FD4: A Framework for Highly Scalable Dynamic Load Balancing and Model Coupling

Symposium on HPC and Data-Intensive Applications in Earth Sciences

13 Nov 2014, Trieste, Italy

Matthias Lieber (matthias.lieber@tu-dresden.de)

Center for Information Services and High Performance Computing (ZIH)
Technische Universität Dresden, Germany
"Climate models now include more cloud and aerosol processes, and their interactions, than at the time of the AR4, but there remains low confidence in the representation and quantification of these processes in models."

Motivation: Spectral Bin Cloud Microphysics Schemes

- Bin discretization of cloud particle size distribution
- Allows more detailed modeling of interaction between aerosols, clouds, and precipitation
- Computationally too expensive for forecast
- Only used for process studies up to now

Widely used bulk models

Spectral bin microphysics

Motivation: Tropical Cyclone Forecast with SBM?

Horizontal grid: 1000 x 1000

Real-time forecast requires ~10,000 CPU cores

Model systems must be tuned for efficient usage of large machines
Outline

- Bottleneck Analysis
- Concept of Load-balanced Coupling
- FD4's Features
- Benchmarks
- Conclusion
FD4 Motivation: COSMO-SPECS Performance

- COSMO-SPECS: Atmospheric model COSMO extended with highly detailed cloud microphysics model SPECS

Small 3D case with 64x64x48 grid

![Small 3D case with 64x64x48 grid](image)

<table>
<thead>
<tr>
<th>Number of CPU Cores</th>
<th>Runtime x Cores (h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td></td>
</tr>
<tr>
<td>256</td>
<td></td>
</tr>
<tr>
<td>512</td>
<td></td>
</tr>
<tr>
<td>1024</td>
<td></td>
</tr>
</tbody>
</table>

- Growing cumulus cloud

Analysis: Common Parallelization Scheme

- 3D domain partitioned into rectangular boxes
- 2D decomposition (horizontal dimensions)
- Regular communication with 4 direct neighbors required (periodic boundary conditions)
- Based on MPI (Message Passing Interface)
Analysis: Load Imbalance due to Microphysics

- SPECS computing time varies strongly depending on the range of the particle size distribution and presence of frozen particles.
- Leads to load imbalances between partitions.

Solution: Apply dynamic load balancing.
Surface-to-volume-ratio of partitions grows with number of partitions, in theory (best case):

- 2D decomposition: \( A^{2D}(P) = 4 \ G^{2/3} \ P^{1/2} \sim P^{1/2} \)
- 3D decomposition: \( A^{3D}(P) = 6 \ G^{2/3} \ P^{1/3} \sim P^{1/3} \)

Analysis: Increasing Communication Volume

Solution: Apply 3D decomposition
Outline

- Bottleneck Analysis
- **Concept of Load-balanced Coupling**
- FD4's Features
- Benchmarks
- Conclusion
Concept of Load-Balanced Coupling

Atmospheric Model & Spectral Bin Microphysics

2D Decomposition

Static Partitioning
Concept of Load-Balanced Coupling

Atmospheric Model & Spectral Bin Microphysics

High Scalability
\[ P \approx 10,000 \]

Model Coupling

2D Decomposition

Static Partitioning

Block-based 3D Decomposition

Dynamic Load Balancing

Optimized Data Structures

Concept of Load-Balanced Coupling

Implemented as independent framework FD4

FD4: Four-Dimensional Distributed Dynamic Data structures

High Scalability $P \approx 10,000$

Model Coupling

Block-based 3D Decomposition

Dynamic Load Balancing

Optimized Data Structures

Outline

- Bottleneck Analysis
- Concept of Load-balanced Coupling
- **FD4's Features**
- Benchmarks
- Conclusion
FD4: Dynamic Load Balancing

- 3D block decomposition of rectangular grid
- Space-filling curve (SFC) partitioning to assign blocks to ranks
- SFC reduces 3D partitioning problem to 1D
- High locality of SFC leads to moderate comm. costs
- Developed a highly scalable, hierarchical method for high-quality 1D partitioning of the SFC-indexed blocks
FD4: Model Coupling

- Data exchange between FD4 based model and an external model
  - E.g. weather or CFD model
  - Transfer in both directions

- FD4 computes partition overlaps after each repartitioning of FD4 grid
  - Highly scalable algorithm

- No grid transformation / interpolation
  - External model must provide data matching the FD4 grid

- “Sequential” coupling only
  - Both models run alternately on same set of MPI ranks
FD4: 4th Dimension

- Extra, non-spatial dimension of grid variables, e.g.
  - Size resolving models
  - Array of gas phase tracers

- FD4 is optimized for a large 4th dimension

- COSMO-SPECS requires
  $2 \times 11 \times 66 \sim 1500$ values
**FD4: Adaptive Block Mode**

- Grid allocation adapts to spatial structure of simulated problem
  - Save memory in case data and computations are required for a subset only
- For multiphase problems like drops, clouds, flame fronts
- FD4 ensures existence of all blocks required for correct stencil operations
Outline

- Bottleneck Analysis
- Concept of Load-balanced Coupling
- FD4's Features
- **Benchmarks**
- Conclusion
Benchmarks: COSMO-SPECS Performance Comparison

- Almost 3 times faster at 1024 CPU cores
- Load balancing & coupling scale well, but can we reach > 10,000 processes?
Benchmarks: Scalability on Blue Gene/Q

- Grid size: 1024 x 1024 x 48 grid cells, > 3M blocks
- 256k: 30 min forecast in <5min (w/o init and I/O)
- Runs on Blue Gene/Q with up to 262,144 MPI ranks
- 14 x speed-up from 16k to 256k

COSMO-SPECS+FD4

Benchmarks: Load Balancing & Coupling Scalability

- Grid size: 1024 x 1024 x 48 grid cells, > 3M blocks
- Load balancing scales comparatively very well
- Coupling scales nearly perfect

Dynamic Load Balancing Runtime %

- Migration
- Partitioning calculation
- Synchronization

Coupling Runtime %

- Coupling Communication
- Partition Matching

Benchmarks: 1D Partitioning Comparison on Blue Gene/Q

- ExactBS: exact method, but slow and serial
- H2: fast heuristic, but may result in poor load balance
- HIER*: hierarchical algorithm implemented in FD4, achieves nearly optimal load balance

<table>
<thead>
<tr>
<th>Process</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>27208</td>
<td>313.860578s +0.25 s +0.50 s +0.75 s +1.00 s +1.25 s +1.50 s +1.75 s</td>
</tr>
<tr>
<td>27216</td>
<td></td>
</tr>
<tr>
<td>27224</td>
<td></td>
</tr>
<tr>
<td>27232</td>
<td></td>
</tr>
<tr>
<td>27240</td>
<td></td>
</tr>
<tr>
<td>27248</td>
<td></td>
</tr>
<tr>
<td>27256</td>
<td></td>
</tr>
<tr>
<td>27264</td>
<td></td>
</tr>
<tr>
<td>27272</td>
<td></td>
</tr>
<tr>
<td>27280</td>
<td></td>
</tr>
<tr>
<td>27288</td>
<td></td>
</tr>
<tr>
<td>27296</td>
<td></td>
</tr>
<tr>
<td>27304</td>
<td></td>
</tr>
<tr>
<td>27312</td>
<td></td>
</tr>
<tr>
<td>27320</td>
<td></td>
</tr>
<tr>
<td>27328</td>
<td></td>
</tr>
<tr>
<td>27336</td>
<td></td>
</tr>
<tr>
<td>27344</td>
<td></td>
</tr>
<tr>
<td>27352</td>
<td></td>
</tr>
<tr>
<td>27360</td>
<td></td>
</tr>
<tr>
<td>27368</td>
<td></td>
</tr>
<tr>
<td>27376</td>
<td></td>
</tr>
<tr>
<td>27384</td>
<td></td>
</tr>
<tr>
<td>27392</td>
<td></td>
</tr>
<tr>
<td>27400</td>
<td></td>
</tr>
<tr>
<td>27408</td>
<td></td>
</tr>
<tr>
<td>27416</td>
<td></td>
</tr>
<tr>
<td>27424</td>
<td></td>
</tr>
<tr>
<td>27432</td>
<td></td>
</tr>
<tr>
<td>27440</td>
<td></td>
</tr>
<tr>
<td>27448</td>
<td></td>
</tr>
<tr>
<td>27456</td>
<td></td>
</tr>
<tr>
<td>27464</td>
<td></td>
</tr>
<tr>
<td>27472</td>
<td></td>
</tr>
<tr>
<td>27480</td>
<td></td>
</tr>
<tr>
<td>27488</td>
<td></td>
</tr>
<tr>
<td>27496</td>
<td></td>
</tr>
<tr>
<td>27504</td>
<td></td>
</tr>
<tr>
<td>27512</td>
<td></td>
</tr>
<tr>
<td>27520</td>
<td></td>
</tr>
<tr>
<td>27528</td>
<td></td>
</tr>
</tbody>
</table>
Outline

- Bottleneck Analysis
- Concept of Load-balanced Coupling
- FD4's Features
- Benchmarks
- Conclusion
FD4 provides for simulation models
- Parallelization of numerical grid
- Communication between neighbor partitions
- Dynamic load balancing
- Model coupling
- High scalability

Initially developed for atmospheric modeling, but generally applicable

FD4 is available as open source software
- Fortran 95, MPI-2, NetCDF
- Tested on many different HPC systems

Conclusions

FD4 website:
http://wwwpub.zih.tu-dresden.de/~mlieber/fd4


Acknowledgments

Verena Grützun, Ralf Wolke, Oswald Knoth, Martin Simmel, René Widera, Matthias Jurenz, Matthias Müller, Wolfgang E. Nagel

Thank you very much for your attention!

Funding

www.tropos.de
www.cosmo-model.org
picongpu.hzdr.de
A large number of small blocks are good for performance:

- Size-resolved approach / ~1000 variables per grid cell: Only small blocks do not exceed processor cache
- Load balancing: \#blocks > \#partitions to enable fine-grained balancing

Additional memory costs for a boundary of ghost cells

- Too high for small blocks!

Add ghost blocks at the partition borders only
Space-filling curve (SFC) partitioning widely used
  - nD space is mapped to 1D by SFC
  - Mapping is fast and has high locality
  - Migration typically between neighbor ranks
1D partitioning is core problem of SFC partitioning
  - Decomposes task chain into consecutive parts
Two classes of existing 1D partitioning algorithms:
  - Heuristics: fast, parallel, no optimal solution
  - Exact methods: slow, serial, but optimal


Sequential vs. Concurrent Model Coupling

Both models run alternately on same set of MPI ranks
Allows tight coupling (data dependencies)
Avoids load imbalances between models

MPI ranks are split into groups
Loose coupling, codes may be separate
Scales to higher total number of ranks
Large scale applications require a fully parallel method, i.e. without gathering all task weights

- Run parallel H2 to create $G < P$ coarse partitions:
  
  **Orig part.**
  
<table>
<thead>
<tr>
<th>P0</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>P6</th>
<th>P7</th>
</tr>
</thead>
</table>
  
  **Parallel Heuristic H2**
  
<table>
<thead>
<tr>
<th>P0</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>P6</th>
<th>P7</th>
</tr>
</thead>
</table>
  
  **Coarse part.**
  
<table>
<thead>
<tr>
<th>P0 / P1</th>
<th>P2 / P3</th>
<th>P4 / P5</th>
<th>P6 / P7</th>
</tr>
</thead>
</table>
  
  - Run $G$ independent instances of exact QBS* ($q=1.0$) to create final partitions within each group:

  **QBS**
  
  |----|----|----|

- Parameter $G$ allows trade-off between scalability (high $G \rightarrow$ heuristic dominates) and load balance (small $G \rightarrow$ exact method dominates)

H2 nearly optimal if $W_{\text{max}} << W_N / P$: Miguet, Pierson, *Heuristics for 1D rectilinear partitioning as a low cost and high quality answer to dynamic load balancing*, LNCS, vol. 1225, 1997, pp. 550-564.
FD4: Implementation

- Implemented in Fortran 95
- MPI-based parallelization
- Open Source Software
- www.tu-dresden.de/zih/clouds

```fortran
! MPI initialization
call MPI_Init(err)
call MPI_Comm_rank(MPI_COMM_WORLD, rank, err)
call MPI_Comm_size(MPI_COMM_WORLD, nproc, err)
! create the domain and allocate memory
call fd4_domain_create(domain, nb, size, &
  vartab, ng, peri, MPI_COMM_WORLD, err)
call fd4_util_allocate_all_blocks(domain, err)
! initialize ghost communication
call fd4_ghostcomm_create(ghostcomm, domain, &
  4, vars, steps, err)
! loop over time steps
do timestep=1,nsteps
  ! exchange ghosts
  call fd4_ghostcomm_exch(ghostcomm, err)
  ! loop over local blocks
  call fd4_iter_init(domain, iter)
do while(associated(iter%cur))
    ! do some computations
    call compute_block(iter)
call fd4_iter_next(iter)
end do
! dynamic load balancing
  call fd4_balance_readjust(domain, err)
end do
```
Benchmarks: COSMO-SPECS Performance Comparison

Process 0
Process 2
Process 4
Process 6
Process 8
Process 10
Process 12
Process 14
Process 16
Process 18
Process 20
Process 22
Process 24
Process 26
Process 28
Process 30
Process 32
Process 34
Process 36
Process 38
Process 40
Process 42
Process 44
Process 46
Process 48
Process 50
Process 52
Process 54
Process 56
Process 58
Process 60
Process 62

COSMO-SPECS+FD4

238.374357 s +1 s +2 s +3 s +4 s

489.28388 s +5 s +10 s

Process 0
Process 2
Process 4
Process 6
Process 8
Process 10
Process 12
Process 14
Process 16
Process 18
Process 20
Process 22
Process 24
Process 26
Process 28
Process 30
Process 32
Process 34
Process 36
Process 38
Process 40
Process 42
Process 44
Process 46
Process 48
Process 50
Process 52
Process 54
Process 56
Process 58
Process 60
Process 62

COSMO-SPECS
Cloud simulation, 1 357 824 tasks
System: JUQUEEN, IBM Blue Gene/Q
HIER*, G=64 achieves 99.2% of the optimal load balance at 262 144 processes
**HIER* seen in Vampir (one Group of 256 out of 64Ki)**

1.935252 s  +0.25 ms  +0.50 ms  +0.75 ms  +1.00 ms  +1.25 ms  +1.50 ms  +1.75 ms  +2.00 ms  +2.25 ms

- Process 64244
- Process 64253
- Process 64262
- Process 64272
- Process 64281
- Process 64290
- Process 64299
- Process 64308
- Process 64317
- Process 64326
- Process 64335
- Process 64344
- Process 64353
- Process 64362
- Process 64372
- Process 64382
- Process 64391
- Process 64400
- Process 64409
- Process 64418
- Process 64427
- Process 64436
- Process 64445
- Process 64454
- Process 64463
- Process 64472
- Process 64481
- Process 64490
- Process 64499
- Process 64508
- Process 64517
- Process 64526

**TECHNISCHE UNIVERSITÄT DRESDEN**

**ZIH**

Center for Information Service & High Performance Computing
ExactBS in Action (COSMO-SPECS+FD4)
Figure 3. Influence of the group count $G$ on the hierarchical 1D partitioning algorithm in COSMO-SPECS+FD4 with 65,536 processes on BlueGene/Q. The exact method and the heuristic are included as reference.
Scalable Coupling: Meta Data Subdomains

- “Handshaking” – Identifying partition overlaps between the coupled models – turned out to be the main scalability bottleneck
- Solved with spatially indexed data structure for coupling meta data in FD4
- Time for locating overlap candidates does not depend on number of ranks

(a) Required meta data subdomains.

(b) Contained coupled partitions.