### How to overcome common performance problems in legacy climate models

Jörg Behrens<sup>1</sup>

Contributing: Moritz Handke<sup>1</sup>, Ha Ho<sup>2</sup>, Thomas Jahns<sup>1</sup>, Mathias Pütz<sup>3</sup>

Deutsches Klimarechenzentrum (DKRZ)
 Helmholtz-Zentrum Geesthacht (HZG)
 CRAY, formerly IBM



ECMWF workshop 2012

1

### How to overcome common performance problems in legacy climate models

# Problems:

- Legacy models: adapted to past hardware generations
- Today: much more cores, models do not scale
- #{models} x #{performance-problems} > manpower
   > Solution path: improve software infrastructure (libraries)
- Order and number of bottlenecks change with #{cores}
  - Performance analysis comes first



### Outline

• Quick performance analysis:

- Example (TRIM: "Tidal, Residual, and Intertidal Mudflat Model")

• Problems & solutions

More analysis (dynamic load imbalance)

- Performance software infrastructure

   Motivating example (MPIOM)
   ➤YAXT (Yet Another Exchange Tool)
- Quick view on indirect data access



### **Performance analysis example**

#### From: Program Analysis and Tuning Workshop (DKRZ, June 2012)

User were invited to analyze their applications using Vampir / Scalasca. Example: coupled model (TRIM & CICE & CCLM & OASIS3)





### **Performance problem pattern (1)**



### **Performance problem pattern (2)**





### Solutions to problems 1 & 2

- 1. p2p-gather & serial IO:
  - Quick: use ScalES\* 2-phase gather (much faster)
  - Better: parallel IO (not discussed here)
- 2. Load imbalance:
  - Analyze further: Decomposition? Cost-function?
  - Inspect source code
  - Detailed measurement of work load

\*Scalable Earth System Models



### **TRIM: average work load**



- nlev(t)<sub>i,j</sub> ~ cost(t)<sub>i,j</sub>
- Model: (0/1 land-sea mask) cost function for partitioner
- Partitioner gives suboptimal decompositions
- ➢ No Problem for low nprocs
- But steals efficiency



# TRIM: σ[work load]



- $\sigma_{i,j}$  over one month
- Low nlev-variation
- Error of a static
   decomposition should be
  - small enough to justify work
  - on improvement.
- O (Dynamic load-balancing would require major rewrite)



### **TRIM: iteration count** $\rightarrow$ **efficiency**



Can be improved to 91% by using the next magic partition numbers (16x8)



### MPIOM [TP04L40: 8x4] Load Balance





- 1. Wet-point-only optimized
- workload changed
- Unfit legacy decomposition
- Nothing gained

- 2. Adapted decomposition
- Old boundary exchange fails
- Reprogramming exchange?
- YAXT-formulation:
  - works for both cases
  - > faster



ECMWF workshop 2012

# YAXT (Yet Another Exchange Tool)

- Successor of UniTrans (ScalES prototype)
- General communication library on top of MPI
  - C with Fortran interfaces
- Simple definition of communication
  - Change: programming communication  $\rightarrow$  parameterization of comm.
- Fast & flexible communication made easy
  - Automatically generates derived MPI datatypes
  - Easy aggregation of communication steps



### **UniTrans/YAXT: key concepts**

- Global domain: indexed elements {e<sub>i</sub>}
- Source and target indices: {s<sub>i</sub>}, {t<sub>i</sub>} per process
- Automatic construction of index map ~ Extended communication matrix
- Automatic or user-supplied (static) data offsets:
   ➢ local index position → local data position
   ➢ index map → data redistributor
- Automatic generation of optimized MPI-datatypes
- Aggregation of several data redistributors
- Focus on static communication pattern
  - Initialization is considered a one time cost





Source decomposition:

Core points

Target decomposition:

• Halo points



#### Hallo update



### **Complexity of data exchange**

### Some aspects:

- Topology & physical quantities

   not always simple (e.g.: tripolar grid)
   possible sign change at boundary
- Stencil shape
  - depends on physical operator
  - halo update requirements
- Subdomain boundaries
  - given by partitioner
  - Communication matrix

Independent of parallelism (serial concept)

depends on parallelism (parallel concept)



# Simplified generation of communication objects (prototype)



#### Input:

- Global topology (GTOs)
- Stencil shape
- Local subdomain boundaries



#### **Output:**

Data redistributor for halo

update

#### possible data representations:

- Simple arrays, or
- derived element offsets



### **Example: MPIOM boundsexchnage**

# Handwritten – requires symmetric regular decompositions

#### UniTrans/YAXT general solution

<u>F</u> ile	<u>W</u> indow	<u>H</u> elp	

Ances1       Antesis	12.7433 s	12.7434 s 12.7435 s	12.7436 s 12.74	37 s 12.7438 s 12.7439 s	12.7440 s 12.7441	1 s 12.7442 s 12.7	443 s 12.7444 s 12	/445 s 12./446 s 12./447 s
Access 1       Access 2       Access 2 <td< td=""><td>Process 0</td><td>MPLIsend</td><td></td><td>MPI_Isend</td><td>MPI_Isend</td><td>MPI_Isend</td><td>A Contraction of the second seco</td><td>lisend MPI_Barrier</td></td<>	Process 0	MPLIsend		MPI_Isend	MPI_Isend	MPI_Isend	A Contraction of the second seco	lisend MPI_Barrier
Ancess 2         Her press         Her press <th< td=""><td>Process 1</td><td>MPLISend</td><td>MPI_Wait</td><td>MPI_Wait</td><td>MR Isend</td><td>MPI_Isend</td><td>MPI_Isend</td><td>MP Waltall</td></th<>	Process 1	MPLISend	MPI_Wait	MPI_Wait	MR Isend	MPI_Isend	MPI_Isend	MP Waltall
Aucus 1       Influence	Process 2	MPI_Isend		MPI_lsand	MPL send	MPI_Isend	MB Isend	MPI Dead
Maccas /       Marganet       Marganet <th< td=""><td>Process 3</td><td>MPI_Isend</td><td>MPI Wait</td><td></td><td>MPI_Wait</td><td>MPI_Isend</td><td></td><td>Harry Wandah</td></th<>	Process 3	MPI_Isend	MPI Wait		MPI_Wait	MPI_Isend		Harry Wandah
Maccess 0       Marcanowy Maccess 0       Marcanowy Maccess 0       Marcanowy	Process 4	MP_Isend		MBK Ise X	MPLISEND	MPI_Isend	MPI Isend	
Maccess         Maccess <t< td=""><td>Process 5</td><td>MPLisend</td><td>MPI_Wait</td><td>Mer Wait</td><td>MPI Isend</td><td>MPI_Isend MPI_Isend</td><td></td><td>PAPE Without Matri Bairier</td></t<>	Process 5	MPLisend	MPI_Wait	Mer Wait	MPI Isend	MPI_Isend MPI_Isend		PAPE Without Matri Bairier
Process         Mill sond	Process 6	MPI_Isend		MPILISON	MPI_Isend	MPI_Isend MPI_Isend		
Process 0       May Lead       May Lead <t< td=""><td>Process 7</td><td>MPI_Isend</td><td>MPI_Wait</td><td></td><td>MPI_Wait</td><td>MPI_Isend</td><td>MPLISend MPL</td><td>MPI Waital A MIP Garrier</td></t<>	Process 7	MPI_Isend	MPI_Wait		MPI_Wait	MPI_Isend	MPLISend MPL	MPI Waital A MIP Garrier
mocess 0	Process 8	MPI_Isend MPI_Isend	MPL_Wait	MP1_wait A Pape	Barrier	MPI_Isend MPI Iseng		
Marcado	Process 9	MPLisend		MPLISENC	MPI_Barrier	MPI Isend MPI Isend		HAPLISend MPR Barner
Process12       MPT1SFRId       MPLASED       MPLASED <td>Process 10</td> <td>HIPLISEND</td> <td>MPI_Wait</td> <td>MHILYS and</td> <td>MPI_Barrier</td> <td>MPt_Isend</td> <td>MPI_Isend</td> <td>Mit watan</td>	Process 10	HIPLISEND	MPI_Wait	MHILYS and	MPI_Barrier	MPt_Isend	MPI_Isend	Mit watan
Process 1       Mm rend Mm Land       Mm V       Mm Rame       Mm rend	Process 11	MPI_Isend		MR Isend	Pl_Barrier 🔸	MPISend	MPP Isend	THE PARTY MANAGER
Process13       MMT tend       MMT tend <t< td=""><td>Process 12</td><td>MPI_Isend MPI_Isend</td><td>MP/_Wait</td><td>MIPI_Wate</td><td>Barrier</td><td>MPI_tsend</td><td>MPI Isend</td><td></td></t<>	Process 12	MPI_Isend MPI_Isend	MP/_Wait	MIPI_Wate	Barrier	MPI_tsend	MPI Isend	
Process 14       MM Lend       MM VAL       MM Lend       MM VAL       MM Lend	Process 13	MPI_Isend		MIPI_Isena	MPI_Barrier	MPI_tsend MPI_tsend		MPL GENERAL TO A PT
Process15       MMT Ison2	Process 14	MPLIsend	MPI Wait	MPLSend	TMPI_Barrier	MPI Isend MPI Isend	MBLISEDIC MPLISED	Miller wantal
Process10       MMR_Lsand       MMR Vait       MMR Vait       MMR Barrier       MMR Sand	Process 15	MPI_Isend		MPL Isend	4P) Barrier	19PI_Isend	MPL Isend	
Process17       MMT Desci	Process 16	MPLIsend	MPI_Wait	MPI_Isend	MPI_Barrier	MEL Isend		MPL Wantan
Process 18 MB_ Wat MB_ Wat MB_ Wat MB_ Wat MB_ Barrier <td>Process 17</td> <td>MPI_ISEND</td> <td></td> <td>MPL Wate MpL Wait</td> <td>MPI_Barrier +</td> <td>MPI Isend MPI Isend</td> <td>- Alle</td> <td>MAN Gener Anna Dealtan</td>	Process 17	MPI_ISEND		MPL Wate MpL Wait	MPI_Barrier +	MPI Isend MPI Isend	- Alle	MAN Gener Anna Dealtan
Process19       MMT Isend       MMP Barrier	Process 18	MPL Wait	MPI_Wait	MPI_Wait MPI_Wait	• MPLBarrier	MPI_Isend MPI_Isend		MPI Warrier
Process 20       IMPL Send	Process 19	MPI_Isend		MPI_Isend	MPI_Barrier	MPI_Isend MPI_Isend		Barrier
Process 21       MMr tsend       MMr usend	Process 20	MPLIsend	MRL Wait	MPLIsend MPLWa	ait MPI_Barrier	MPI_Isend		
Process 22       MPL Wait       MPL Wait       MPL Wait       MPL Barrier       MPL Send	Process 21	MPI_Isend		MPI_Wait MPI_Wait	MPI_Barrier ++	MPI_Isend MPI_Isend		Mini Jacod
Process 23     MMP Isend     MMP	Process 22	MPI_Wait	MPI_Wait	MPI_Wait MIPI_Wait	• API_Barrier	MPLIsend MPLIsend	METISENd	Mits Manager All
Process 24     PMP Lsend     MMP	Process 23	MPI_Isend		MR Isend	MPI_Barrier	MPI Isend MPI Isend	MIN Isend	
Process 25     MHr Isend     MHr	Process 24	MPI_Isend MPI_Isend		MPI dualit	MiPl_Barrier	MPL Isong		Tele Woldan Henrier
Process 26     MHr tsend     MH Lsend     MH	Process 25	MP_Isend		MRI ISEDA	(er \	MPILSEND	MPTIseed	MPI_Barrier
Process 27     MH- Isend     MH- My Wit     MH/ MH Barrier     MH Isend     MH Isend     MH Isend       Process 28     MH- Isend     MH- Isend     MH/ Wait     MH/ Barrier     MH Isend	Process 26	MP_Isend MPLIsend	MPI_Wait	MPL Send MPL Barrier		MELISEND		MPI_Wattant MPI_Barrier
Process 28     MHF_Isend     IMP_Vail     MPI_Vail     MPI_Vail     MPI_Parrier       Process 29     MHF_Isend     MPI_Vail     MPI_Send     MPI_Send     MPI_Send       Process 30     MHF_Isend     MPI_Vail     MPI_Send     MPI_Send     MPI_Send       Process 31     MHF_Isend     MPI_Vail     MPI_Matrier     MPI_Send     MPI_Send	Process 27	MPI_Isend		MPI_Wait MPI_Wait	MPI_Barrier	MPI_Isend	MPI Isend	HILL MPL_Barrier
Process 29     MHr_isend     MP_isend     MP	Process 28	MPI_Isend MPI_Isend		MPI_Walt MPI_Walt	MPI_Barrier	MPL Isend		MPL Waitain MPL Barrier
Process 30 MHP_Isend MPL_Isend MPL_I	Process 29	MPI_Isend		MPL/send MPL_Barri	ier	MPLISERIO	4	Pl_Isend MPl_Waitate WMPl_Barrier
MPI_Barrier 🗰 MPI_Bend MPI_BE	Process 30	MPI_Isend MPI_Isend	MPL Wait	MPL_Isend 44PL_Barrie	er 🔪	MPL Iseng	MPI_Isend MPI_Waitall	Aler Barrier
	Process 31	MP_Isend		MPI_Wait MPI_Wait	MPI_Barrier	MPP_TSend	MPI_Isend	MPI_Barrier





\_ # ×

### Indirect data access MPIOM: baroclinic kernel, example





### **Performance Visualization** (MPIOM: selected 3d-loop: iteration cost)





### **COSMOS runtime with improved communication**

