Optimizing a vector code for cache-based multi-core systems: Porting ECHAM6

Luis Kornblueh

Max-Planck-Institut für Meteorologie, Hamburg

MAX-PLANCK-GESELLSCHAFT

Max-Planck-Institut für Meteorologie

The climate model ECHAM6

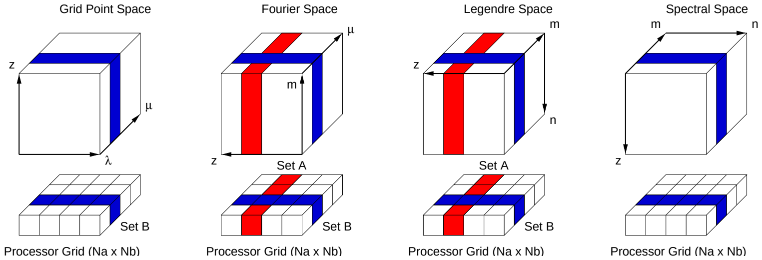
- Spectral dynamical core
- Semi-implicit leapfrog time differencing
- Flux-form semi-Lagrangian transport of passive tracers
- Shortwave and longwave radiation schemes (latest SRTM, LRTM)
- Stratiform clouds based on micro-physical prognostic equations
- Convection solved by a mass-flux scheme (Tiedtke/Nordeng)
- Subgrid-scale induced gravity wave drag
- Vertical diffusion (subgrid-scale turbulence closure by TKE)
- Extensive multi-tile land-surface model including dynamic vegetation (JSBACH)
- parameterizations for the statosphere - gravity waves and water vapour chemistry
- real orbit and variable solar activity per shortwave band

Model states

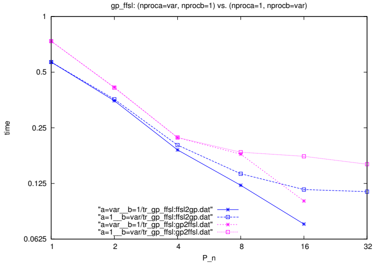
Decomposition challenge

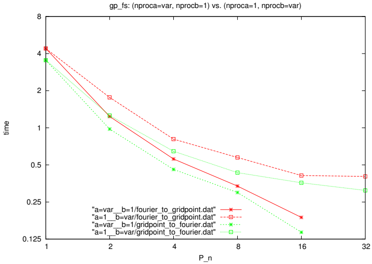
- spectral space
  - Legendre space
  - Fourier space
- grid point space
- transport flux-form space
- grid point space
  - Fourier space
  - Legendre space
- spectral space

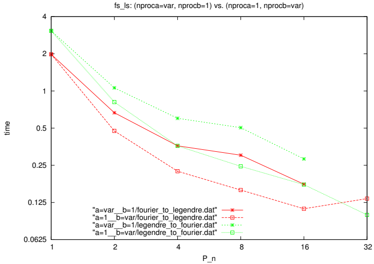
Double transposition strategy

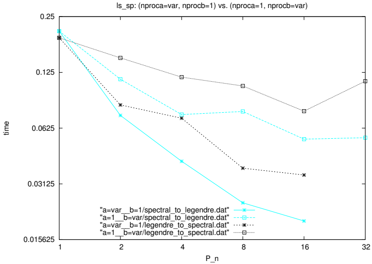


Extensive optimization work: transpose by unitrans library (ScaLES, based on MPI collectives, license not clear yet), or fully reworke qd and OpenMP optimized hand coded transposes.


gp_ffsl: (nproca=var, nprocb=1) vs. (nproca=1, nprocb=var)
time
P_n
1
0.5
0.25
0.125
0.0625
1 2 4 8 16 32
"a=var__b=1/tr_gp_ffsl:ffsl2gp.dat"
"a=1__b=var/tr_gp_ffsl:ffsl2gp.dat"
"a=var__b=1/tr_gp_ffsl:gp2ffsl.dat"
"a=1__b=var/tr_gp_ffsl:gp2ffsl.dat"


gp_fs: (nproca=var, nprocb=1) vs. (nproca=1, nprocb=var)
time
P_n
8
4
2
1
0.5
0.25
0.125
1 2 4 8 16 32
"a=var__b=1/fourier_to_gridpoint.dat"
"a=1__b=var/fourier_to_gridpoint.dat"
"a=var__b=1/gridpoint_to_fourier.dat"
"a=1__b=var/gridpoint_to_fourier.dat"


fs_ls: (nproca=var, nprocb=1) vs. (nproca=1, nprocb=var)
time
P_n
4
2
1
0.5
0.25
0.125
0.0625
1 2 4 8 16 32
"a=var__b=1/fourier_to_legendre.dat"
"a=1__b=var/fourier_to_legendre.dat"
"a=var__b=1/legendre_to_fourier.dat"
"a=1__b=var/legendre_to_fourier.dat"


ls_sp: (nproca=var, nprocb=1) vs. (nproca=1, nprocb=var)
time
P_n
0.25
0.125
0.0625
0.03125
0.015625
1 2 4 8 16 32
"a=var__b=1/spectral_to_legendre.dat"
"a=1__b=var/spectral_to_legendre.dat"
"a=var__b=1/legendre_to_spectral.dat"
"a=1__b=var/legendre_to_spectral.dat"

Redistribution for tracer transport

Scaling improvements

- resort in latitudinal direction (Na)
  - higher wind speed (transporting) in East-West direction, hardly predictable Courant number
  - low wind speed in north-south direction, Courant number < 1
- and vertical levels and tracers (Nb)

Transposition back into grid point space before calculation of vertical transport.

Loop-level parallelization

OpenMP usage

- whole physics block - OpenMP orphaning
- remaining model - OpenMP on loop level
- communication not in OpenMP regions yet
- Performance problem: OS jitter - no detailed analysis on AIX yet, use of coscheduling under evaluation

Vectorisation/Cache blocking

Flexibility

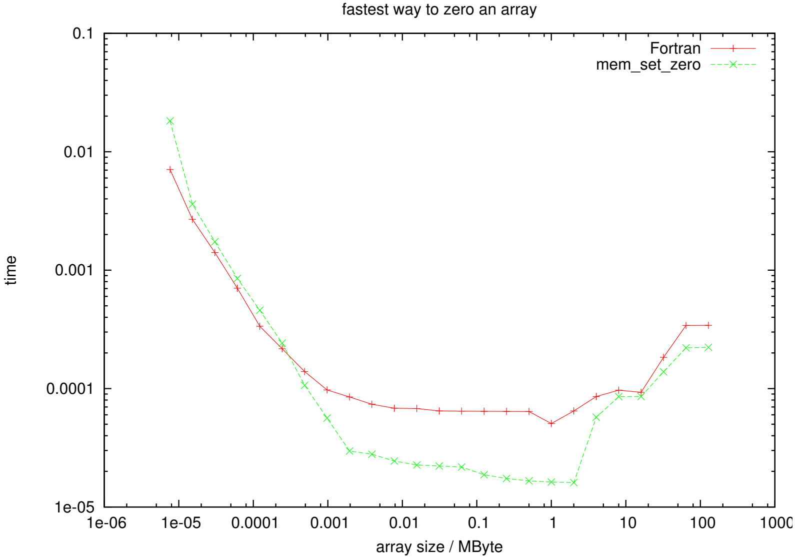
- Physics block - high level strip mining (VL/average optimal cache blocking), *optimal nproma PWR6: 72 could be reduced down to 48 without big performance degradation (5%) - fully L2 resident*
- replaced calculation of water vapour saturation from lookup tables of size 400000 (flushing L3 cache) to spline interpolation with base table L2 cache resident
- remaining optimizations are basically improving the vectorization by hand gather/scatter for several loops to reduce the compiler generated once.
- replace IFs by FSEL/MERGE or precomputed list vectors
- select the best nondestructive compiler options on a per file base and adding directives

xlf specifics

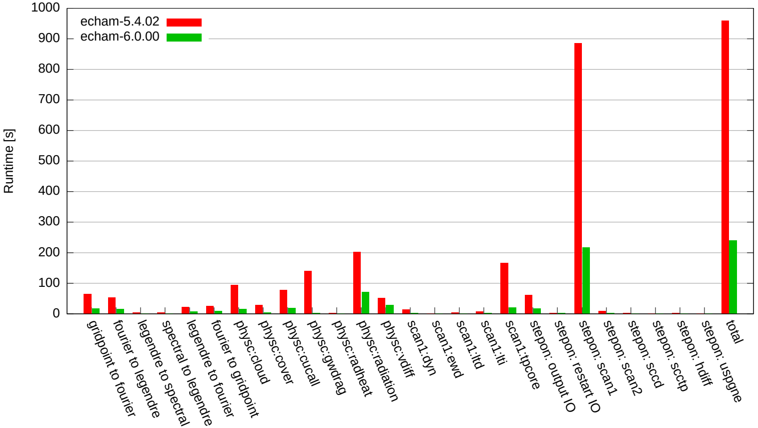
Supporting the xlf compiler

- 114 compiler directives
- 166 FSEL
- 44 SWDIV_NOCHECK
- 267 compiler option directives in the files to allow a single common set of options for the Makefile
- set an array to 0 via __dcbz

Set an array to zero

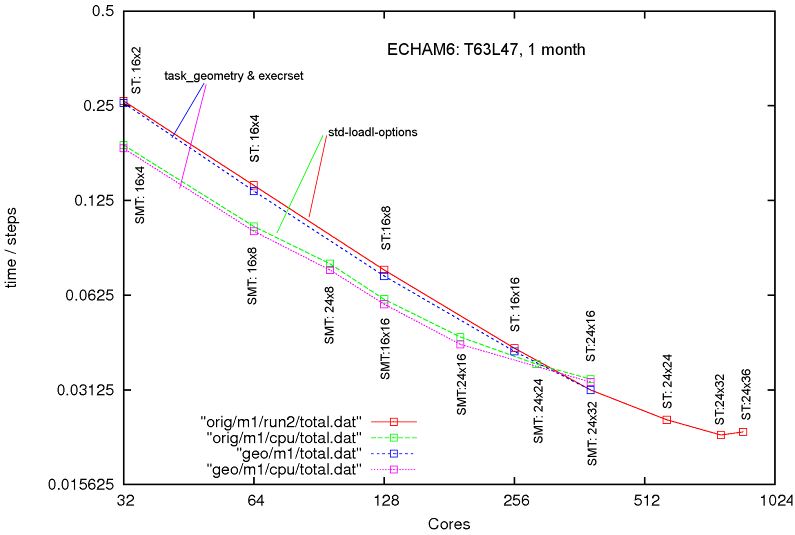


Optimization improvements

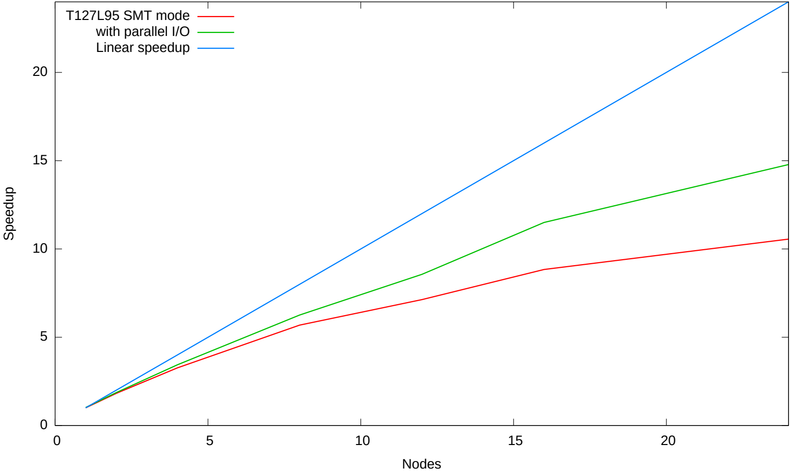


echam6 has roughly 35 % more floating point instructions than echam5!

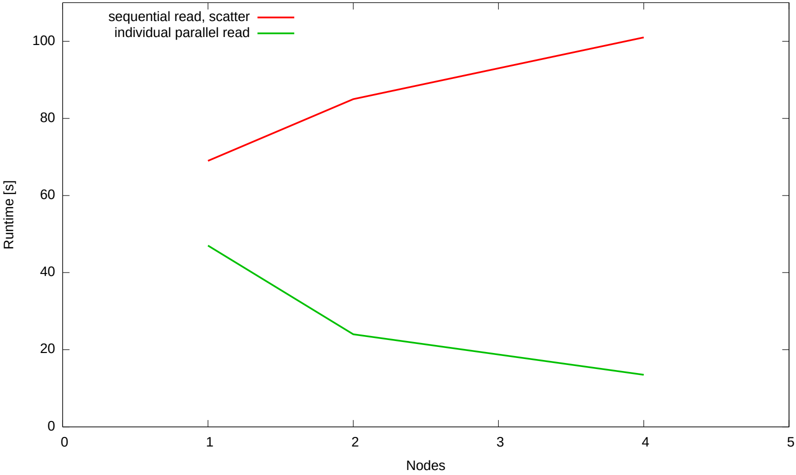
Scaling and ST/SMT mode



Scaling of a larger resolution



Restart I/O - netcdf



Parallel I/O - GRIB

see the other talk from Wednesday.

Conclusion

Work done

- Scaling is at least possible with an efficiency of 50% when talking care on sufficient computational load per core
  Most model resolutions we use can use bacause of that only a couple of hundreds to a couple of thousand cores.
- Per core efficiency is between 15 and 25% in SMT mode

Work to do

- Finalize/polish implementation of parallel I/O
- Improve coding style of physics section, develop coding guide
- Add directives and compiler options for other platforms and compiler.

Thanks to

- Jörg Behrens, DKRZ
- Deike Kleberg, MPIM
- Mathias Pütz, IBM
- ScaLES, BMBF
- Uwe Schulzweida, MPIM