JÜLICH
FORSCHUNGSZENTRUM

The Single Parameter Fast Multipole Method

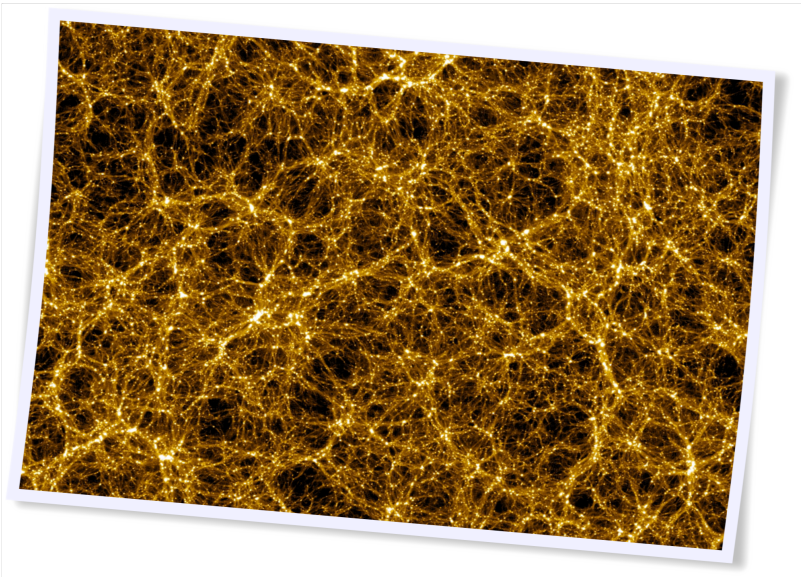
A Coulomb solver for the masses.

September 4, 2012 | Ivo Kabadshow Holger Dachsel |

JÜLICH
FORSCHUNGSZENTRUM

Real World Problems

Astrophysics: Structure Formation



Mitglied der Helmholtz-Gemeinschaft

JÜLICH FORSCHUNGSZENTRUM

Real World Problems

Astrophysics: Galaxy Formation

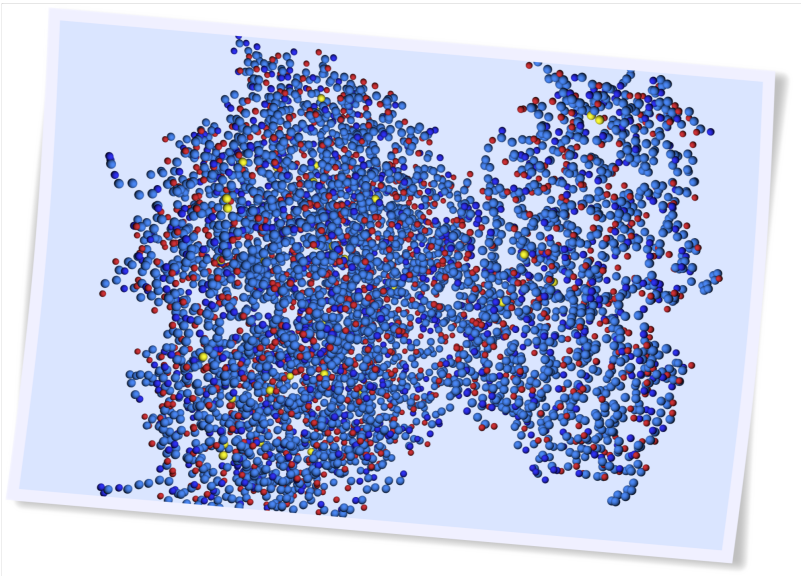
JÜLICH FORSCHUNGSZENTRUM

Real World Problems

Laser-Plasma Physics: Coulomb Explosion

Real World Problems

Softmatter Physics: Biological Systems



The Coulomb Problem

Classical Force Calculation

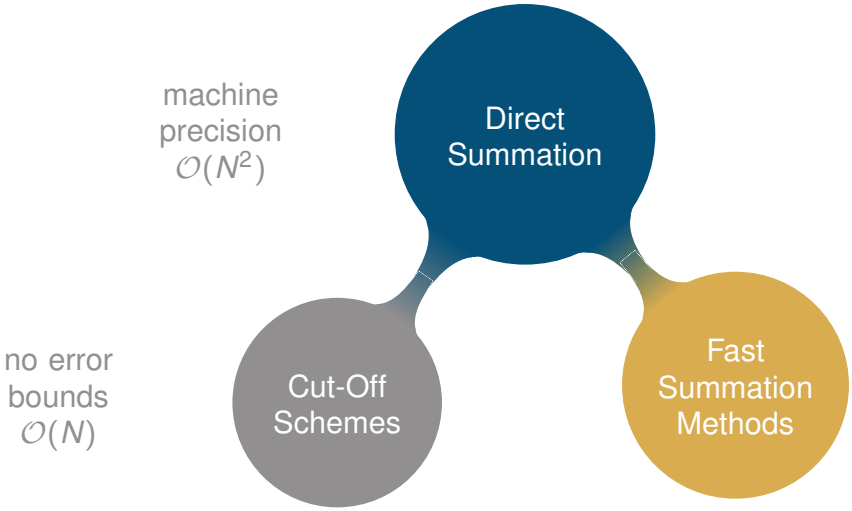
Direct Summation $\mathcal{O}(N^2)$

$$\mathbf{F}(\mathbf{r}_i) = q_i \sum_{j=1}^{N}{}' \frac{q_j}{r_{ij}^3}\mathbf{r}_{ij}$$

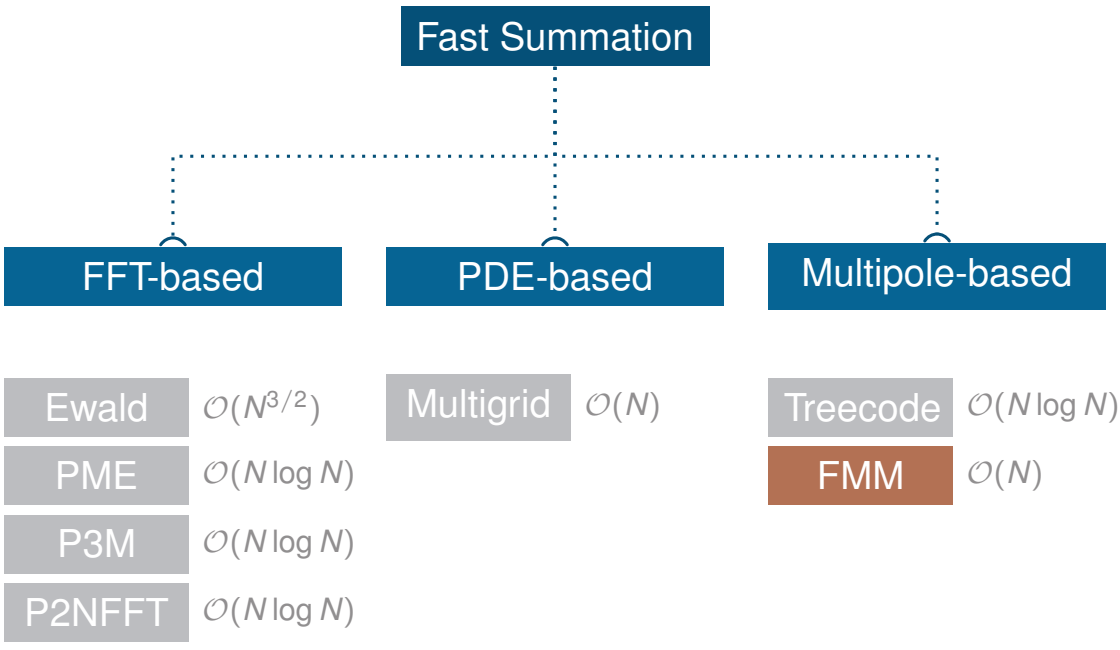
In a Simulation

- 95% – 99% of time spend in force calculation
- 1% – 5% of time spend in integration and post processing

Alternatives To Direct Summation



Classification of Fast Summation Methods



The Coulomb Problem

Classical Scheme

Direct Summation $\mathcal{O}(N^2)$

$$E_C = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \frac{q_i q_j}{r_{ij}}$$

Fast Summation Algorithms

Fast Multipole Method $\mathcal{O}(N)$

$$E_C = \sum_{\text{level}} \sum_{\text{A}} \sum_{\text{B(A)}} \sum_{l=0}^{p} \sum_{m=-l}^{l} \sum_{j=0}^{p} \sum_{k=-j}^{j} (-1)^j \omega_{lm}^{\text{A}}(\mathbf{a}) M2L(\mathbf{R}) \omega_{jk}^{\text{B}}(\mathbf{b})$$

Idea: factorization of inverse distance

The Fast Multipole Method

There's no such thing as a free lunch

Pro

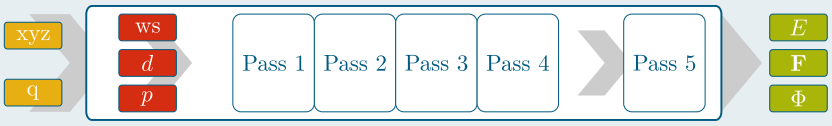
- Complexity reduces to $\mathcal{O}(N)$
- Computation time $t = f(\epsilon)$ is a function of requested precision.

Con

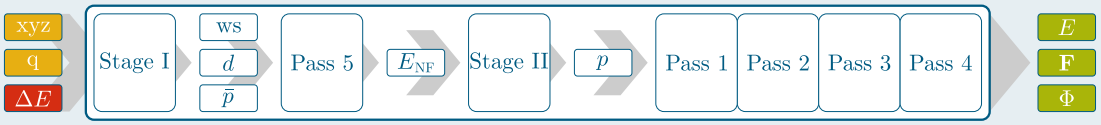
- Three new parameters are introduced
  - Tree depth (d)
  - Separation Criterion (ws)
  - Number of poles (p)
- Precision dependency $\epsilon = g$(ws,d,p) is not known explicitly

Towards a Single-Parameter FMM

Original FMM Workflow

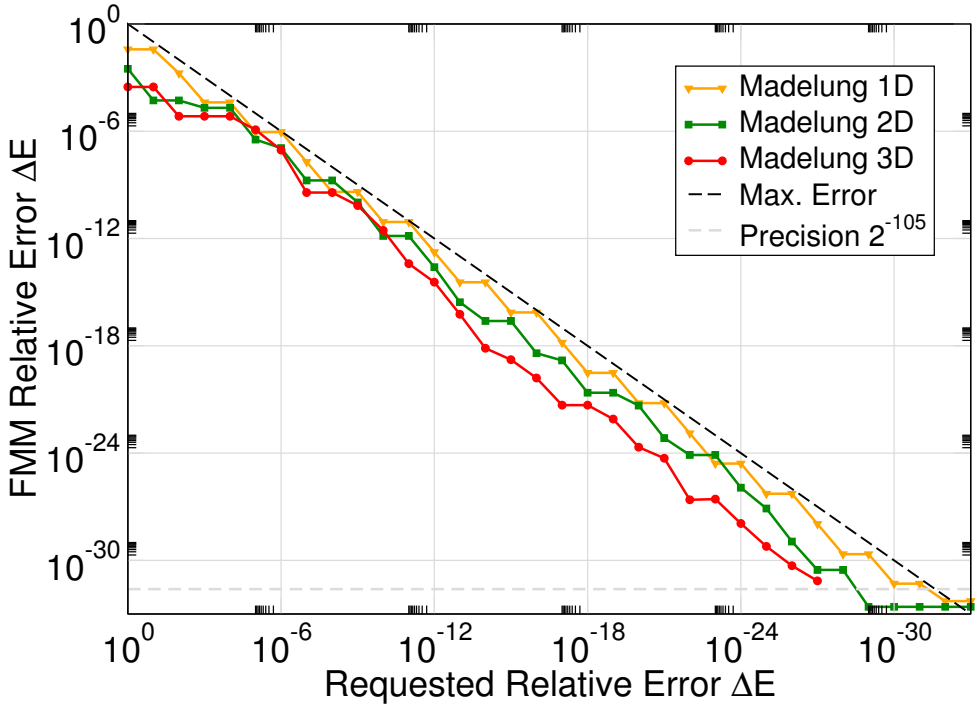


Enhanced FMM Workflow with Error Control

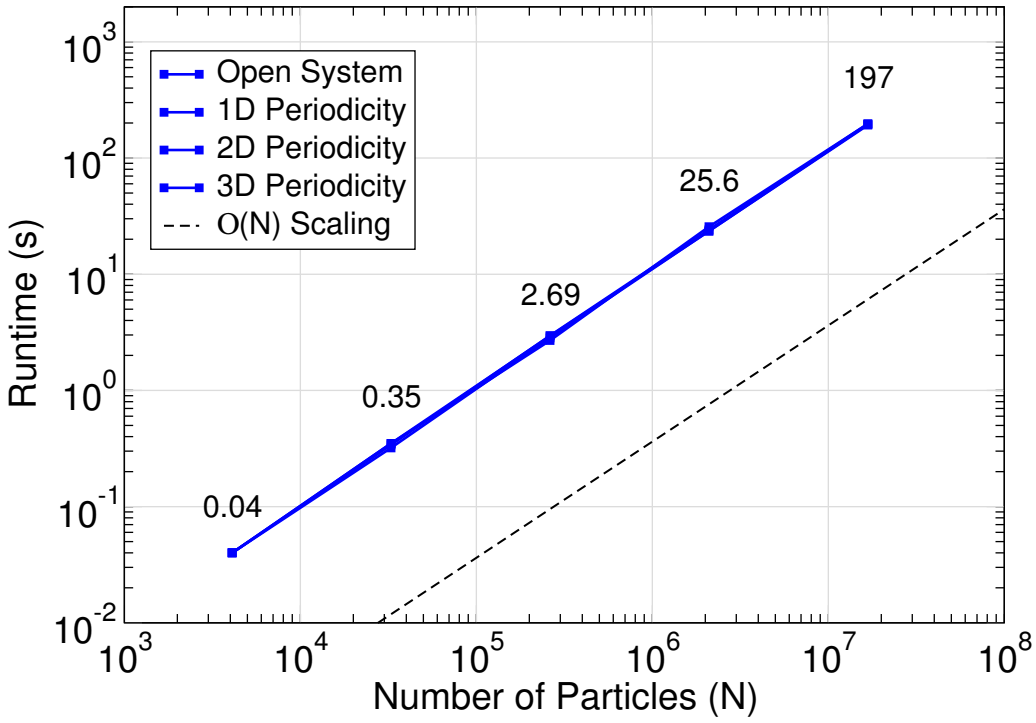


Quadruple Precision Check

1D, 2D and 3D Periodicity

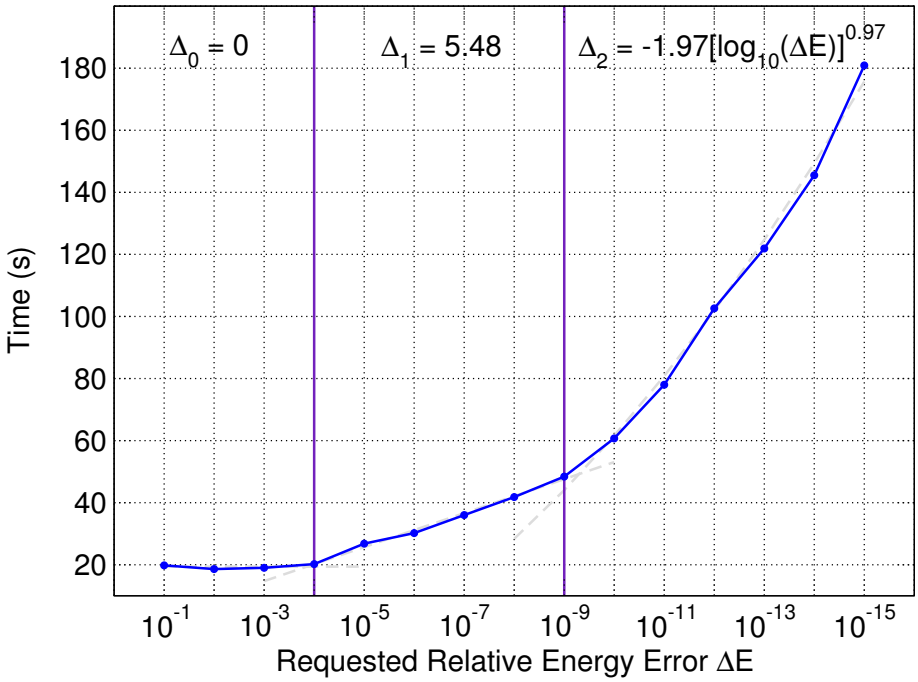


Scaling with N, $\Delta E_{rel} = 10^{-5}$



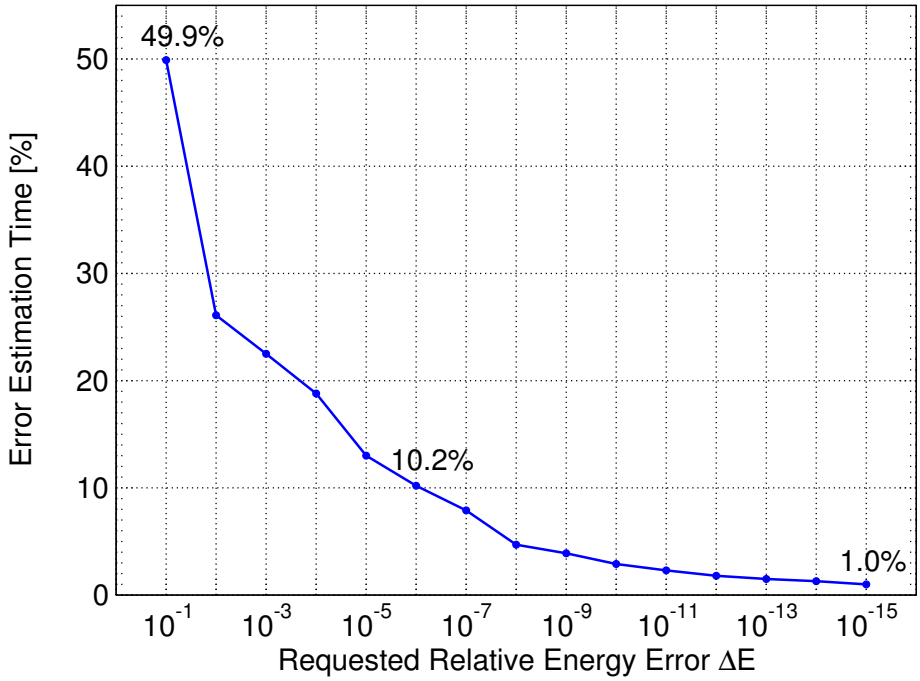
Precision Scaling

2.097.152 Particles - Total Computation Time



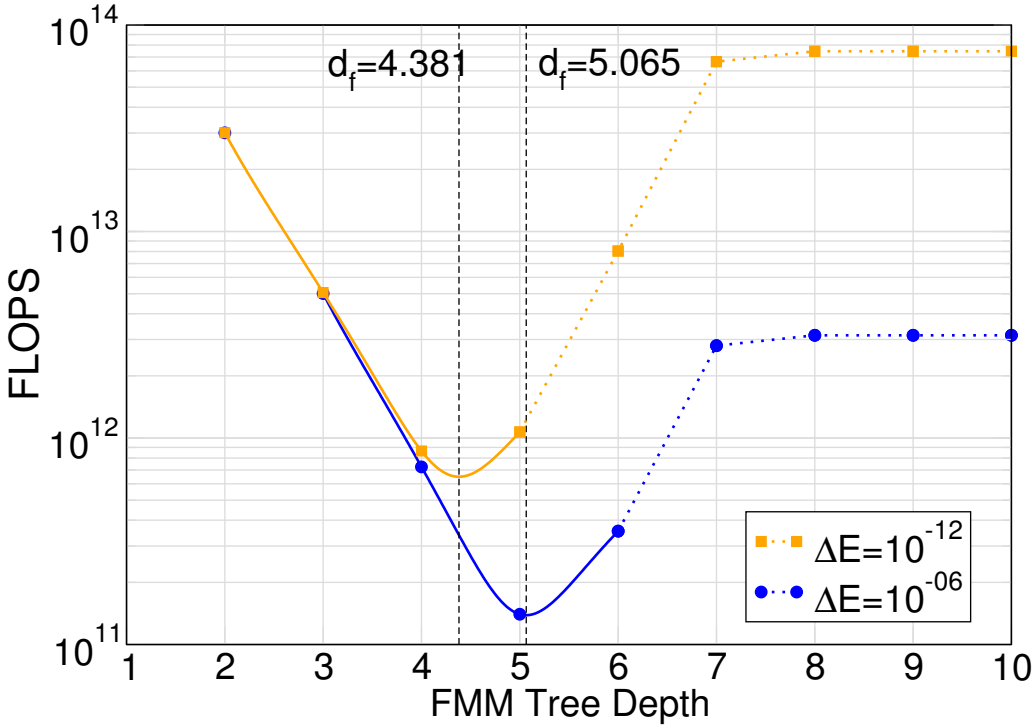
Tuning Costs

2.097.152 Particles - Error Estimation Scheme Runtime



FMM Runtime Optimization

Trade `DIV` and `SQRT` for `ADD` and `MULT`



Going Parallel

Limiting Details

- Fast summation methods trade memory for speed
  → Additional memory again limits system size
  → Critical on distributed memory systems like BG/P

Improve Weak Scaling

- Reduce algorithmic overhead
  - Don't provide storage for empty regions of space
  - Don't store redundant data (neighbor lists etc.)
- Reduce parallelization overhead
  - Latency: overlap communication with computation
  - Memory: avoid $\mathcal{O}(p)$ data structures

Parallelization Properties

Use . . .

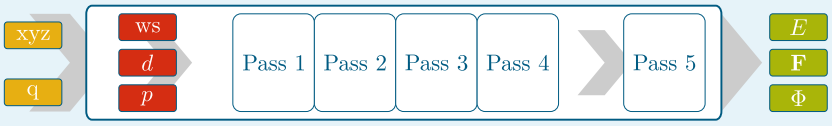
- data locality
- fastest one-sided memory access (put) via OSPRI/ARMCI
- minimal communication functionality
- distribute load via space-filling curve
- MPI (inplace) for collectives only (allgather,allreduce, barrier)

Things to Avoid

- Avoid domain decomposition to distribute work
- Avoid accumulate and non-contiguous send operations
- Avoid global counters

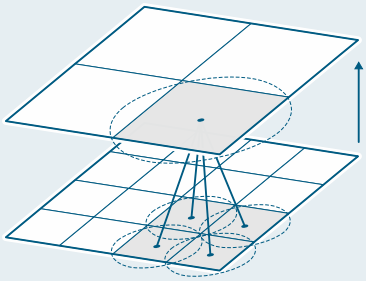
FMM Workflow

FMM Workflow



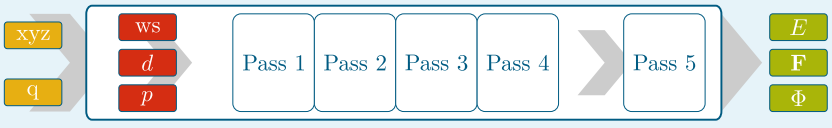
Pass 1

- Setup FMM tree and expand multipoles
- Shift multipoles to root node (M2M)



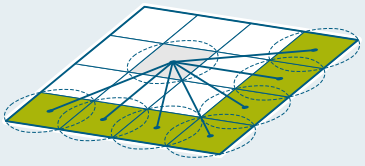
FMM Workflow

FMM Workflow



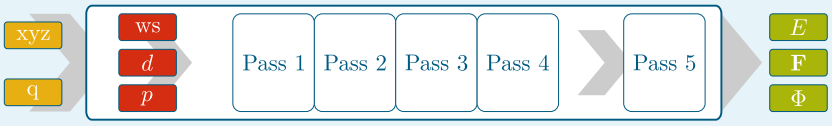
Pass 2

- Translate multipole moments into Taylor coefficients (M2L) at each level of the tree



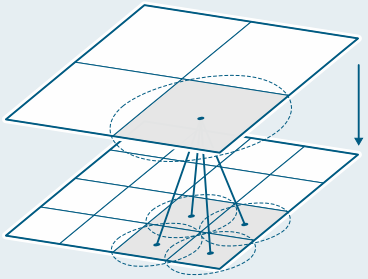
FMM Workflow

FMM Workflow



Pass 3

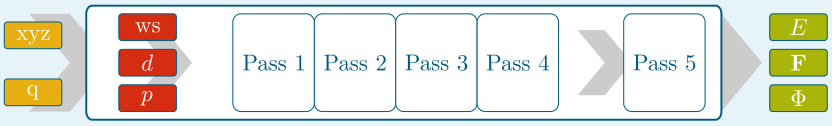
- Shift Taylor coefficients to the leaf nodes (L2L)



JÜLICH FORSCHUNGSZENTRUM

FMM Workflow

FMM Workflow

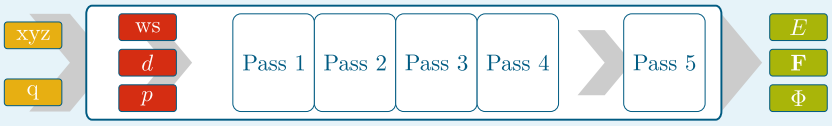


Pass 4

- Computation of the far field energy, forces, potentials

FMM Workflow

FMM Workflow



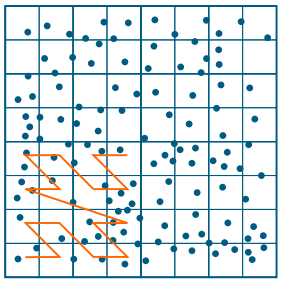
Pass 5

- Computation of the near field energy, forces, potentials

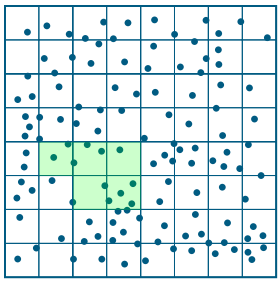
Data Layout

Different Local Domains

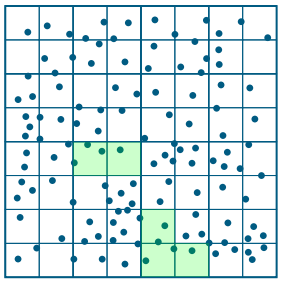
- All boxes are sorted along a space-filling curve
- Static load balancing is essential



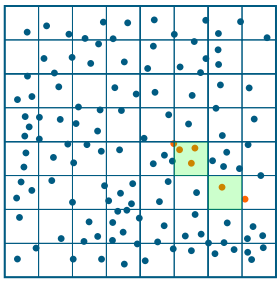
All boxes are z-ordered



Compact Domain



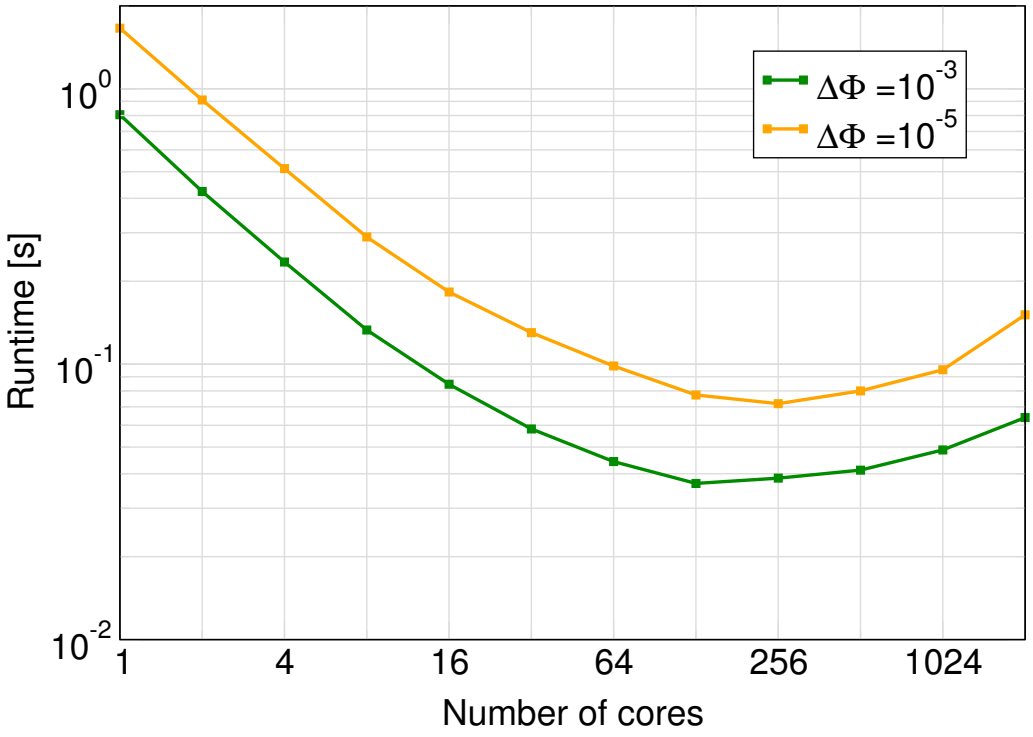
Two Domains



Splitted particles/multipoles

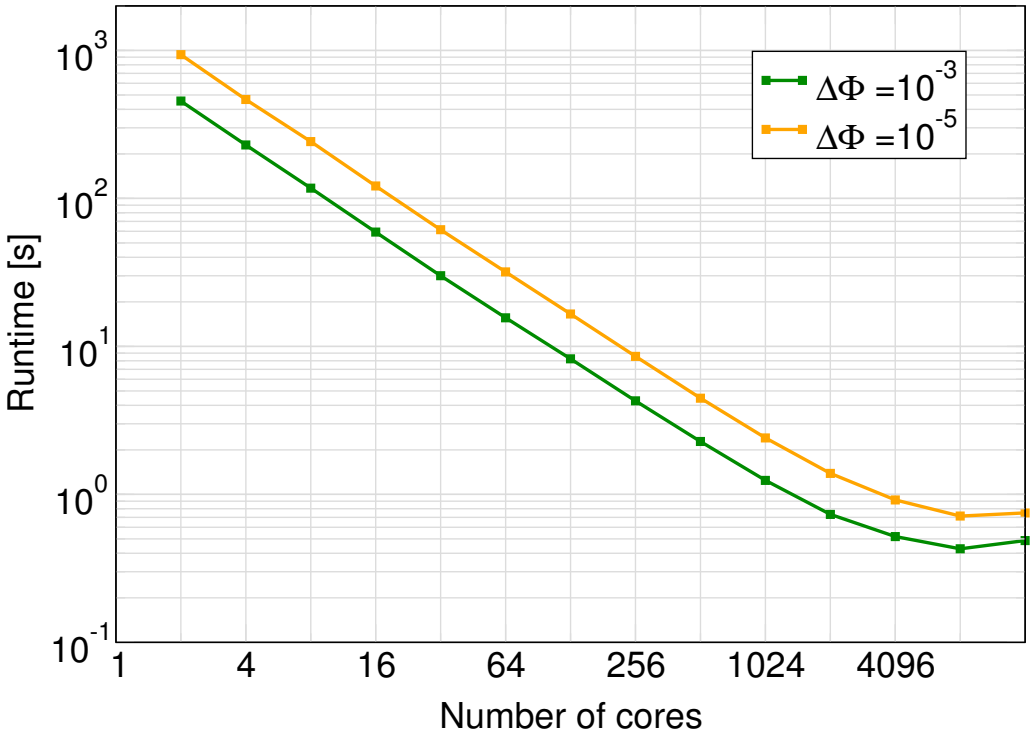
Strong Scaling On Jugene

FMM with 3D Periodic Bondaries, 8.100 Particles



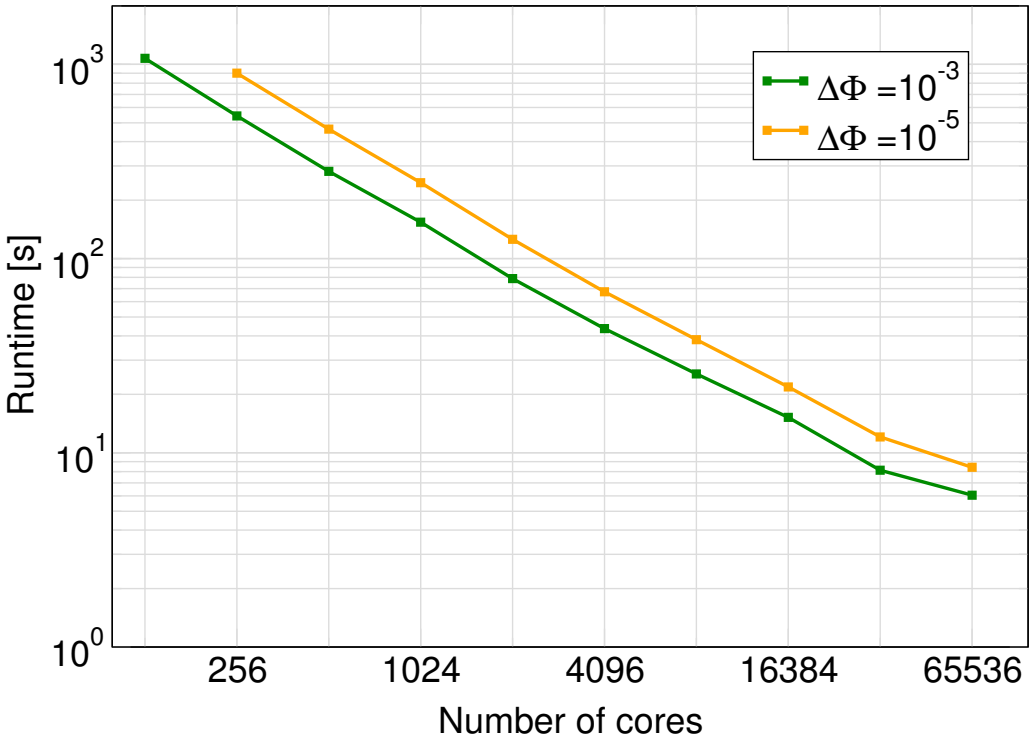
Strong Scaling On Jugene

FMM with 3D Periodic Bondaries, 9.830.400 Particles



Strong Scaling On Jugene

FMM with 3D Periodic Bondaries, 1.012.500.000 Particles



Lower Memory Bound

How many memory would a direct calculation need?

Input (Floats)	Size
Coordinates `xyz(3,n)`	12 Bytes
Charges/Masses `q(n)`	4 Bytes

Output (Floats)	Size
Gradient/Forces/Field `grad(3,n)`	12 Bytes

Total Memory per Particle	Size
	28 Bytes

How Many Particle Can Fit On The Full BGP?

Upper Bound With Direct Summation $\mathcal{O}(N^2)$

- Available memory $\approx 132TB$ (472MB per core)
- 5.2 trillion ($5.2 \cdot 10^{12}$) particles would fit
  $\rightarrow$ 32.000 years to compute all forces

Direct Summation Introduces Errors

- Machine epsilon $\epsilon = 5.96 \times 10^{-8}$ (single precision)
- Round-off error accumulation $\mathcal{O}(\sqrt{\epsilon N^2})$
  $\rightarrow$ No significant figure will remain

JÜLICH FORSCHUNGSZENTRUM

Pushing The Limits Beyond 1.000.000.000.000

Mitglied der Helmholtz-Gemeinschaft

JÜLICH FORSCHUNGSZENTRUM

Passing The One-Trillion Particle Limit

System Characteristics

- 1.030.607.060.301 particles
- 32768 nodes (SMP mode, 32768 cores)

Results

- 3285s runtime, unsorted data (9576 particles/second/core)
- 2203s runtime, presorted data (14276 particles/second/core)

JÜLICH FORSCHUNGSZENTRUM

Passing The Two-Trillion Particle Limit

System Characteristics

- 2.010.394.559.061 particles
- 73728 nodes (VN mode, 294912 cores)

Results

- 2288s runtime, unsorted data (2279 particles/second/core)
- 530s runtime, presorted data (12862 particles/second/core)

Mitglied der Helmholtz-Gemeinschaft

JÜLICH FORSCHUNGSZENTRUM

Passing The Three-Trillion Particle Limit

System Characteristics

- 3.011.561.968.121 particles
- 73728 nodes (VN mode, 294912 cores)
- $0.254 \cdot 10^{27}$ FLOPS direct

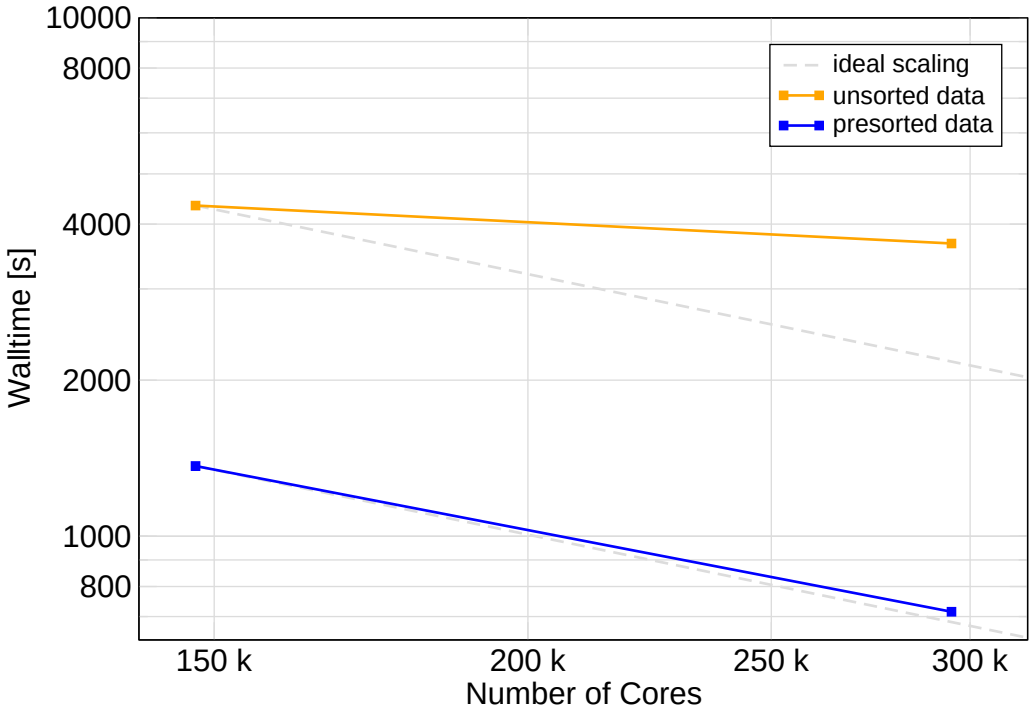
Results

- (38.00) 44.16 Bytes/particle (16.2% parallelization overhead)
- 3812s runtime, unsorted data (2755 particles/second/core)
- 715s runtime, presorted data (14687 particles/second/core)
  $\approx$ 4.3 billion particles per second
- $0.410 \cdot 10^{17}$ FLOPS via FMM

JÜLICH FORSCHUNGSZENTRUM

Passing the Three-Trillion Particle Limit

Open System, 3.011.561.968.121 Particles, $\Delta E = 10^{-2}$, 44.16 Bytes/particle



Conclusion & Outlook

Take Away Message

1. FMMs can offer on-the-fly, single parameter
   - error control
   - runtime minimization
2. FMMs can be implemented
   - with a small memory footprint
   - with a small parallel overhead

FMM is part of Scafacos

- FMM implementation is available as part of Scafacos
- FMM will be coupled with GROMACS and DLPOLY