On the best use of HPC resources for ensemble climate forecasting with EC-Earth

M. Asif, O. Mula, D. Manubens, V. Guémas,
F.J. Doblas-Reyes

Outline

- HPC Resources Available
- Autosubmit (a tool)
- A Typical Climate Forecasting Experiment
- Scaling EC-Earth v3.1 (Lindgren, PDC)
- Wrapper Performance
- Future Work (Autosubmit)

HPC Resources Available

- The EC-Earth and NEMO models are used at the CFU
- The CFU's internal capability to perform expensive experiments is very limited and has to apply for competitive resources on different HPC platforms
- The CFU has following computing platforms available:
  - Ithaca (IC3): Sun Blade X6270 Servers with Dual Quad-Core Intel Xeon 5570 processors (2.93GHz), Interconnected with Infiniband and Comprises of 384 Cores
  - MareNostrum (BSC): IBM BladeCenter JS21 with IBM Power PC 970MP processors (2.3GHz) and Myrinet Interconnect
  - ECMWF: IBM pSeries Power 575 SMP Servers with IBM POWER6 processors (4.7GHz) and Interconnected by Using Infiniband
  - HECToR: Cray XE6 system,16-core AMD Opteron Interlagos processors (2.3GHz) and Cray Gemini Communication Chips Used as Interconnect
  - Lindgren (PDC): Cray XE6 system with AMD Opteron 12-core "Magny-Cours" processors (2.1GHz) and Cray Gemini Technology Used as Interconnect.

Autosubmit

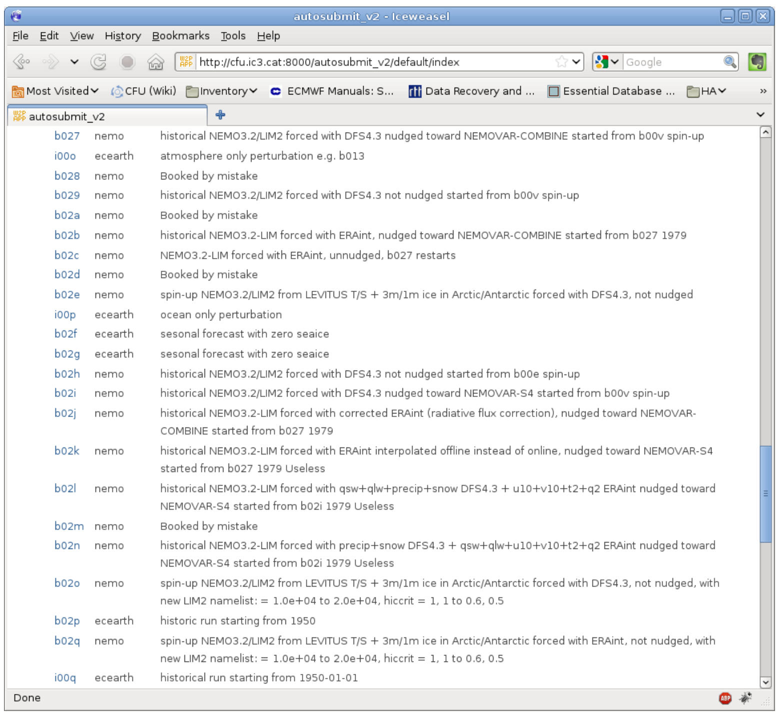
- Autosubmit is a tool developed at CFU using Python (with object-oriented concepts) and SQLite (to document the experiments) to create, manage and monitor experiments
- Interacts with HPC platforms remotely via ssh
- Easily installable on any desktop/server machine with GNU/Linux
- Queuing Systems tested:
  - SGE (Ithaca – IC3)
  - SLURM (MareNostrum - BSC)
  - PBS (HECToR and Lindgren - PDC)

Autosubmit (contd.)

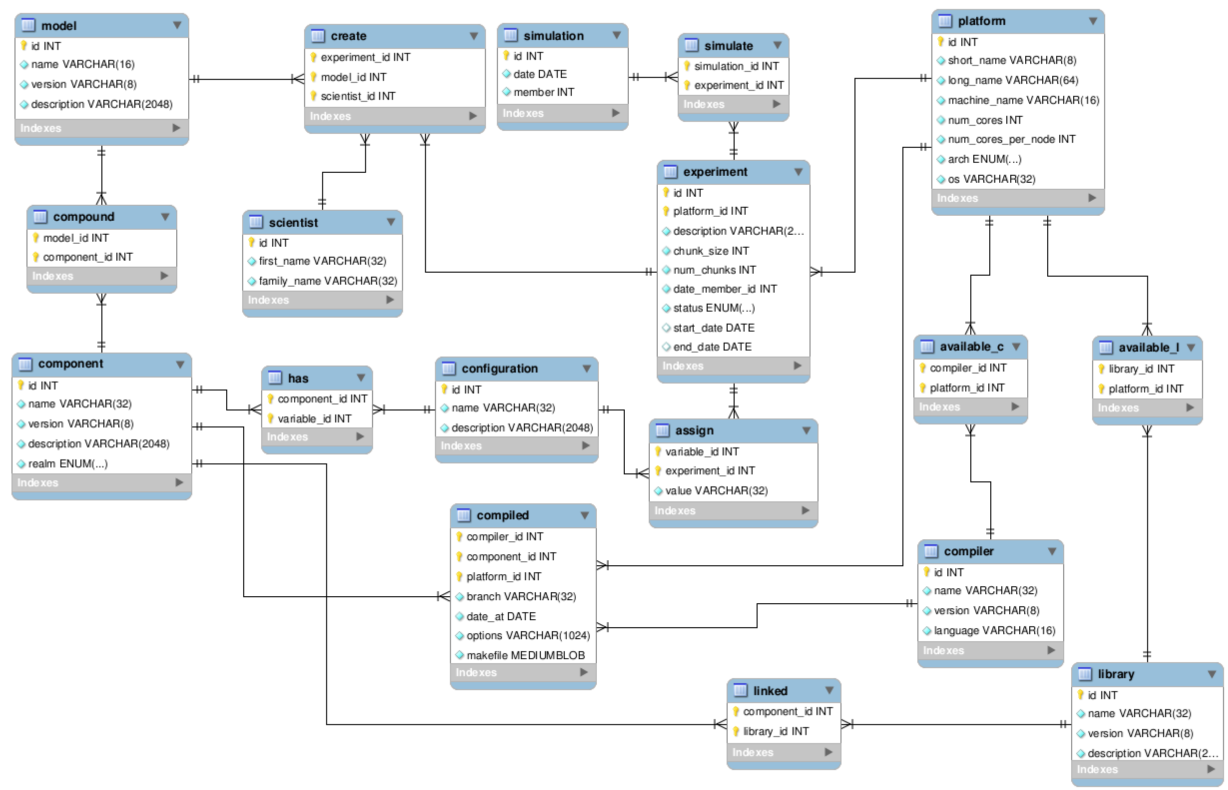
- Major Goals:
  - HPC-independent framework to perform experiments
  - Efficient utilization of available computing resources at HPC's
  - User-friendly interface to start, stop, document and monitor experiments
  - Auto restarting the experiment or some part of experiment (in case of failure)
  - Ability to reproduce the completed experiments either fully or partially
- Why not SMS or ECflow? For increased portability and improved interaction with PRACE/ENES tools (e.g. SAGA) and METAFOR (e.g. CIM)

Web Access and Database Schema

Current Version



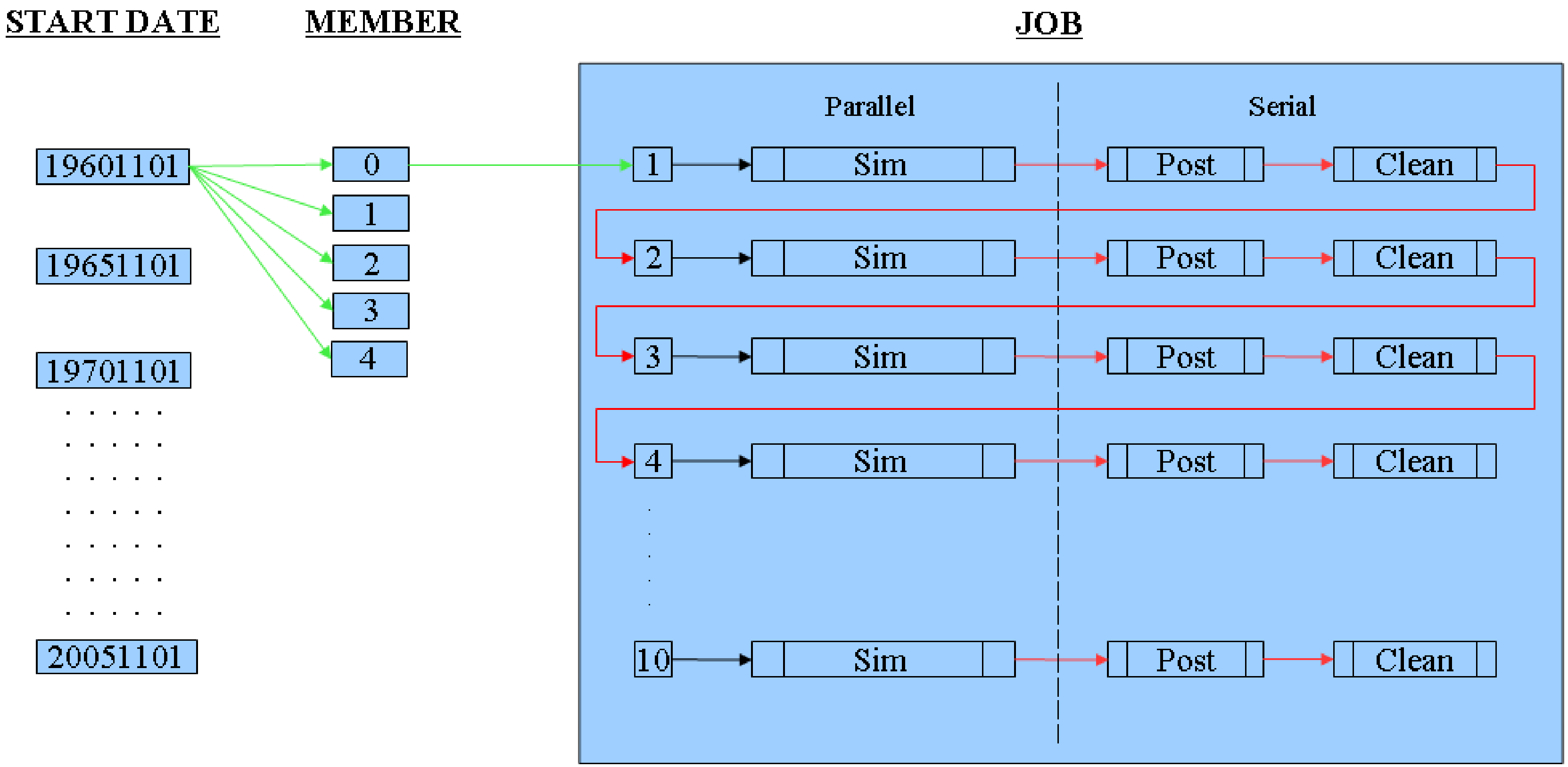
Upcoming Version



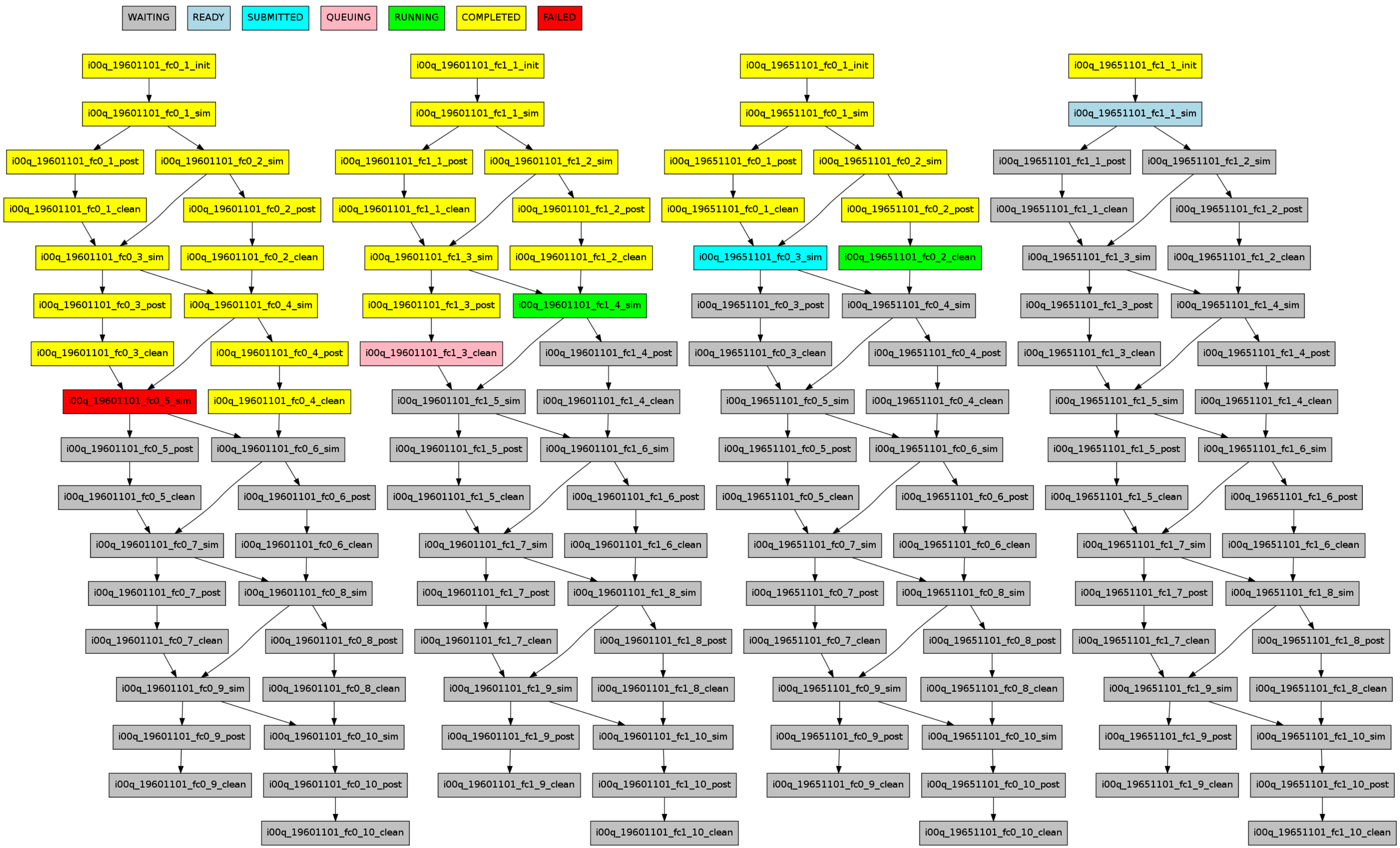
Currently using SQLite and in future shifting to MySQL

A Typical Forecasting Experiment

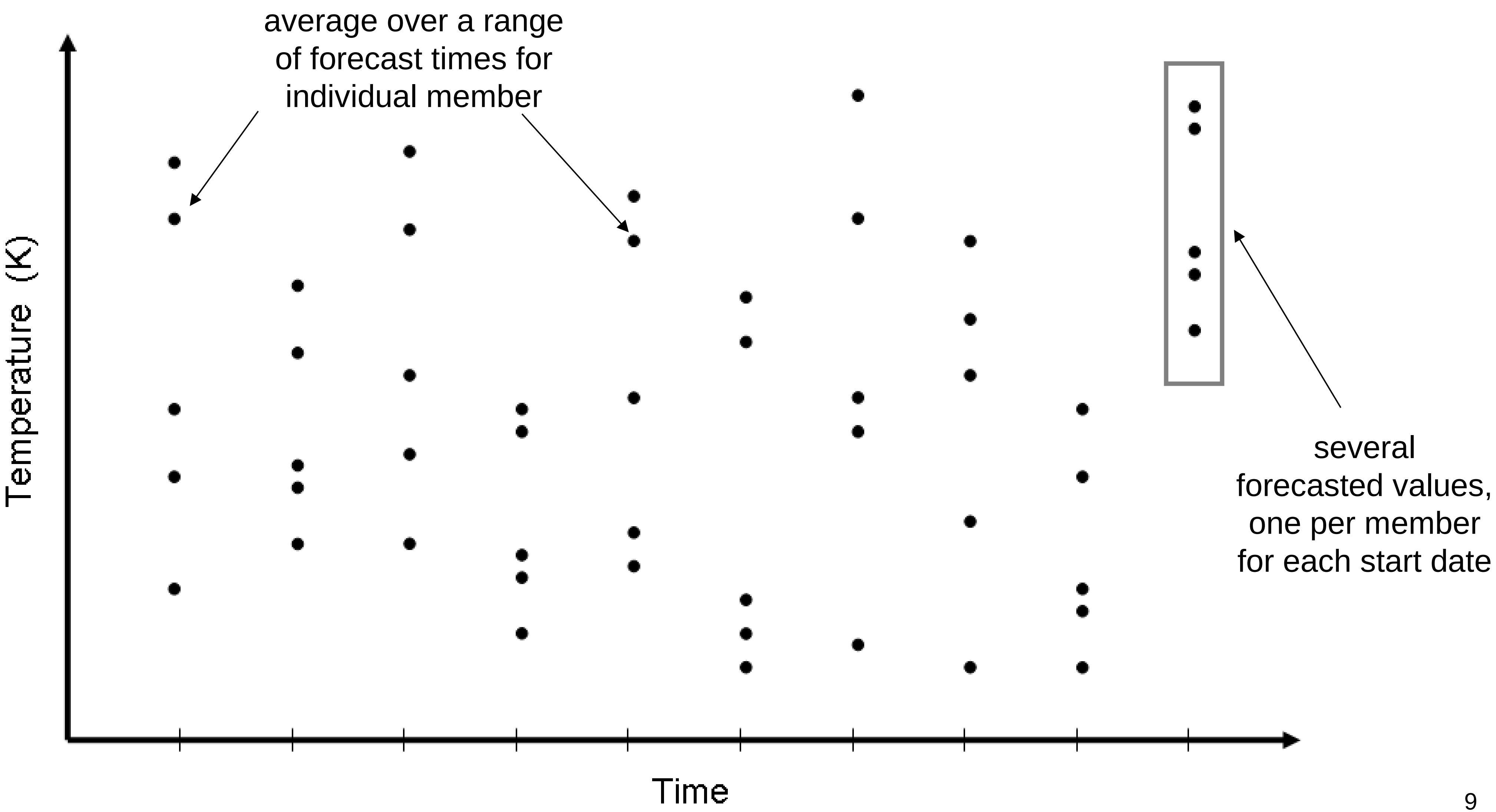
In a typical climate forecast experiment with five members and ten start dates, each start date and member is being run for ten years. Many EC-Earth partners run them using 10 chunks of one year forecast length, with accompanying post-processing and cleaning jobs. The experiment will be made of **50 independent simulations**, each submitting 30 jobs (10 simulations, 10 post-processing and 10 cleaning) with specific dependencies between them.



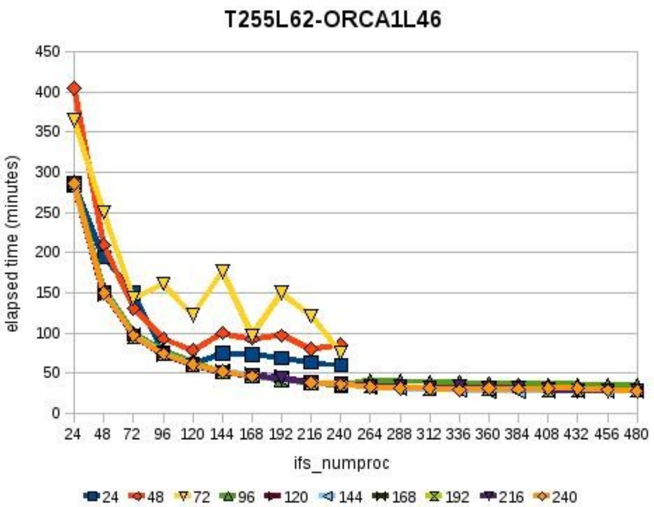
Autosubmit Graphical Monitoring

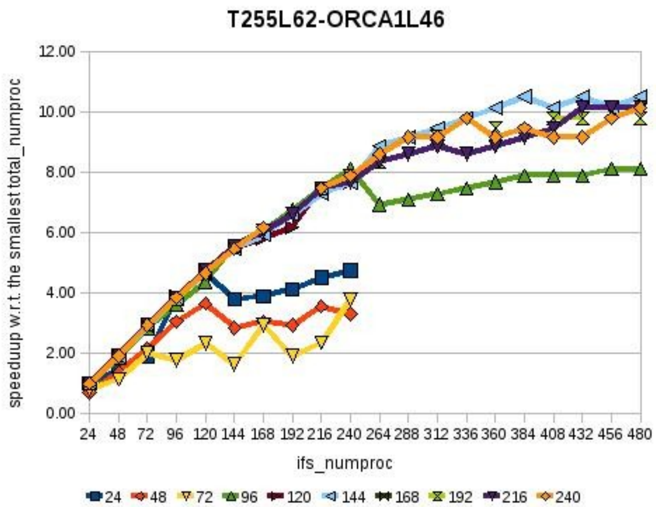


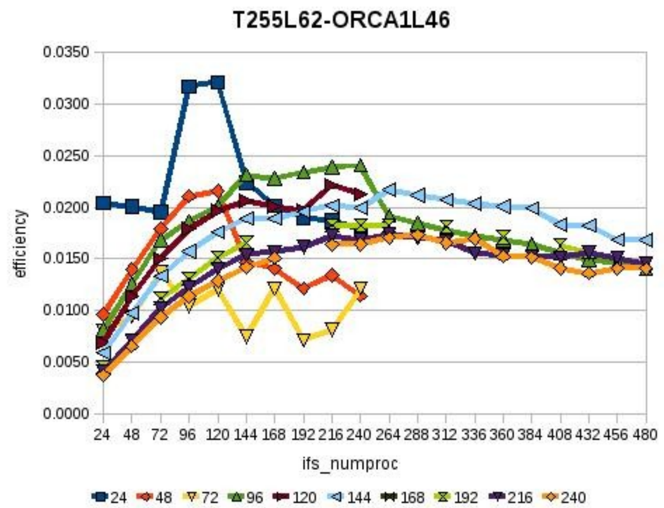
e.g.: 10 Start Dates and 5 Members

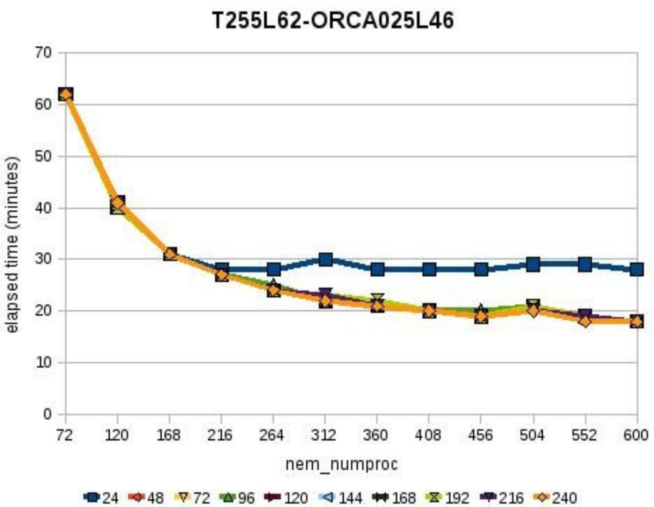


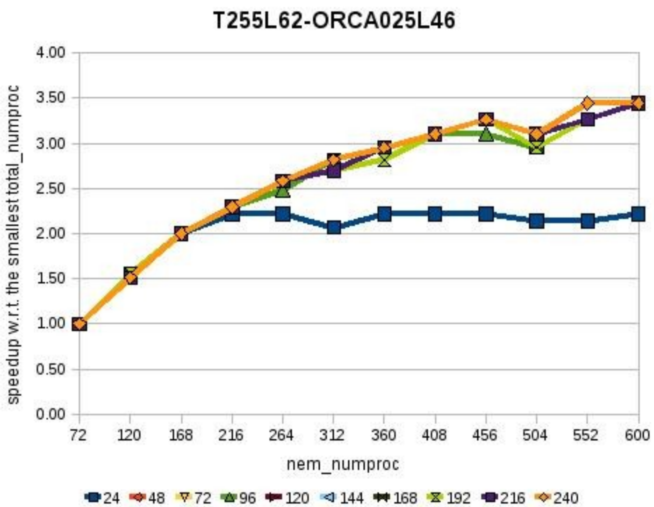
Scaling EC-Earth v3.1 (Lindgren, PDC)

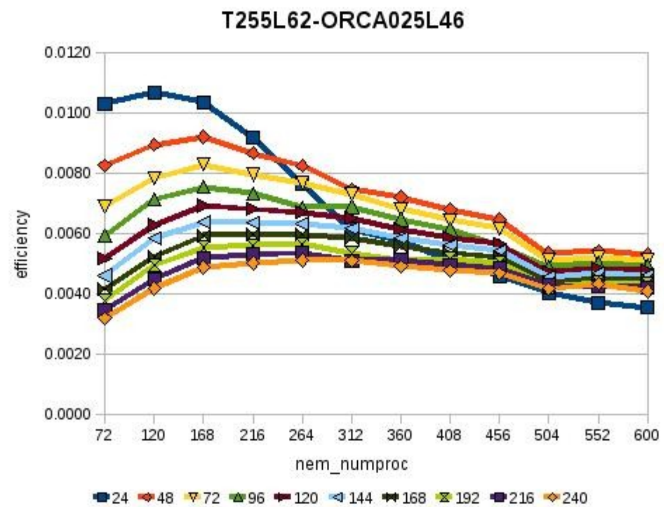






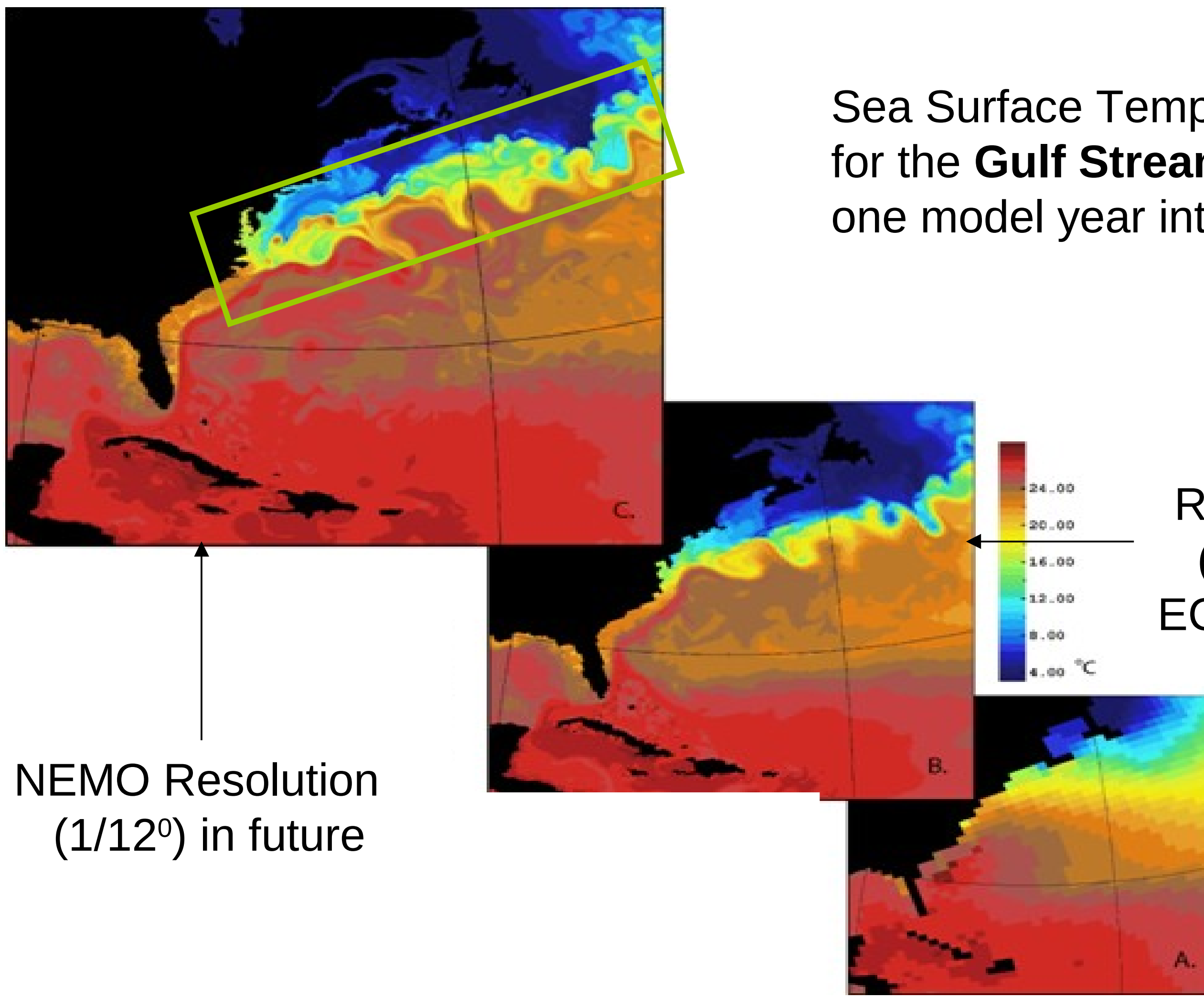






Increasing Climate Model Resolution

Sea Surface Temperature (SST) for the **Gulf Stream region** after one model year integration



NEMO Resolution (1/4^0) in EC-Earth v3

NEMO Resolution (1^0) in EC-Earth v2

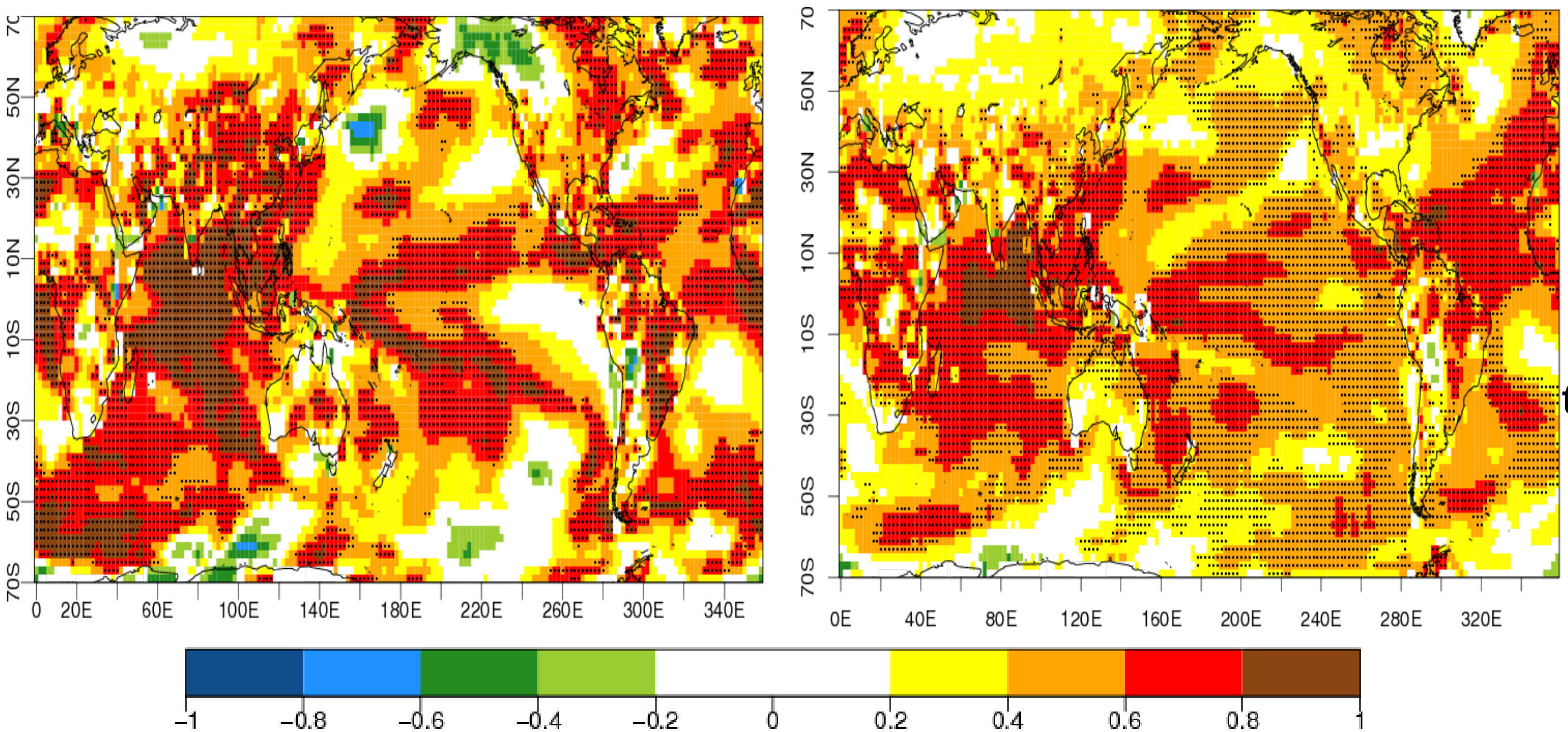
NEMO Resolution (1/12^0) in future

http://noc.ac.uk/science-technology/climate-sea-level/changing-circulation/high-resolution-global-modelling

Increasing Start Date Numbers

Performance (measured by ensemble-mean correlation) of EC-Earth v2.3 for near-surface air temperature during the first forecast year.
Verification Data: combination of GHCN, ERSSTv3b and GISS

10 start dates: every 5 years over the period 1960-2005



46 start dates: every year over the period 1960-2005

Black Dots: correlation significant at the 95% level

Why Need More Resources?

- To increase resolution, no. of start dates and ensemble size
- EC-Earth could be scaled efficiently up to a few hundred cores
  - As far as IFS is concerned*:
    - I/O and/or Coupling could be bottleneck; if scaled beyond 2000 cores
    - Two “MPI_Alltoallv” (Transpose Data) calls at every time step
    - BLAS routine “DGEMM” (Direct Legendre Transform)
- How to run EC-Earth at HPC’s with restrictions of minimum scalability?
  - PRACE tier-0 machines: **8,192** cores at JUGENE, **4,096** at SuperMUC, **2,048** at HERMIT and MareNostrum
  - US DOE INCITE project: **60,000** cores at Oak Ridge Leadership Computing Facility (OLCF)
- Options are being explored: Wrapping many independent simulations (start dates or ensemble members) and run as **“a big single job”**

* http://www.prace-project.eu (PRRACE/IS-ENES collaboration)

Why Need More Resources? (contd.)

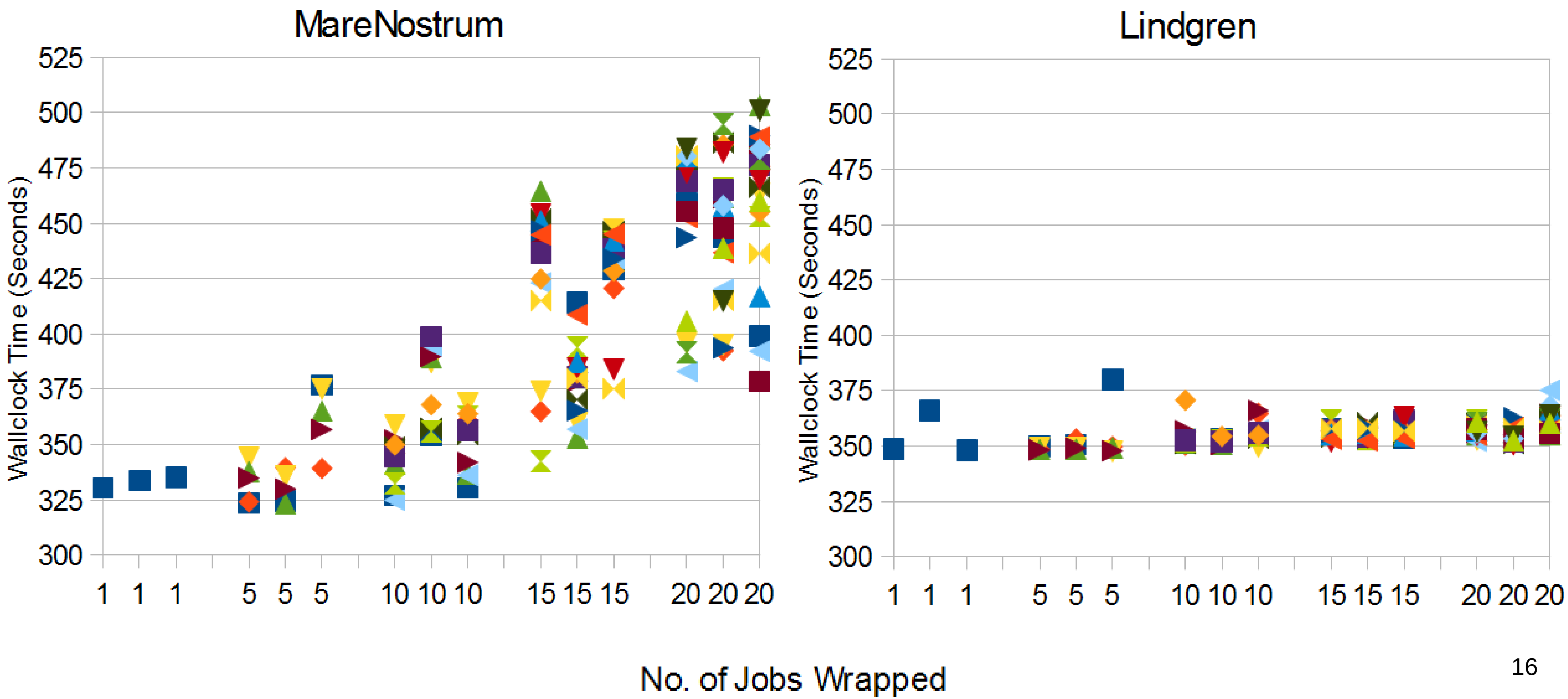
EC-Earth v3.1 (IFS+OASIS+NEMO) at Lindgren, PDC						
No. of Start Dates		1	5	10	10	20
No. of Members		1	5	5	10	10
No. of Independent Simulation Jobs		1	25	50	100	200
T255L62-ORCA1L46	Cores (10+1+4)*24	360	9,000	18,000	36,000	72,000
	Wallclock Time (Hours) / 1y	5	5	5	5	5
	CPU Time (Hours) 1y	1,800	45,000	90,000	180,000	360,000
	Output Size (GB) / 1y	48	1,200	2,400	4,800	9,600
	I/O					
T255L62-ORCA025L46	Cores (2+1+15)*24	432	10,800	21,600	43,200	86,400
	Wallclock Time (Hours) / 1y	25	25	25	25	25
	CPU Time (Hours) / 1y	10,800	270,000	540,000	1,080,000	2,160,000
	Output Size (GB) / 1y	1,176	29,400	58,800	117,600	235,200
	I/O					
T799L62-ORCA025L46	Cores (30+1+15)*24	1,104	27,600	55,200	110,400	220,800
	Wallclock Time (Hours) / 1y	40	40	40	40	40
	CPU Time (Hours) / 1y	44,160	1,104,000	2,208,000	4,416,000	8,832,000
	Output Size (GB) / 1y	1,800	45,000	90,000	180,000	360,000
	I/O					

Wrapper Performance

- MareNostrum
  - EC-Earth v2.3
  - T159L62-ORCA1
  - Store Output at Disk After 6h
  - Coupling Frequency 3h
  - Forecast Length 2 days
  - I/O Time Step 0,3,6,…
  - Total NPROC 45 per Single Independent Simulation
- Lindgren
  - EC-Earth v3.1
  - T255L62-ORCA1
  - Store Output at Disk After 6h
  - Coupling Frequency 3h
  - Forecast Length 10 days
  - I/O Time Step 0,4,8,…
  - Total NPROC 360 per Single Independent Simulation

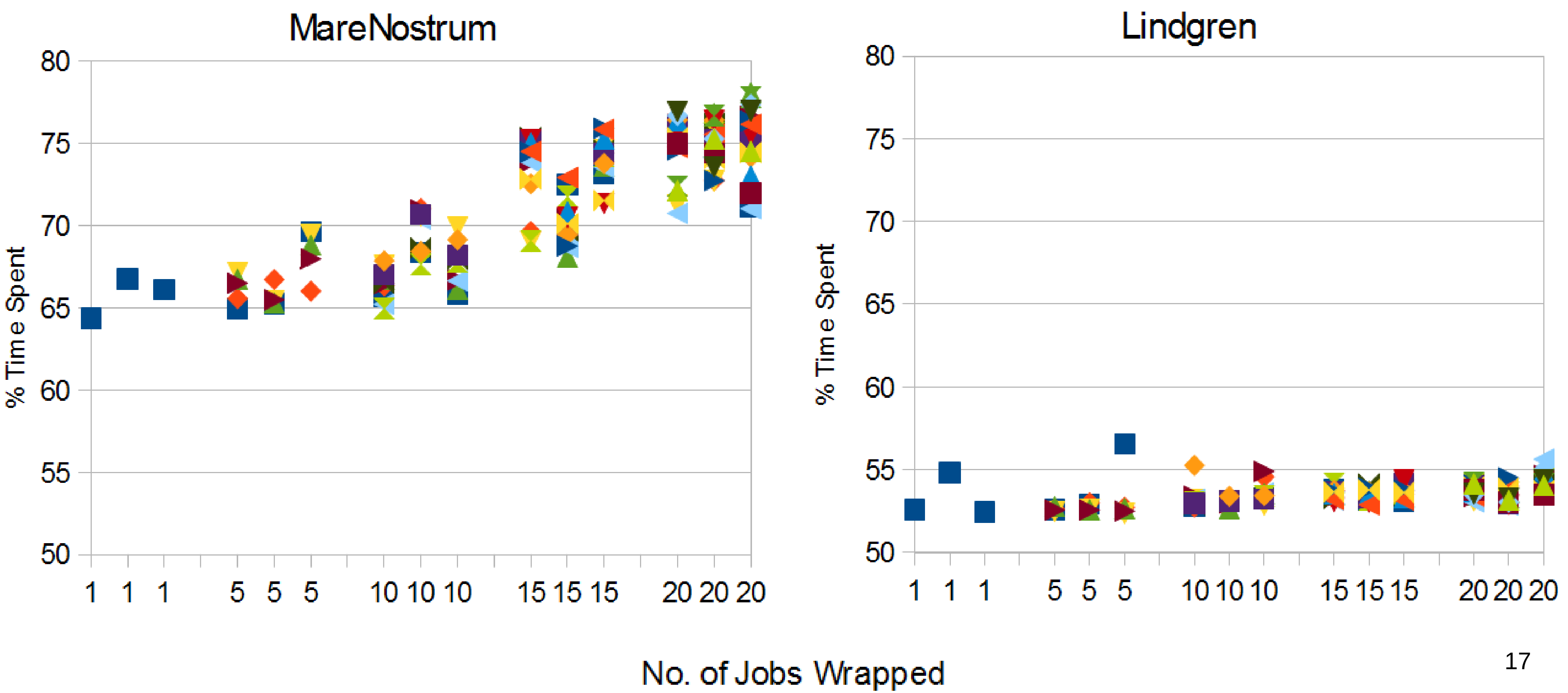
Wrapper Performance (contd.)

A “big single job” was run by increasing no. of wrapped jobs from 1, 5, 10, … 20.
Each “big single job” was run thrice and **wallclock time** (elapsed time) is plotted.
The range of colors depicted below cover the range of wrapped jobs.



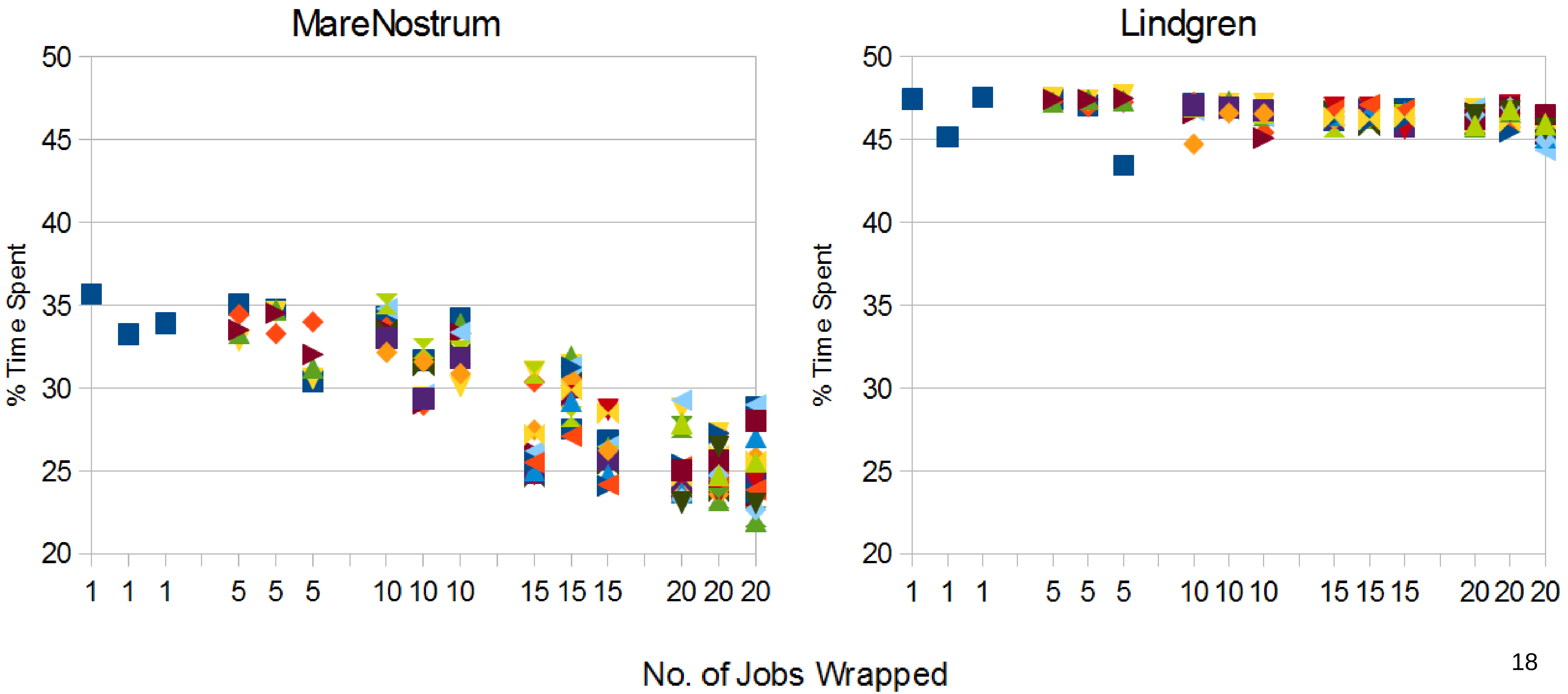
Wrapper Performance (contd.)

The % time spent for **I/O time step** is plotted below against increasing no. of wrapped jobs.



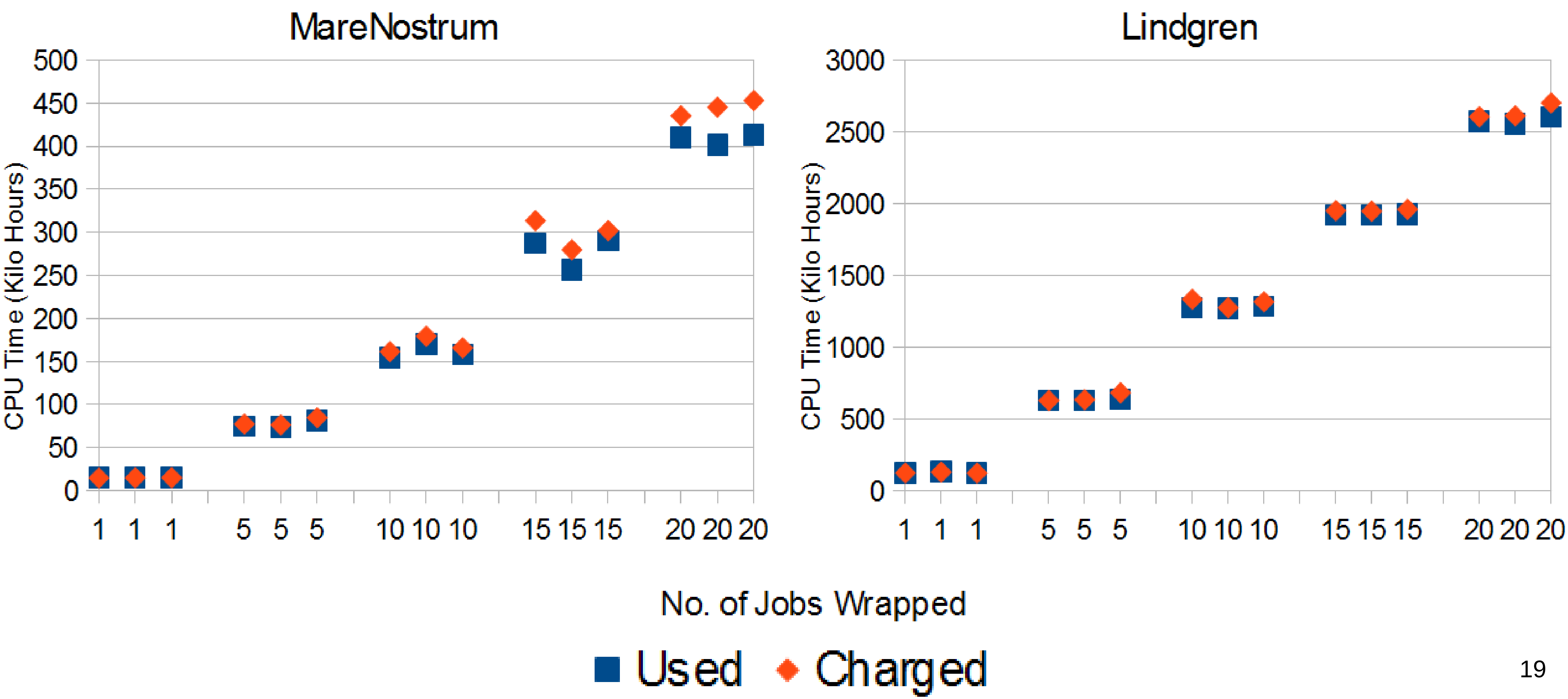
Wrapper Performance (contd.)

The % time spent for **non I/O time step** is plotted below against increasing no. of wrapped jobs.



Wrapper Performance (contd.)

CPU time used is determined by adding the wallclock times used by all wrapped jobs per "big single job" and **CPU time charged** is determined on the basis of the slowest job among wrapped jobs.



Future Work (Autosubmit)

- Explore options to implement wrapper to ensemble simulation jobs, this piece of work will be done by IC3 under IS-ENES2
- Integration of HPC's using SAGA (Simple API for Grid Applications)
  - BLISS-SAGA (a light-weight implementation for Python) comply with OGF (Open Grid Forum) standards (how to interact with the middleware)
  - A number of adaptors are already implemented, to support different grid and cloud computing backends
  - SAGA provides units to compose high-level functionality across distinct distributed systems (e.g. submit jobs from same experiment to different platforms)
- Documenting experiments on simplified METAFOR standards by using relational databases (MySQL)
- Designing a web front-end for experiment creation and monitoring
- Storing user-defined job dependency tree in XML Scheme file
- Installation package and open source license