

# On use of HPC resources for ensemble climate forecasting

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 (Lindgren, PDC)
- Wrapper Performance
- Future Work (Autosubmit)

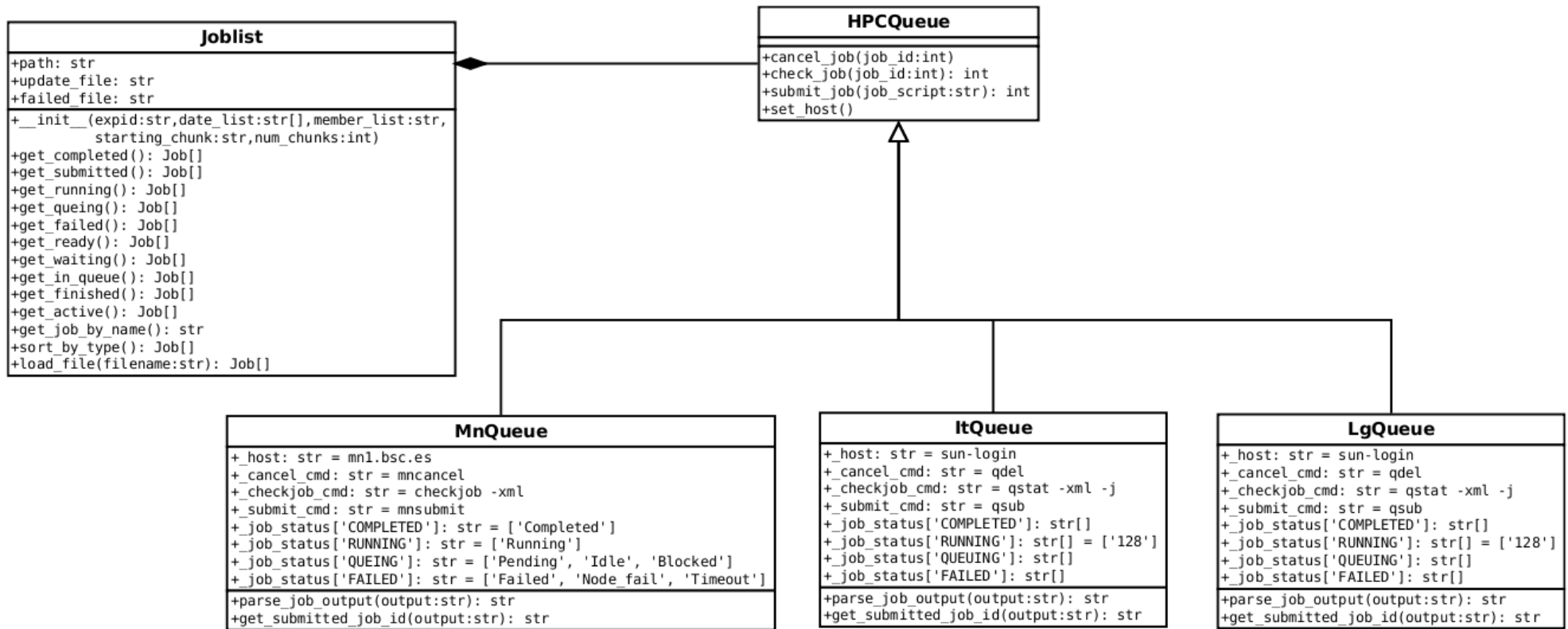
# HPC Resources Available

- The EC-Earth (Earth System Model) and NEMO (state of the art oceanographic model) 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.)

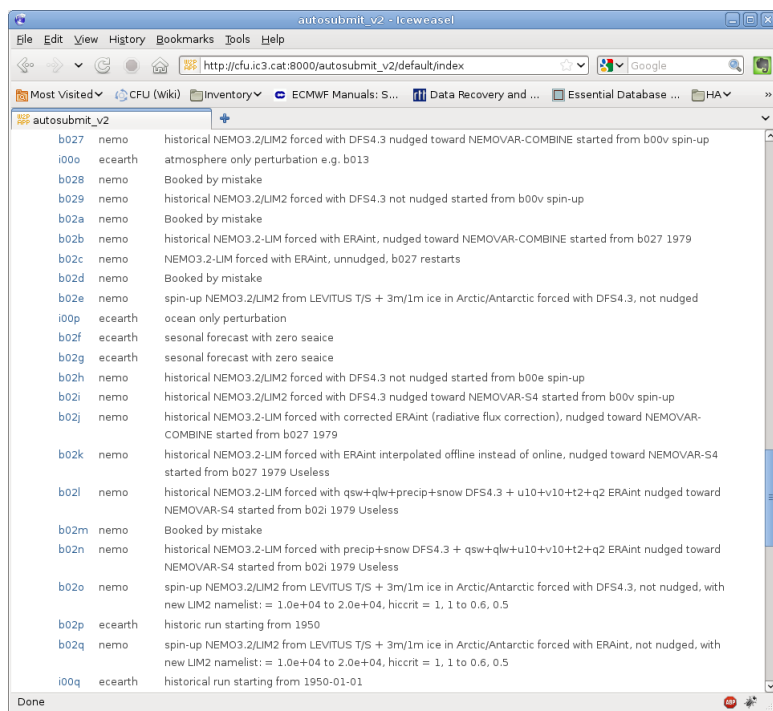


# Autosubmit (contd.)

- Major Goals:
  - HPC-independent framework to perform experiments
  - Efficient utilization of available computing resources at HPC's
  - User-friendly interface to start, stop and monitor experiments
  - Auto restarting the experiment or some part of experiment
  - Ability to reproduce the completed experiments
- Why not SMS or ecFlow (developed at ECMWF)?
  - Increased portability
  - Improved interaction
    - PRACE/ENES tools (e.g. SAGA)
    - METAFOR (e.g. CIM)

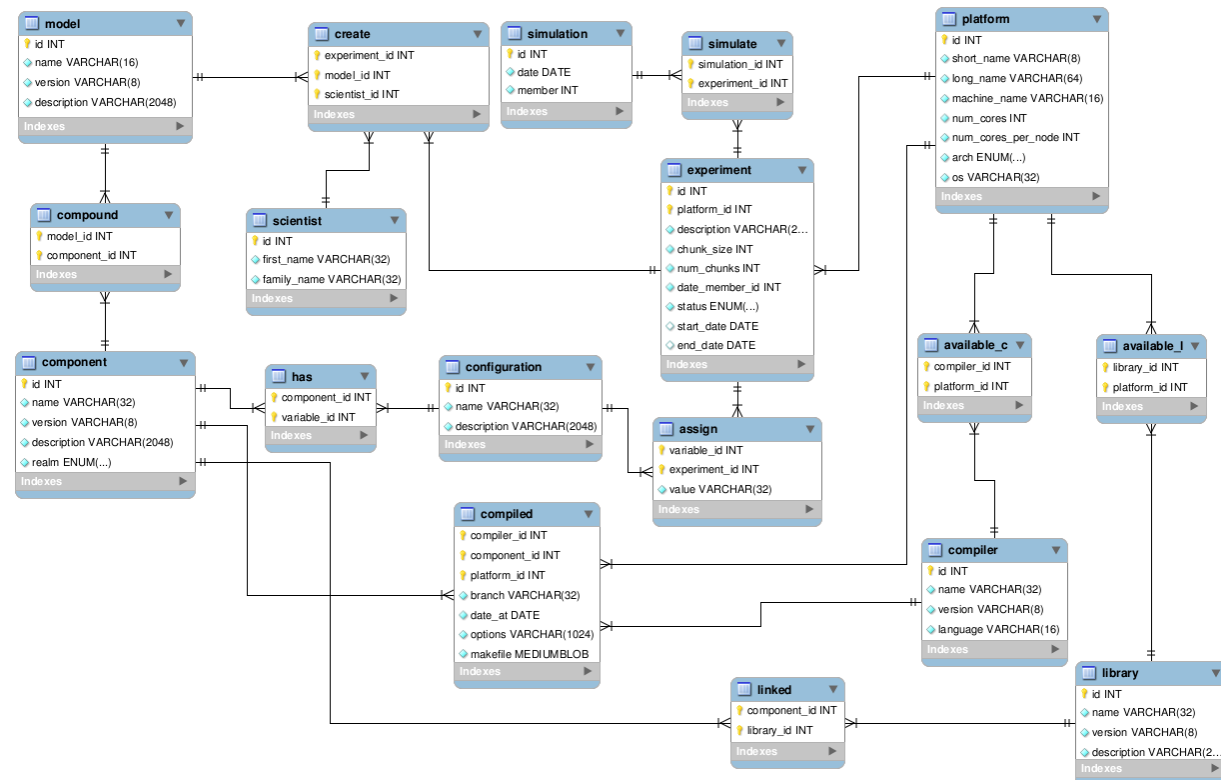
# Web Access and Database Schema

## Current Version



id	model	description
b027	nemo	historical NEMO3.2/LIM2 forced with DFS4.3 nudged toward NEMOVAR-COMBINE started from b00v spin-up
i00a	ecearth	atmosphere only perturbation e.g. b013
b028	nemo	Booked by mistake
b029	nemo	historical NEMO3.2/LIM2 forced with DFS4.3 not nudged started from b00v spin-up
b02a	nemo	Booked by mistake
b02b	nemo	historical NEMO3.2-LIM forced with ERAInt, nudged toward NEMOVAR-COMBINE started from b027 1979
b02c	nemo	NEMO3.2-LIM forced with ERAInt, unnudged, b027 restarts
b02d	nemo	Booked by mistake
b02e	nemo	spin-up NEMO3.2/LIM2 from LEVITUS TJS + 3m/1m ice in Arctic/Antarctic forced with DFS4.3, not nudged
i00p	ecearth	ocean only perturbation
b02f	ecearth	seasonal forecast with zero seaice
b02g	ecearth	seasonal forecast with zero seaice
b02h	nemo	historical NEMO3.2/LIM2 forced with DFS4.3 not nudged started from b00e spin-up
b02i	nemo	historical NEMO3.2/LIM2 forced with DFS4.3 nudged toward NEMOVAR-S4 started from b00v spin-up
b02j	nemo	historical NEMO3.2-LIM forced with corrected ERAInt (radiative flux correction), nudged toward NEMOVAR-COMBINE started from b027 1979
b02k	nemo	historical NEMO3.2-LIM forced with ERAInt interpolated offline instead of online, nudged toward NEMOVAR-S4 started from b027 1979 Useless
b02l	nemo	historical NEMO3.2-LIM forced with qsw+qlw+precip+snow DFS4.3 + u10+v10+H2+q2 ERAInt nudged toward NEMOVAR-S4 started from b02i 1979 Useless
b02m	nemo	Booked by mistake
b02n	nemo	historical NEMO3.2-LIM forced with precip+snow DFS4.3 + qsw+qlw+u10+v10+H2+q2 ERAInt nudged toward NEMOVAR-S4 started from b02i 1979 Useless
b02o	nemo	spin-up NEMO3.2/LIM2 from LEVITUS TJS + 3m/1m ice in Arctic/Antarctic forced with DFS4.3, not nudged, with new LIM2 namelist: = 1.0e+04 to 2.0e+04, hicrit = 1, 1 to 0.6, 0.5
b02p	ecearth	historic run starting from 1950
b02q	nemo	spin-up NEMO3.2/LIM2 from LEVITUS TJS + 3m/1m ice in Arctic/Antarctic forced with ERAInt, not nudged, with new LIM2 namelist: = 1.0e+04 to 2.0e+04, hicrit = 1, 1 to 0.6, 0.5
i00q	ecearth	historical run starting from 1950-01-01

## Upcoming Version



Currently using SQLite and in future shifting to MySQL

# Creating a new experiment

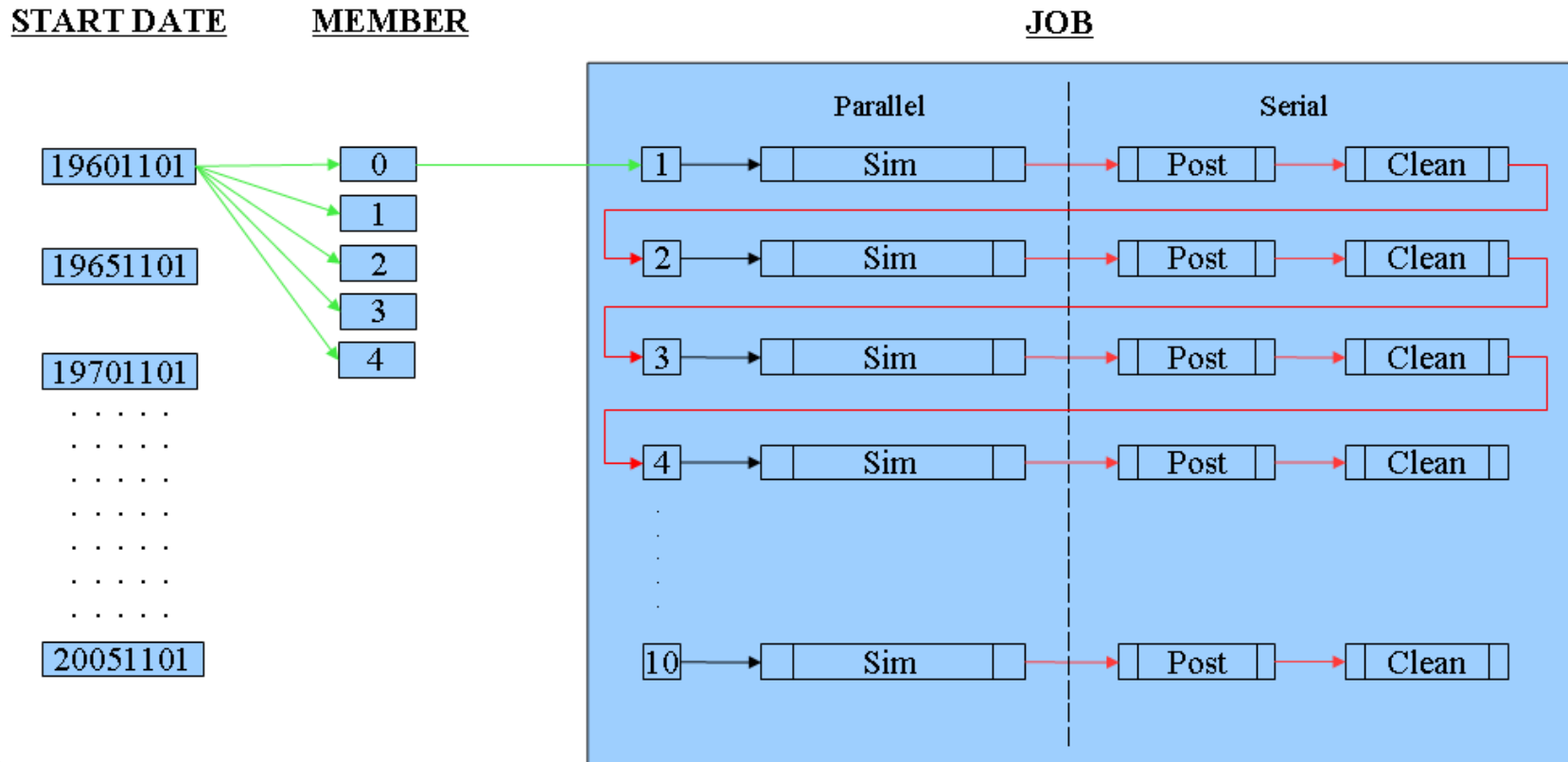
- Get an ID and create the needed directory structure
- Fill the configuration files
  - autosubmit\_i00e.conf
  - expdef\_i00e.conf
- Generate the pickle file
  - python create\_exp.py <expid>

```
i00e
|-- conf
|   |-- archdef_i00e.conf
|   |-- autosubmit_i00e.conf
|   `-- expdef_i00e.conf
|-- pkl
|-- plot
|-- templates
|   |-- template_i00e.clean
|   |-- template_i00e.ini
|   |-- template_i00e.post
|   `-- template_i00e.sim
`-- tmp
```

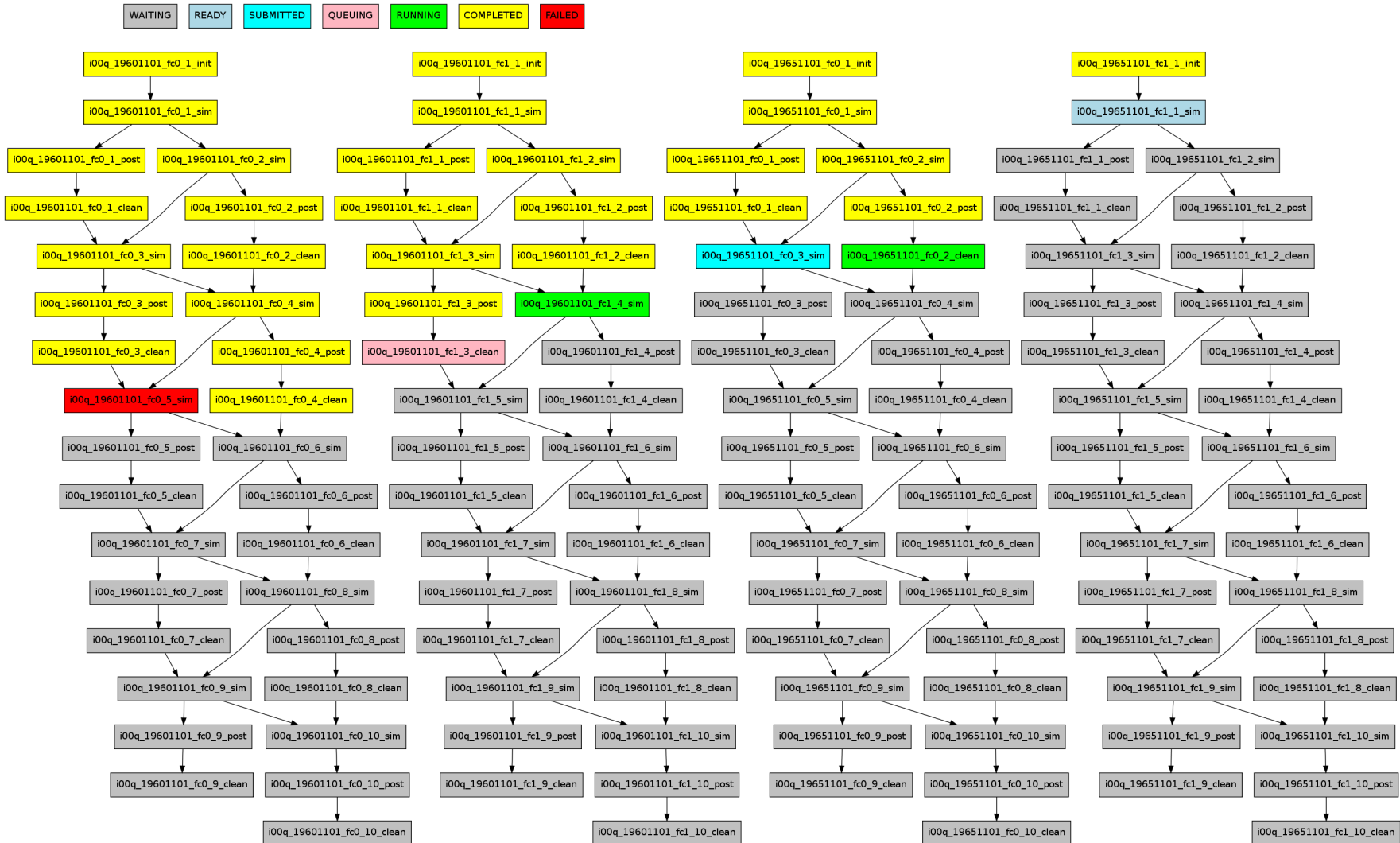


# A Typical Forecasting Experiment

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.

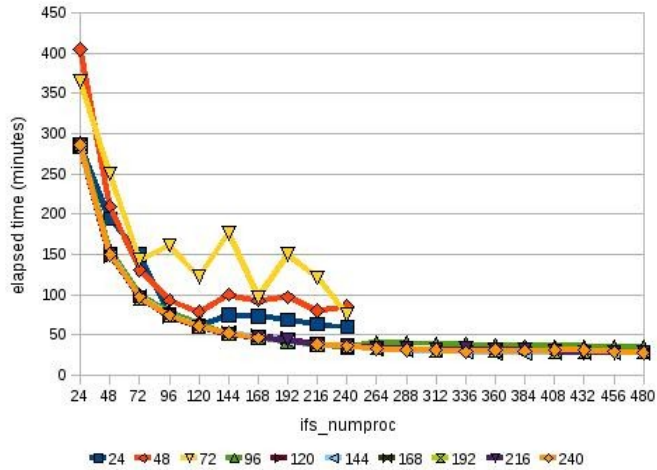


# Autosubmit Graphical Monitoring

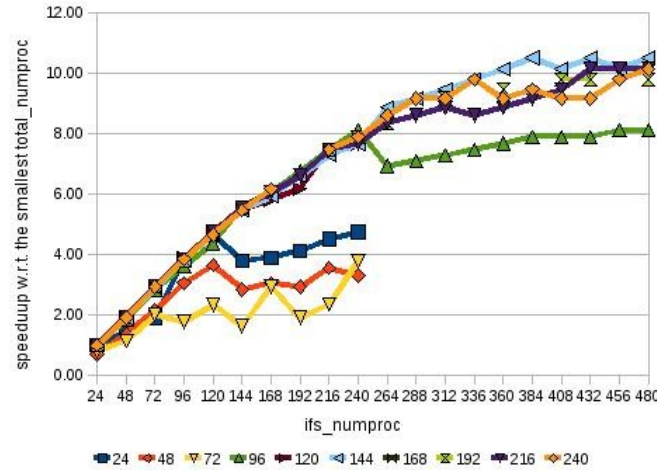


# Scaling EC-Earth v3 (Lindgren, PDC)

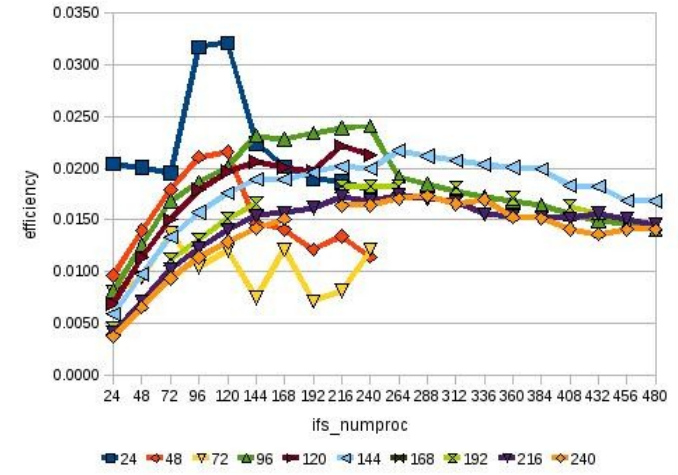
T255L62-ORCA1L46



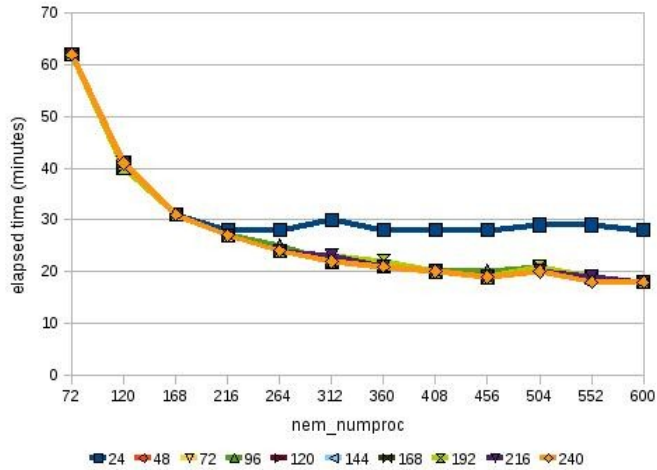
T255L62-ORCA1L46



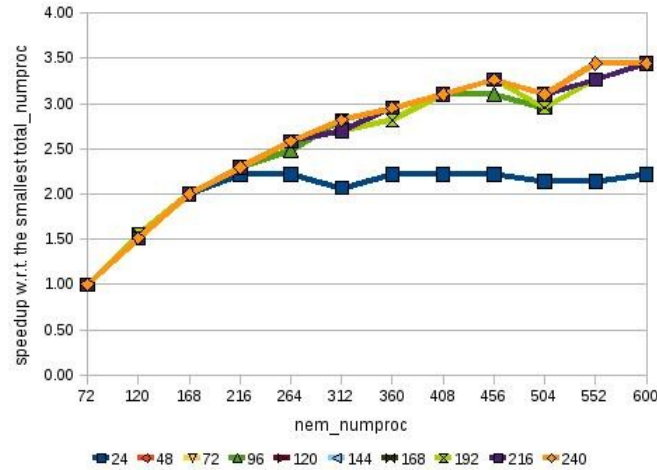
T255L62-ORCA1L46



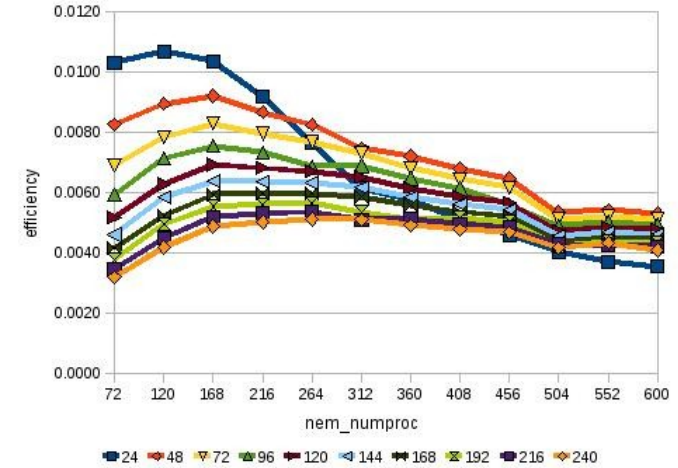
T255L62-ORCA025L46



T255L62-ORCA025L46



T255L62-ORCA025L46



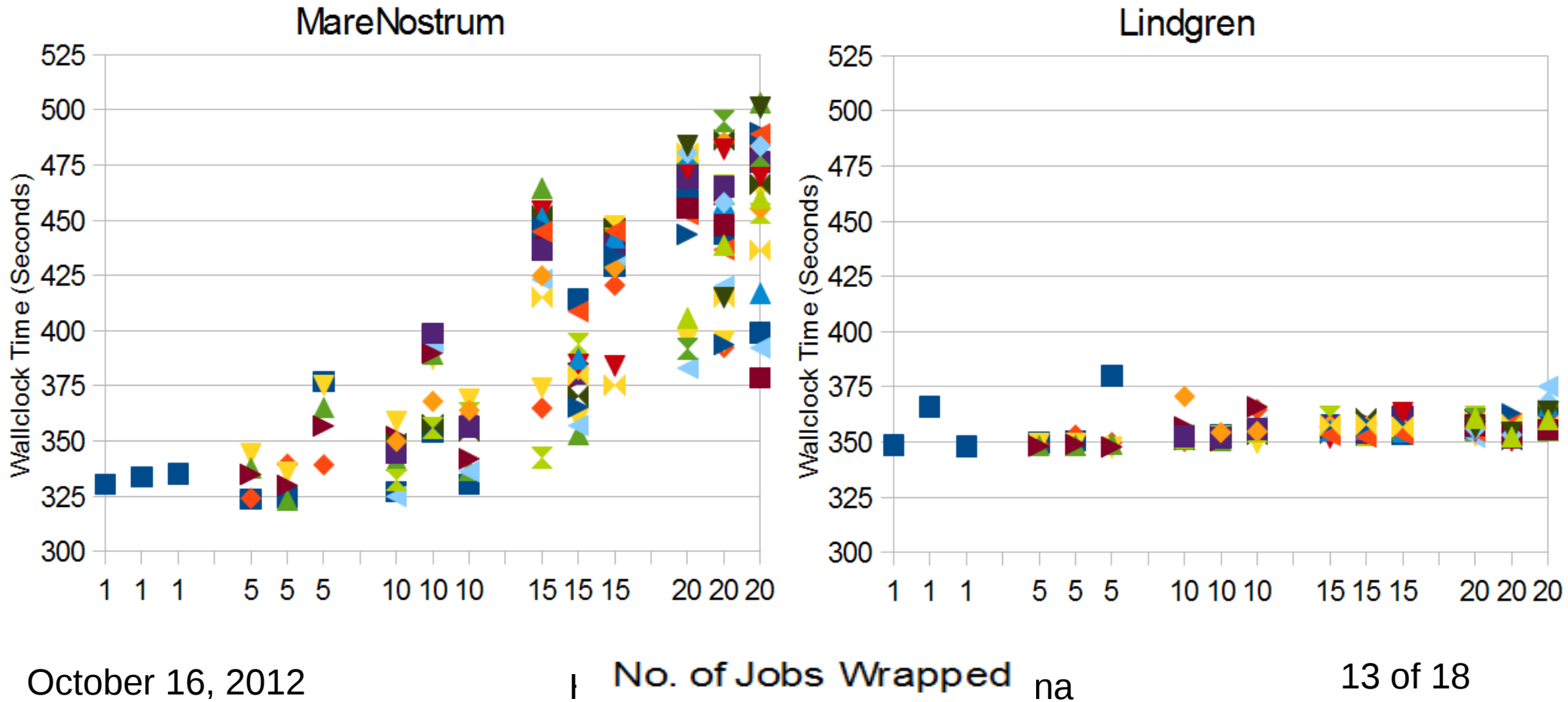
# Why Need More Resources?

- 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)

# Wrapper Performance (contd.)

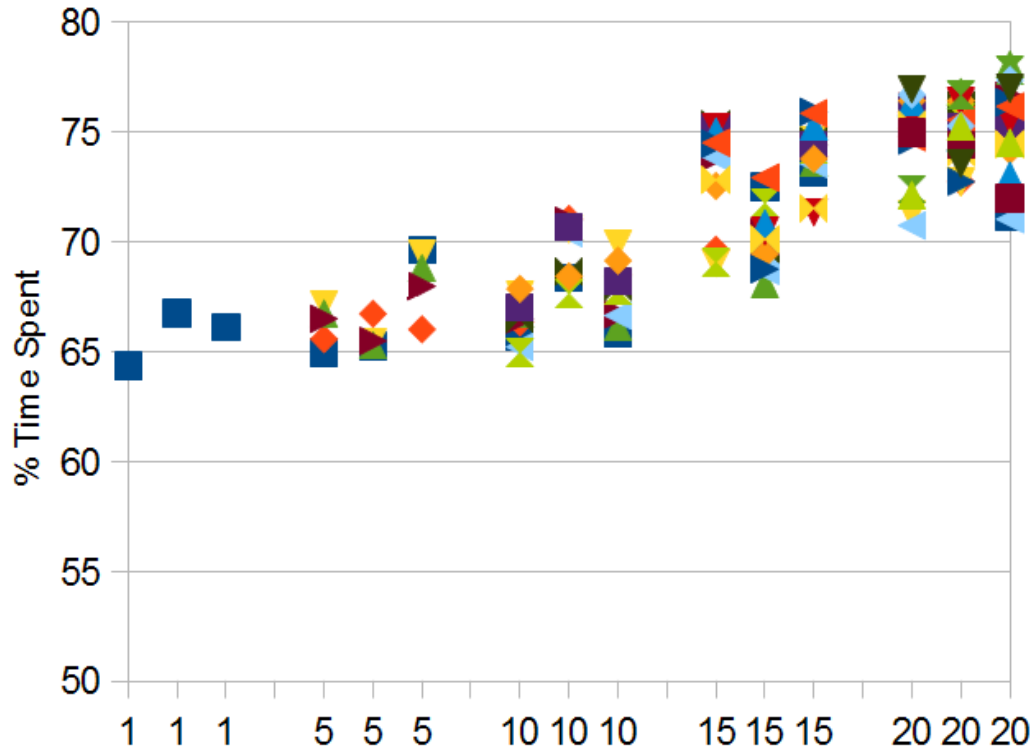
A “big single job” was run by increasing no. of wrapped jobs from 1, 5, 10, ... 20. Each “big single job” was run thrice and **wallclock time** (elapsed time) is plotted. The range of colors depicted below cover the range of wrapped jobs.



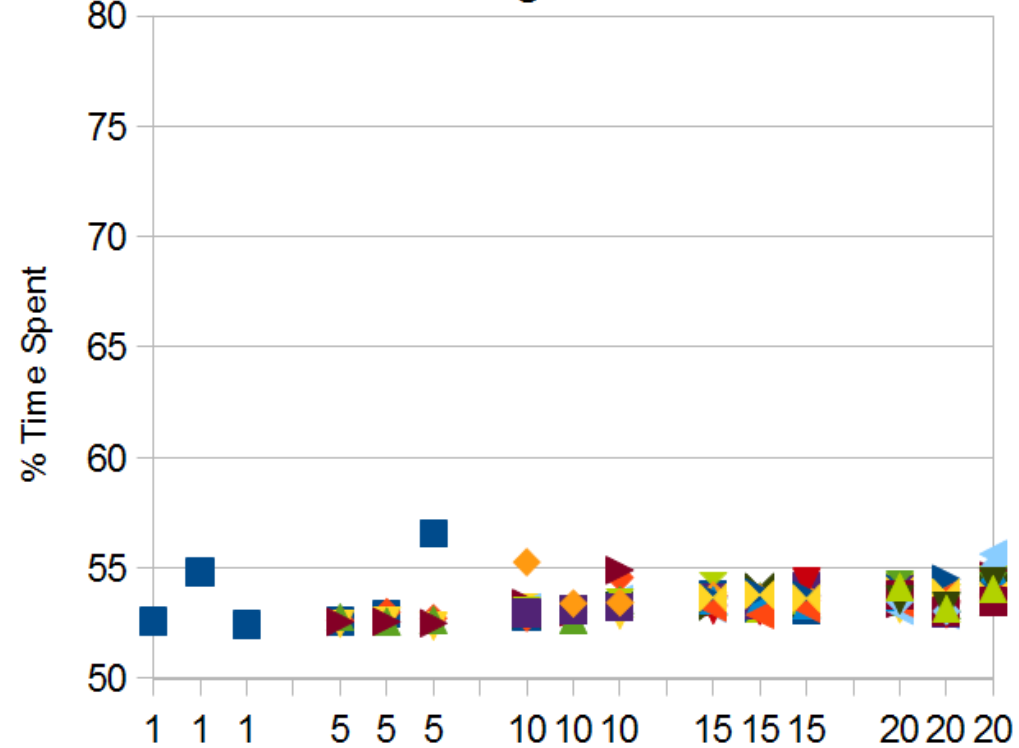
# Wrapper Performance (contd.)

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

MareNostrum



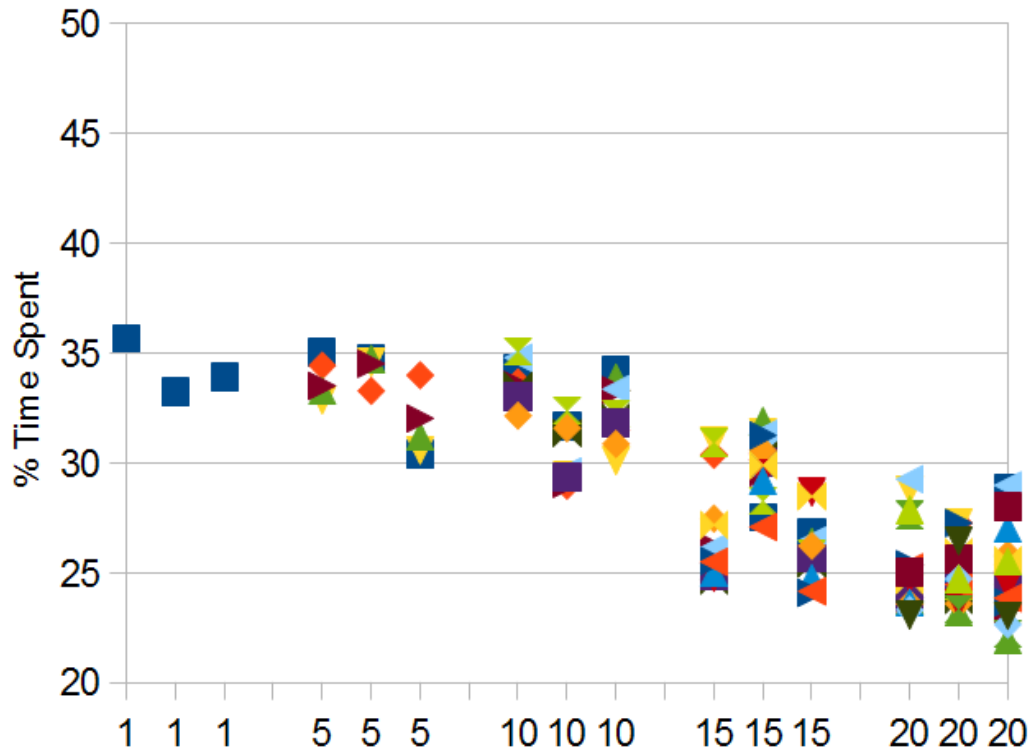
Lindgren



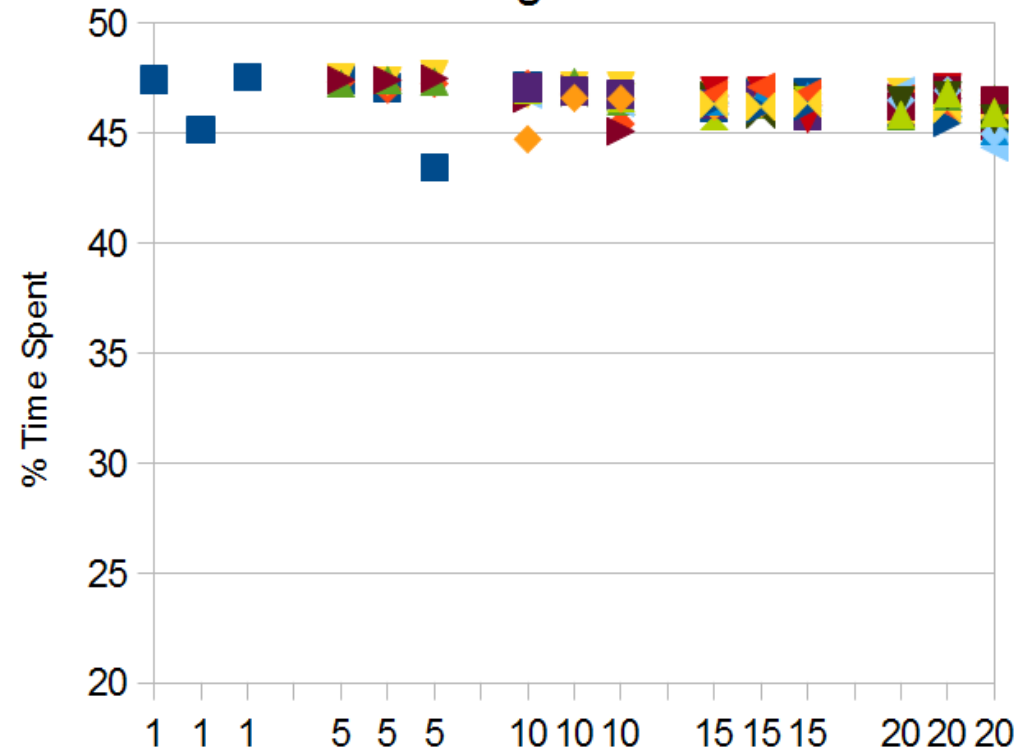
# Wrapper Performance (contd.)

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

MareNostrum



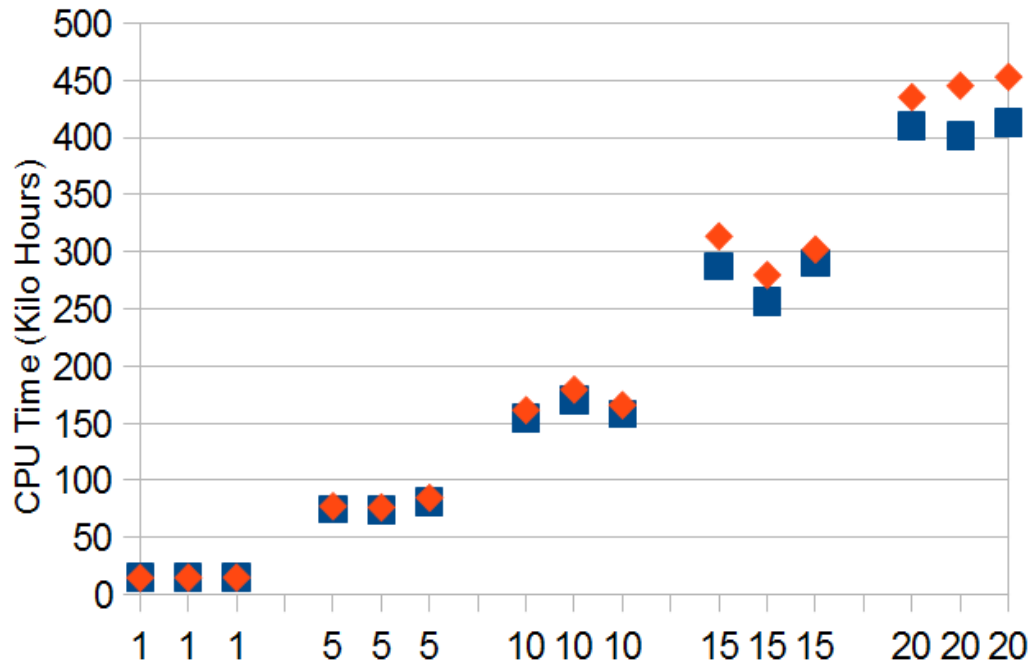
Lindgren



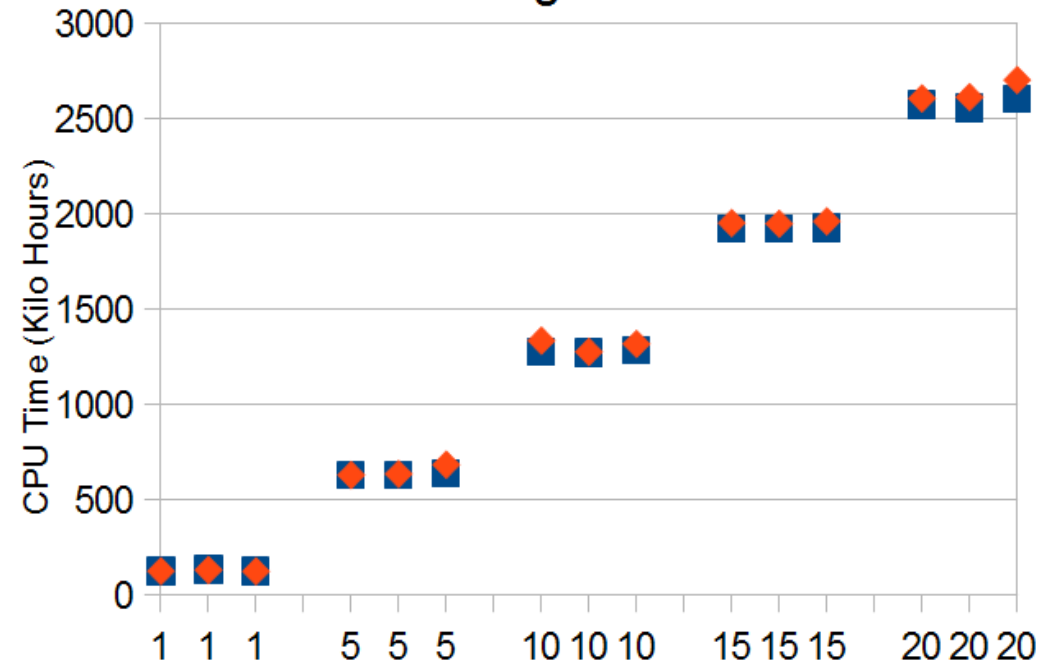
# Wrapper Performance (contd.)

**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.

MareNostrum



Lindgren



No. of Jobs Wrapped

■ Used    ◆ Charged



# 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

