The Icosahedral Nonhydrostatic (ICON) Model

Scalability on Massively Parallel Computer Architectures

Florian Prill, DWD + the ICON team

15th ECMWF Workshop on HPC in Meteorology
October 2, 2012

ICON = ICOsahedral Nonhydrostatic model

Global circulation model for atmosphere and ocean

- Joint development of DWD and Max-Planck-Institute for Meteorology

$$
\left|\begin{array}{lcl}
\partial_t v_n + (\zeta + f)\, v_t + \partial_n K + w\, \partial_z v_n & = & -c_{pd}\theta_v \partial_n \pi \\
\partial_t w + \nabla\cdot(\mathbf{v}_n w) - w\, \nabla\cdot \mathbf{v}_n + w\, \partial_z w & = & -c_{pd}\theta_v \partial_z \pi - g \\
\partial_t \rho + \nabla\cdot(\mathbf{v}\rho) & = & 0 \\
\partial_t(\rho\theta_v) + \nabla\cdot(\mathbf{v}\rho\theta_v) & = & 0
\end{array}\right.
$$

(v_n, w, ρ, θ_v: prognostic variables)

Code design goals:

- data parallelism and task parallelism
- multi-vendor interoperability
- operational schedule: fixed run time characteristics

In a Nutshell

ICON implements optimization strategies for

- data placement and locality
- reduction of serial bottlenecks
- system-level optimization

ICON parallel program design

	SPMD style programming with master-only message passing
+	OpenMP task parallelism
+	asynchronous I/O
+	advanced data structures for special purposes

ICON's Unstructured Grid

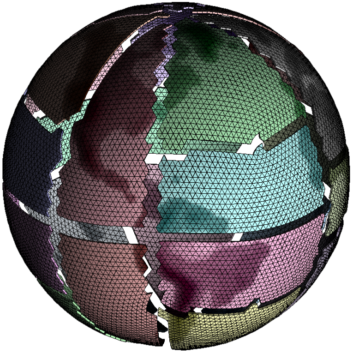
Primal cells: **triangles**

- uses icosahedron for macro-triangulation
- staggered grid, velocity at edge midpoints
- local subdomains ("nests")

Example:
20 km global res.
$\approx 1.3 \cdot 10^6\ \Delta$

$\times$ 90 vertical levels
(up to 75 km)

ICON's Domain Decomposition



Geometric decomposition, 20 partitions

Criteria:

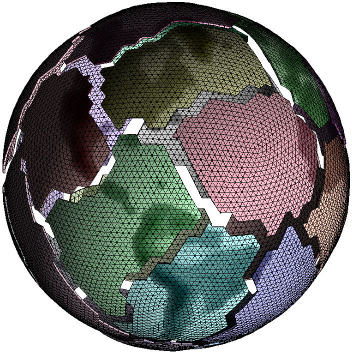
1. **Static load balancing**, e. g. every PE comprises sunlit and shadowed parts of the globe
2. **Communication reduction**

Explicit array partitioning with

- halo regions
- lateral boundary regions
- nest overlap regions
- interior points

. . . avoids conditionals in iterations.

ICON's Domain Decomposition



Min-cut decomposition, 20 partitions

Example, 40 PEs, 40 km global:
Comm. ***ca. -4 %***, *Connections* ***ca. -7.9%***

Criteria:

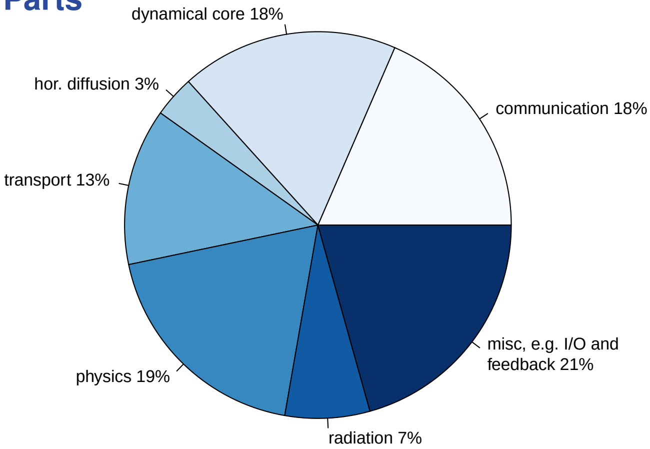
1. **Static load balancing**, e. g. every PE comprises sunlit and shadowed parts of the globe
2. **Communication reduction**

Explicit array partitioning with

- halo regions
- lateral boundary regions
- nest overlap regions
- interior points

. . . avoids conditionals in iterations.

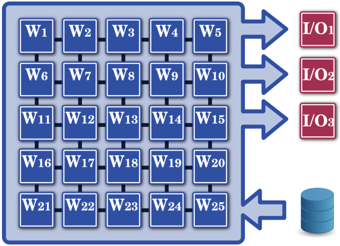
Solver Parts



Run-time portions (example)

I/O: Bottleneck for High-Resolution NWP

Classical root I/O of high-resolution data becomes a critical issue.



The ICON model offloads all computed results to dedicated output nodes.

- computation and I/O overlap
- fast system layer: *Climate Data Interface*
- WMO GRIB2 standard

Advanced Data Structures

Proximity queries in a metric space

"Find the near neighbors in a large data set."

set of cell centers

distance on the sphere

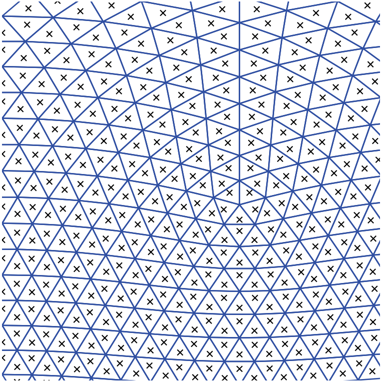
Elementary approach: **Grid method**

Divide search area into small spaces, keep short lists of points.

Alternative approach: **Search trees**

- **Setup phase**: OpenMP pipelining during tree construction
- **Query phase**: Different threads may traverse the tree structure in parallel

Geometric Near-neighbor Access Trees

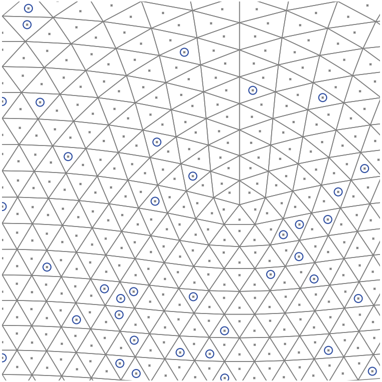


Generalization of k*d-trees*.

1. At the top node of a GNAT choose several distinguished *split points*.
2. The remaining points are classified into groups depending on what Voronoi cell they fall into.
3. Recursive application forms tree levels.
   Additionally store ranges of distance values from split points to the data points associated with other split points.

S. Brin: Near Neighbor Search in Large Metric Spaces. *VLDB '95*

Geometric Near-neighbor Access Trees

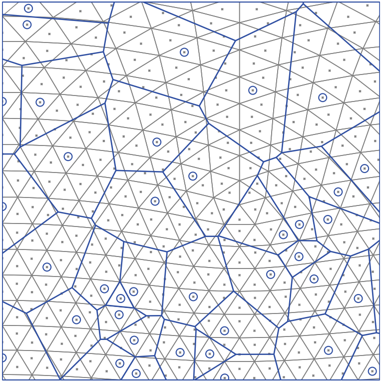


Generalization of *kd-trees*.

1. At the top node of a GNAT choose several distinguished *split points*.
2. The remaining points are classified into groups depending on what Voronoi cell they fall into.
3. Recursive application forms tree levels.
   Additionally store ranges of distance values from split points to the data points associated with other split points.

S. Brin: Near Neighbor Search in Large Metric Spaces. *VLDB '95*

Geometric Near-neighbor Access Trees

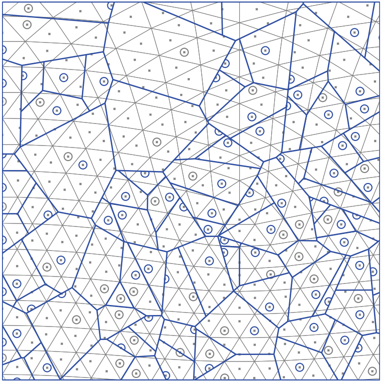


Generalization of *kd-trees*.

1. At the top node of a GNAT choose several distinguished *split points*.
2. The remaining points are classified into groups depending on what Voronoi cell they fall into.
3. Recursive application forms tree levels.
   Additionally store ranges of distance values from split points to the data points associated with other split points.

S. Brin: Near Neighbor Search in Large Metric Spaces. *VLDB '95*

Geometric Near-neighbor Access Trees



Generalization of kd-*trees*.

1. At the top node of a GNAT choose several distinguished *split points*.
2. The remaining points are classified into groups depending on what Voronoi cell they fall into.
3. Recursive application forms tree levels.
   Additionally store ranges of distance values from split points to the data points associated with other split points.

S. Brin: Near Neighbor Search in Large Metric Spaces. *VLDB '95*

Flat-MPI + Hybrid Performance

Objective

75% of Top500 HPC systems have $\geq$ 6 cores per socket.
On-chip clock rates have increased only moderately.

Amdahl '67

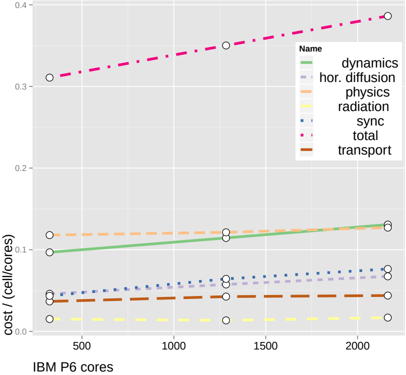
Pessimistic saturation limit
for program scaling on HPC systems:

$$\text{Speedup } S \leq \frac{1}{\alpha} \qquad (\alpha: \text{ non-parallel sections})$$

For NWP: Run time, not problem size is constant.

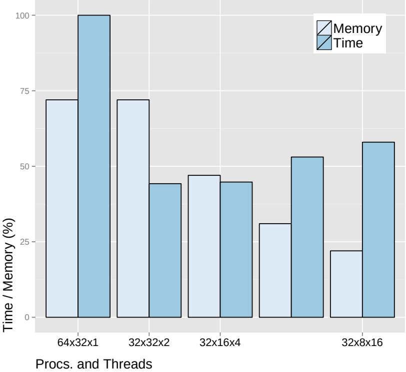
- Parallel amount grows with the number of PEs
- **Weak scaling** matters in practice!

Flat-MPI + Hybrid Performance



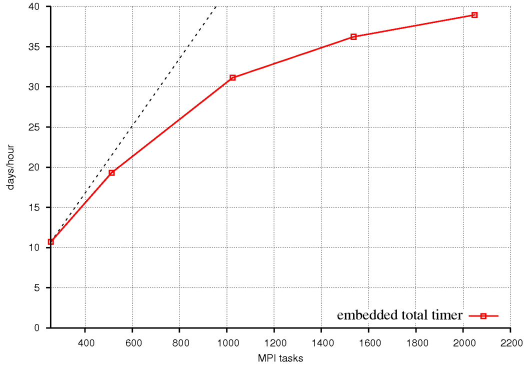
Test setup: **experiment APE_NH**, geom. decomposition, 1 h reduced radiation grid, 48 cells/core

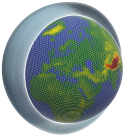
(L. Linardakis, 01/2012)



Test setup: **real data test case**, 2000 steps, geom. decomp., resolution 20 km global, 10 km local

Strong Scaling Test





Test setup:

Non-hydrostatic test, real data,
10 days forecast
Resolution: R2B06 (~ 40km),
90 levels, time step 60s

Parallel setup:

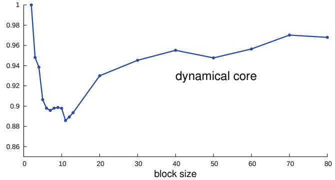
IBM Power6 platform,
6h output,
32 MPI tasks x 2 OpenMP threads/node (smt)

Portable Efficiency?

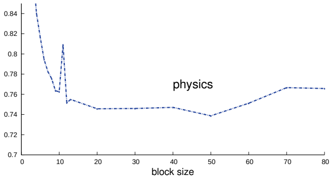
Cache Optimization

Unstructured grids make extensive use of indirectly accessed arrays.
Manual loop transformations:

- **Loop interchange:**
  Array assignments are transformed to allow vectorization <u>and</u> improved cache performance on scalar platforms.
- **Loop tiling:**
  Block-partitioned loop iteration space, enhancing cache reuse.
  Block size = “automatic” optimization for a wide range of platforms.



IBM Power 6, 40 global



Task Assignment

Prevent tasks from going off-node, reduce switch communication.

1	2	3	4	5	6	7	8	9	10	11	12	MPI tasks
9	8	3	4	2	6	7	5	1	12	11	10	CPUs

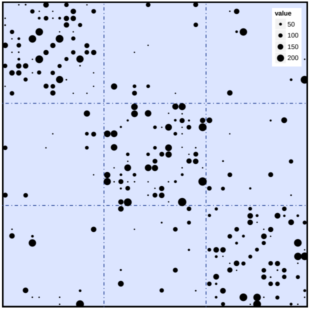
Task placement is a **Quadratic Assignment Problem**:

$$\min_{\pi \in \Pi(n)} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} \cdot b_{\pi_i \pi_j} \,, \qquad \begin{array}{lcl} a_{ij} & : & \text{flow matrix,} \\ b_{ij} & : & \text{distance matrix.} \end{array}$$

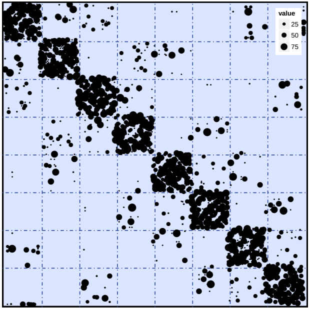
NP-hard problem!

B. Brandfass 2010 ; E. Taillard 1991 (robust taboo search)

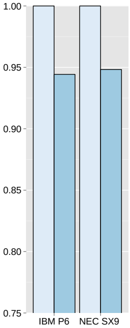
Task Assignment



Reordered communication matrix, 3x15 tasks



Reordered communication matrix, 8x32

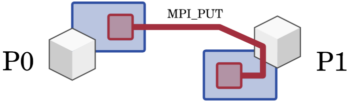


Task assignment benefit

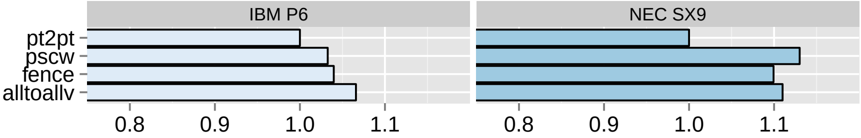
Remote Memory Access

MPI-2 provides a model for remote memory access.

- **potentially more lightweight –** benefit from special hardware?



Local ghost-cell exchange in the ICON model:



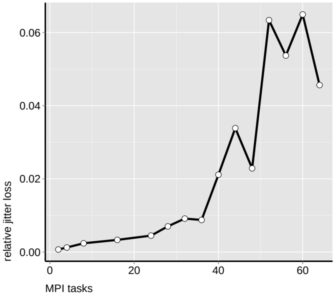
System Noise

- OS jitter: daemons, IRQs
- MPI jitter
- high-frequency perturbations (e. g. cache-related)

Analysis with
n-processor micro-benchmark.

Relative OS jitter loss
(fix work quantum,
m measurements):

$$lost_{rel} = \frac{\sum_{i=1}^{m} \max(\bar{t}_{compute,i,\cdot}) - m\,\bar{t}_{compute}}{m\,\bar{t}_{compute}}$$



In a Nutshell

ICON implements optimization strategies for

- data placement and locality
- reduction of serial bottlenecks
- system-level optimization

ICON parallel program design

	SPMD style programming with master-only message passing
+	OpenMP task parallelism
+	asynchronous I/O
+	advanced data structures for special purposes

In a Nutshell

ICON implements optimization strategies for

- data placement and localit
- redu
- syste

ICON
parallel pr
design

Unsolicited Advice for HPC Vendors:

- OS jitter: System kernel must be minimized to increase performance
- Operational use: Compiler support for reproducible code
- Pragma-based, incremental use of stream architectures

Florian Prill

Met. Analyse und Modellierung
Deutscher Wetterdienst

e-mail: Florian.Prill@dwd.de