MAX-PLANCK-GESELLSCHAFT

ASYNCHRONICITY

THE CHALLENGE OF FINE-GRAINED PARALLELISM

Luis Kornblueh

October 26, 2016

Max-Planck-Institut
für Meteorologie

INVITATION NOT TO BE IN TIME

NOT FOR THE FAINT HEARTED

- Climate modeling requires a lot of computing time for tuning models without scientific output.
- Very difficult to run large ensembles.
- New machines get build up and tested for some time before getting into production.
- Let's join this (needs adventures scientist, courageous computing center director, and non-dogmatic vendor) ...

NOT FOR THE FAINT HEARTED

- Climate modeling requires a lot of computing time for tuning models without scientific output.
- Very difficult to run large ensembles.
- New machines get build up and tested for some time before getting into production.
- Let's join this (needs adventures scientist, courageous computing center director, and non-dogmatic vendor) ...

NOT FOR THE FAINT HEARTED

- Climate modeling requires a lot of computing time for tuning models without scientific output.
- Very difficult to run large ensembles.
- New machines get build up and tested for some time before getting into production.
- Let’s join this (needs adventures scientist, courageous computing center director, and non-dogmatic vendor) ...

NOT FOR THE FAINT HEARTED

- Climate modeling requires a lot of computing time for tuning models without scientific output.
- Very difficult to run large ensembles.
- New machines get build up and tested for some time before getting into production.
- Let’s join this (needs adventures scientist, courageous computing center director, and non-dogmatic vendor) ...

NOT FOR THE FAINT HEARTED

- Climate modeling requires a lot of computing time for tuning models without scientific output.
- Very difficult to run large ensembles.
- New machines get build up and tested for some time before getting into production.
- Let's join this (needs adventures scientist, courageous computing center director, and non-dogmatic vendor) ...

Thanks to Thomas Schulthess,CSCS, and Cray: 100 member historical ensemble, 67 member 1%CO2, 5000 years pi-control, and 3000 years 4xCO2 and a new tuned (Mauritsen, Roeckner, Haak, . . .) HighRes model. A large number of PhD students working on the results.

SETTING THE STAGE

WHAT IS DRIVING NEW DEVELOPMENTS?

Redefinition: the models we talk about consist of all components which are used in the workflow!

WHAT IS DRIVING NEW DEVELOPMENTS?

Redefinition: the models we talk about consist of all components which are used in the workflow!

The development of global circulation models in its current form has to change and respond to major challenges in hardware development.

WHAT IS DRIVING NEW DEVELOPMENTS?

Redefinition: the models we talk about consist of all components which are used in the workflow!

The development of global circulation models in its current form has to change and respond to major challenges in hardware development.

Example:
old node — 12 cores 2.5 GHz
new node 18 cores 2.1 GHz

WHAT IS DRIVING NEW DEVELOPMENTS?

Redefinition: the models we talk about consist of all components which are used in the workflow!

The development of global circulation models in its current form has to change and respond to major challenges in hardware development.

Example:
old node — 12 cores 2.5 GHz
new node 18 cores 2.1 GHz

Consequence: more and more, fine grained parallelism is required to achieve the necessary performance to answer scientific questions posed.

OBJECTIVES

Key points to consider are

- to keep all critical hardware resources concurrently in use,
- to minimize or hide the response time for remote access and service requests,
- to reduce contributions of parallel resources and task scheduling not used for computational work itself, and
- to minimize resource access conflicts.

OBJECTIVES

Key points to consider are

- to keep all critical hardware resources concurrently in use,
- to minimize or hide the response time for remote access and service requests,
- to reduce contributions of parallel resources and task scheduling not used for computational work itself, and
- to minimize resource access conflicts.

OBJECTIVES

Key points to consider are

- to keep all critical hardware resources concurrently in use,
- to minimize or hide the response time for remote access and service requests,
- to reduce contributions of parallel resources and task scheduling not used for computational work itself, and
- to minimize resource access conflicts.

OBJECTIVES

Key points to consider are

- to keep all critical hardware resources concurrently in use,
- to minimize or hide the response time for remote access and service requests,
- to reduce contributions of parallel resources and task scheduling not used for computational work itself, and
- to minimize resource access conflicts.

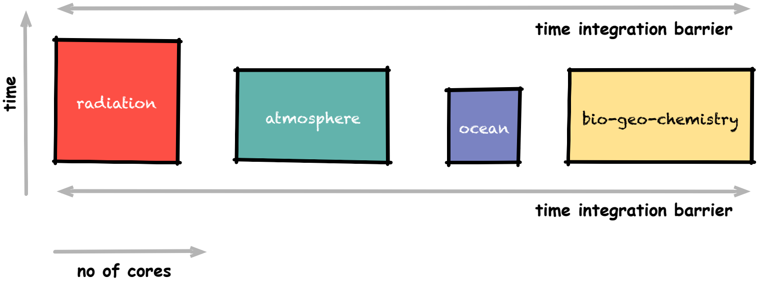
ALGORITHMS

The solution framework proposed consists of the

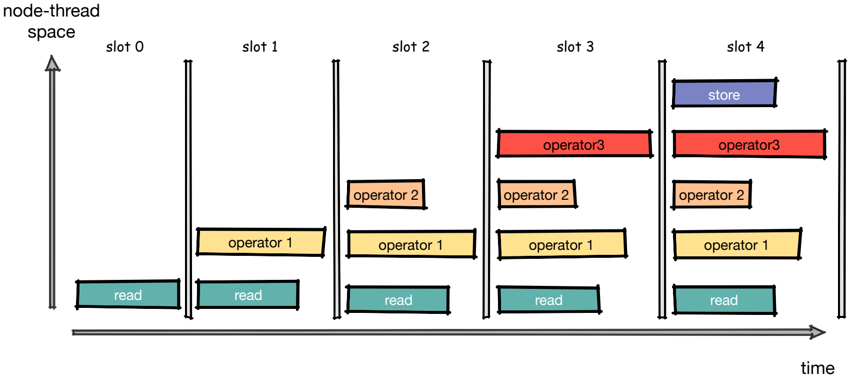
- functional description of processing algorithms, and
- a direct acyclic graph representation (DAG) of processing (to be used for optimization and parallelization).

PROCESSES COMPACTION

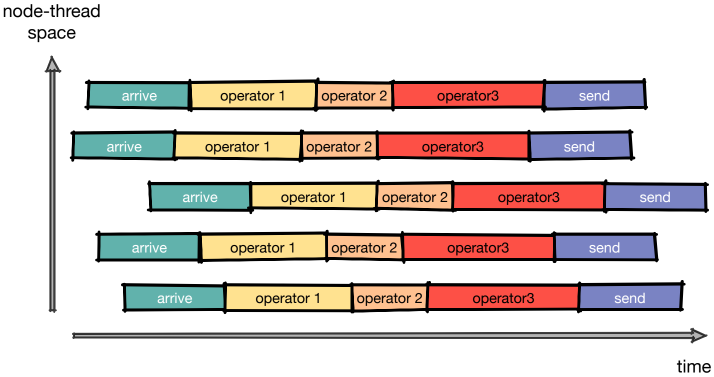
COARSE-GRAINED ASYNCHRONOUS PROCESSES



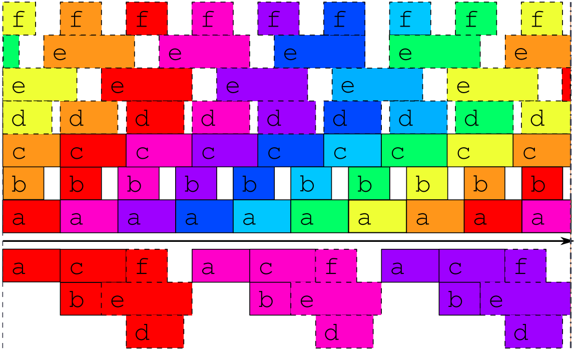
HOW A VECTOR PIPELINING PROCESSING MODEL WORKS



MOVING TO A DAG BASED PROCESSING MODEL



DAG BASED META-SCHEDULING



cylc, Hilary Oliver, NIWA

FUTURE

DEVELOPMENT ACTIVITIES

- Development of a DAG based worker/broker toolkit with arithmetic operators as first test and later add cdo
  Hermes, Florian Rathgeber and Tiago Quintino (ECMWF)
- Refactoring of cdo by moving to C++ and disentangling command line and operator handling
- Develop an evaluation hierarchy for cdo operators

DEVELOPMENT ACTIVITIES

- Development of a DAG based worker/broker toolkit with arithmetic operators as first test and later add cdo
  Hermes, Florian Rathgeber and Tiago Quintino (ECMWF)
- Refactoring of cdo by moving to C++ and disentangling command line and operator handling
- Develop an evaluation hierarchy for cdo operators

DEVELOPMENT ACTIVITIES

- Development of a DAG based worker/broker toolkit with arithmetic operators as first test and later add cdo
  Hermes, Florian Rathgeber and Tiago Quintino (ECMWF)
- Refactoring of cdo by moving to C++ and disentangling command line and operator handling
- Develop an evaluation hierarchy for cdo operators

WHAT NEXT?

- Get a working prototype of post-processing tools and scheduling
- Using meta-scheduling for applicable problems
- Rethink the time operator splitting of the model physics to allow for a more functional, concurrent usable representation of processes — or resolve those explictly …
- Development and application of model developer friendly Domain Specific Languages (DSL)

WHAT NEXT?

- Get a working prototype of post-processing tools and scheduling
- Using meta-scheduling for applicable problems
- Rethink the time operator splitting of the model physics to allow for a more functional, concurrent usable representation of processes — or resolve those explictly . . .
- Development and application of model developer friendly Domain Specific Languages (DSL)

WHAT NEXT?

- Get a working prototype of post-processing tools and scheduling
- Using meta-scheduling for applicable problems
- Rethink the time operator splitting of the model physics to allow for a more functional, concurrent usable representation of processes — or resolve those explictly . . .
- Development and application of model developer friendly Domain Specific Languages (DSL)

WHAT NEXT?

- Get a working prototype of post-processing tools and scheduling
- Using meta-scheduling for applicable problems
- Rethink the time operator splitting of the model physics to allow for a more functional, concurrent usable representation of processes — or resolve those explictly . . .
- Development and application of model developer friendly Domain Specific Languages (DSL)

ADDITIONAL CONSTRAINTS

UNKNOWNS

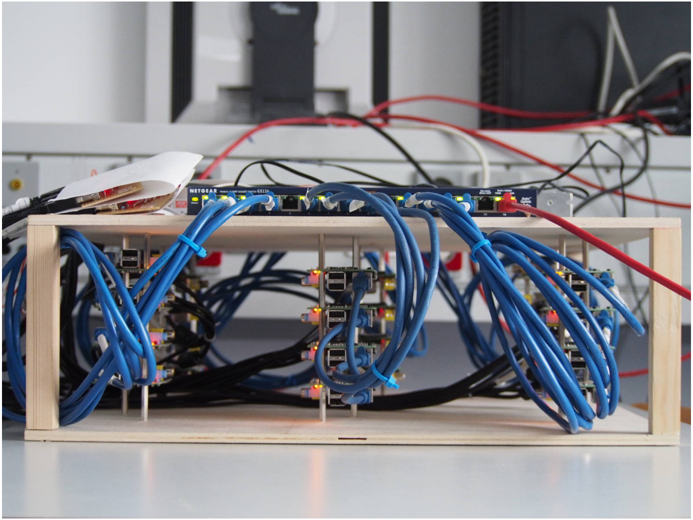
There are two more aspects contributing to effective system usage. Power consumption and the system's reliability.

Who does not have application specific checkpoint/restart?

The influence of this parameters on future development are not in the primary scope of this considerations, but are supposed to have a strong impact on solutions.

PERHAPS . . .

LATEST HARDWARE DEPLOYMENT



Courtesy by Miriam, 7a

SYSTEM CHARACTERISTICS

- 24 nodes with Broadcom BCM2835 SoC (700 MHz ARM 1176JZF-S, VideoCore IV GPU)
- Non-blocking fat tree high speed network IEEE 802.3u (100BASE-TX) via USB-2 Bus (aggregated 273.6 MB/s)
- NFSv4 network filesystem, SLURM, GCC, mpich
- Linux Debian jessie (Kernel 4.4)

SYSTEM CHARACTERISTICS

- 24 nodes with Broadcom BCM2835 SoC (700 MHz ARM 1176JZF-S, VideoCore IV GPU)
- Non-blocking fat tree high speed network IEEE 802.3u (100BASE-TX) via USB-2 Bus (aggregated 273.6 MB/s)
- NFSv4 network filesystem, SLURM, GCC, mpich
- Linux Debian jessie (Kernel 4.4)

SYSTEM CHARACTERISTICS

- 24 nodes with Broadcom BCM2835 SoC (700 MHz ARM 1176JZF-S, VideoCore IV GPU)
- Non-blocking fat tree high speed network IEEE 802.3u (100BASE-TX) via USB-2 Bus (aggregated 273.6 MB/s)
- NFSv4 network filesystem, SLURM, GCC, mpich
- Linux Debian jessie (Kernel 4.4)

SYSTEM CHARACTERISTICS

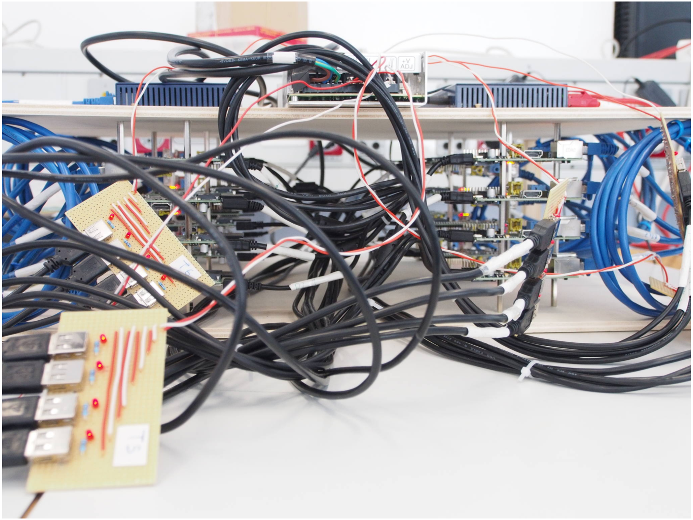
- 24 nodes with Broadcom BCM2835 SoC (700 MHz ARM 1176JZF-S, VideoCore IV GPU)
- Non-blocking fat tree high speed network IEEE 802.3u (100BASE-TX) via USB-2 Bus (aggregated 273.6 MB/s)
- NFSv4 network filesystem, SLURM, GCC, mpich
- Linux Debian jessie (Kernel 4.4)

SYSTEM CHARACTERISTICS

- 24 nodes with Broadcom BCM2835 SoC (700 MHz ARM 1176JZF-S, VideoCore IV GPU)
- Non-blocking fat tree high speed network IEEE 802.3u (100BASE-TX) via USB-2 Bus (aggregated 273.6 MB/s)
- NFSv4 network filesystem, SLURM, GCC, mpich
- Linux Debian jessie (Kernel 4.4)

Successfully run echam 4.6 T31L19 (CVS version 6.00, 2000-09-19 08:26:58 (Git: da9d477) , no code changes) using the full system.

ENERGY CONSUMPTION 100 W



Courtesy by Miriam, 7a