.0000	60317.3945	37784	0.6276			Application
 3795973	63.0566	38034.1289	8628	0. 2273	1	12	dgemx_
398159	6.6140	3989.3936	12895	3.2387	86	196	direct_scf_gmat_cdOMP_1_
119597	1.9867	1198.3140	0	0.0000	16	77	sub_ssss_
103877	1.7255	1040.8059	0	0.0000	16	93	sub_spss_
103722	1.7230	1039.2528	0	0.0000	16	93	sub_psss_
81138	1.3478	812.9703	0	0.0000	1	203	get_tei_pq_fix_cd_
67445	1.1204	675.7719	0	0.0000	1	61	fmt
66832	1.1102	669.6298	0	0.0000	16	115	sub_ppss_
59766	0.9928	598.8314	0	0.0000	119	304	monomer esp electron direc
39887	0.6626	399.6518	0	0.0000	91	351	dimer_oneint_debugOMP_1_

249 / 253

nppiiodoion
dgemx_
direct_scf_gmat_cdOMP_1_
sub_ssss_
sub_spss_
sub_psss_
get_tei_pq_fix_cd_
fmt_
sub_ppss_
monomer_esp_electron_directOMP_1_

**DGEMMは** 63.1%

19

### FMO2-MP2タイミング #3 HIV-1 protease (198残基)

### FX-10 @ RIST

OLD											
		Method	Number of Proc	Elapsed time [seco	ond]						
HIV-P				Monomer SCF	Monomer MP2	Monomer (Total)	Dimer ES	Dimer SCF	Dimer MP2	Dimer (Total)	FMO (Total)
(198 res)	点線	Conv-MP2									
(1 water											
and											
1 ligand)			36	2161.8	84.2	2354.0	1079.4	1748.8	3794.4	7198.5	9552.5
	実線	Full-CD									
		(HF & MP2)								-	
			36	1439.6	20.6	1568.2	1078.9	1704.4	501.9	3315.3	4883.5
	破線	IC-CD									
		(HF & MP2)									
		()									
			36	938.1	6.3	1051.8	1077.0	882.8	243.1	2244.9	3296.7

### FX-100 @ JAXA

32スレッド並列はメリット無(コードの問題) 8スレッド並列で72プロセスの方がベター

21

FX100		(n-p-t)=(node-prod	cess-thread)								
		Method	Number of Proc	Elapsed time [secor	nd]						
HIV-P			(n-p-t)	Monomer SCF	Monomer MP2	Monomer (Total)	Dimer ES	Dimer SCF	Dimer MP2	Dimer (Total)	FMO (Total)
(198 res)	点線	Conv-MP2									
(1 water			36-36-16	733.1	42.2	853.3	1559.0	1184.2	2132.0	5093.1	5946.5
and			36-36-32	446.6	25.1	549.8	1227.1	679.5	1593.9	3571.4	4121.2
1 ligand)			72-72-8	818.9	57.8	916.2	1147.2	1231.1	1868.0	4458.8	5375.1
	実線	Full-CD									
		(HF & MP2)	36-36-16	825.8	10.1	913.8	1558.3	851.0	192.2	2613.3	3527.1
			36-36-32	572.3	7.7	658.1	1234.0	548.0	161.6	1950.9	2608.9
			72-72-8	831.7	12.7	883.3	1147.9	753.0	135.8	2072.8	2956.1
	破線	1C-CD									
		(HF & MP2)	36-36-16	556.2	3.0	637.1	1559.7	472.6	97.3	2146.4	2783.5
			36-36-32	357.5	2.3	437.8	1225.0	300.7	85.5	1622.8	2060.6
			72-72-8	486.9	3.4	529.8	1145.2	409.0	64.6	1648.4	2178.2

FMO2-MP2タイミング #3 HIV-1 protease (198残基)

2016/3/1

多ノード利用でも実用性は有



### MP3計算の基本アルゴリズム:形式的にN<sup>6</sup>

### スピンを積分した実装式

添字を逆変換

Fock行列的な処理

(EEOと呼ばれる)

23

添字を変換

可能な限りDGEMMを使用

· 2p 2b± ;;ループ内で同样に加理

$$E^{"3h-3p"} = \sum_{ijkabc} \frac{[2(ia, jb) - (ij, ab)][2(kc, ia) - (ka, ic)][2(kc, jb) - (kb, jc)]}{(\varepsilon_i + \varepsilon_k - \varepsilon_a - \varepsilon_c)(\varepsilon_k + \varepsilon_j - \varepsilon_b - \varepsilon_c)} - 3\sum_{ijkabc} \frac{(ij, ab)(ka, ic)(kb, jc)}{(\varepsilon_i + \varepsilon_k - \varepsilon_a - \varepsilon_c)(\varepsilon_k + \varepsilon_j - \varepsilon_b - \varepsilon_c)}$$
$$E^{"4h-2p"} = \sum_{ijklab} \frac{(ia, jb)(ik, jl)[2(ka, lb) - (kb, la)]}{(\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b)(\varepsilon_k + \varepsilon_j - \varepsilon_a - \varepsilon_b)} E^{"2h-4p"} = \sum_{ijabcd} \frac{(ia, jb)(ac, bd)[2(ic, jd) - (id, jc)]}{(\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b)(\varepsilon_i + \varepsilon_j - \varepsilon_c - \varepsilon_d)}$$
$$\mathbf{E}^{\mathbf{E}^{"4h-2p"}} = \sum_{ijklab} \frac{(ia, jb)(ac, bd)[2(ic, jd) - (id, jc)]}{(\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b)(\varepsilon_i + \varepsilon_j - \varepsilon_c - \varepsilon_d)}$$
$$\mathbf{E}^{\mathbf{E}^{"4h-2p"}} = \sum_{ijklab} \frac{(ia, jb)(ac, bd)[2(ic, jd) - (id, jc)]}{(\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b)(\varepsilon_i + \varepsilon_j - \varepsilon_c - \varepsilon_d)}$$

End of loop over s  $t_{ij}^{qs} = \sum_{cd} c_{qc} c_{sd} t_{ij}^{cd}$ Calculate partial MP2 energy ! parallelized for ij Prepare half-backtransformed 1<sup>st</sup>-order amplitude ! parallelized for *ij* Perform EEO processing for "2h-4p" contribution ! parallelized for pq  $X_{ij}^{pr} = \sum_{qs} (pq, rs) t_{ij}^{qs}.$ Calculate partial "2h-4p" energy ! parallelized for ij Loop over *k*-batch ! size depending on available memory Loop over *s* ! parallelized Form (*kc,ld*) ! 1<sup>st</sup>-4<sup>th</sup> quarter transformations  $X_{ij}^{ab} = \sum_{pr} c_{pa} c_{rb} X_{ij}^{pr}$ End of loop over s Calculate partial "4h-2p" and "3h-3p" energies ! parallelized for ij End of loop over *k*-batch 2重(*ijとk*)のバッチ構造 End of loop over *ij*-batch (メモリが少ないとコスト高) Sum up "2h-4p", "4h-2p" and "3h-3p" energies for final MP3 energy

2016/3/1

CDによるMP3計算(4p-2h部分): 形式的にM\*N<sup>5</sup>

### FMO2&3-MP3タイミング #4 Ala9Gly (10残基)

IVI	۲z	2					-					
abinitmp	smp-	opn1-160102										
FX10	#T	Thread=16 / intra	-node OpenMP parallelism		MP2/6-31G*							
NEW		Method		Number	Elapsed time [second	]						
Ala9Gly (10 res)				of Proc	Monomer SCF	Monomer MP2	Monomer (Total)	Dimer ES	Dimer SCF	Dimer MP2	Dimer (Total)	FMO (Total)
		Regular-MP2	Energy calc	1	763.4	48.5	815.4	9.0	298.8	609.2	982.1	1797.5
				2	441.8	26.4	470.1	4.6	151.0	302.6	490.5	960.6
				5	214.9	12.8	228.5	1.9	61.1	117.7	202.5	430.9
				10	92.2	7.8	100.6	1.0	31.0	59.5	103.0	203.6
		Full-CD-MP2	Energy calc	1	300.6	8.5	312.6	9.0	254.4	49.5	317.6	630.2
				2	157.0	4.3	163.2	4.6	127.5	24.4	158.7	321.9
				5	68.2	1.9	70.9	1.9	51.3	9.9	65.2	136.2
				10	55.3	1.7	57.6	1.1	25.9	5.2	33.0	90.6
		1CCD-NP2	Energy calc	1	127.4	2.4	133.2	9.0	110.6	23.5	147.7	280.9
				2	66.1	1.2	69.2	4.6	55.1	11.6	74.6	143.8
				5	28.8	0.5	30.2	1.9	21.9	4.6	30.8	61.0
				10	24.2	0.5	25.3	1.0	10.9	24	15.9	41.2

**FX-10 @ RIST** 

### MP3

#### Full CDではMP2比で10倍となる残念な結果に…(通常MP3が上手く出来過ぎている?)

X10	#Thread=16 / intra-node OpenMP parallelisr	1	MP3/6-31G*							
IEW	Method	Number	Elapsed time [second]							
la9Gly		of	Monomer SCE	Monomer MP2	Monomer (Total)	Dimer FS	Dimer SCE	Dimer MP3	Dimer (Total)	FMO (Total)
0 res)		Proc	Woholiker Ser	Monomer wir 2	Wohomer (Total)	Diner Lo	Diner Ser	Diner wir 5	Diner (Total)	Time (Total)
	Regular-MP3 Energy calc	1	761.4	87.2	852.1	9.0	299.7	2650.8	3024.4	3876.5
		2	442.1	43.8	487.8	4.6	151.0	1336.8	1524.9	2012.7
		5	215.0	19.9	235.7	1.9	59.1	426.2	648.7	884.4
		10	91.9	19.1	111.7	1.0	27.3	189.9	427.5	539.2
	Full-CD-MP3 Energy calc	1	300.6	74.9	379.0	9.0	254.7	6335.7	6604.1	6983.0
		2	157.3	43.5	202.7	4.6	127.9	3079.7	3388.5	3591.2
		5	67.9	24.5	93.2	1.9	47.0	992.5	1480.7	1573.9
		10	55.0	24.6	80.2	1.0	21.1	412.8	875.7	955.9
	1CCD-NP3 Energy calc	1	127.1	33.1	163.6	9.1	110.6	2982.8	3107.2	3270.7
		2	66.2	19.2	87.3	4.6	54.8	1448.0	1602.3	1689.6
		5	29.2	10.9	41.0	1.9	20.1	459.6	697.5	738.5
		10	24.2	10.9	35.7	1.0	9.0	193.4	420.3	456.0
2	016/3/1									2

# FMO2-MP2タイミング #4 Ala9Gly (10残基)

スケ	τ—J	レ	41	1	良	好
~ ` '		-		σ,		~1



# FMO2-MP3タイミング #4 Ala9Gly (10残基)

残念ながらCDのメリットが見えない…



ABINIT-MPの性能評価のまとめ、これからの課題

#### 現状

・最も利用度の高いFMO2-MP2/6-31G\*計算は及第の性能 (「京」、FX)

- → FX-100では実タンパク質のMP2ジョブもルーチンに実行可能
- → 複数の計算処理から構成されるFMOジョブでも要所を押さえることが肝要
- ・MP3は精度向上の上では有用(MP2.5スケーリング)
  - → 通常積分の場合でもDGEMMによる行列積が支配的、閾値篩も有効
  - → CDは現状ではコスト高、展開長MをほぼNと出来るRIの導入がマスト

#### ■ 今後

2016/3/1

- ・FXや「京」などを利用してベンチと改良作業を続ける
  - → 大型のタンパク質の実行テストの必要性
  - → エネルギー微分など他の計算機能の評価

#### ・ABINIT-MPのOpenシリーズへの移行と性能向上(FS2020プロジェクト関連)

- → 2016年度内にCDからRIに移行
- →「京」では20-30%効率を1000-2000ノードで担保したい
- → DFT法などの新規計算機能の導入
- ・「京」などでのcapacity computingへの対応
  - → いわゆるパラメータスタディや統計ジョブの扱い (HPC/PFと連携)
  - → ジョブの自動投入/回収のシステム
- ・ABINIT-MPソース共有コンソーシアムの立上げ(2015年度末)
- → 産官学の構成、産業界への普及を促進

27