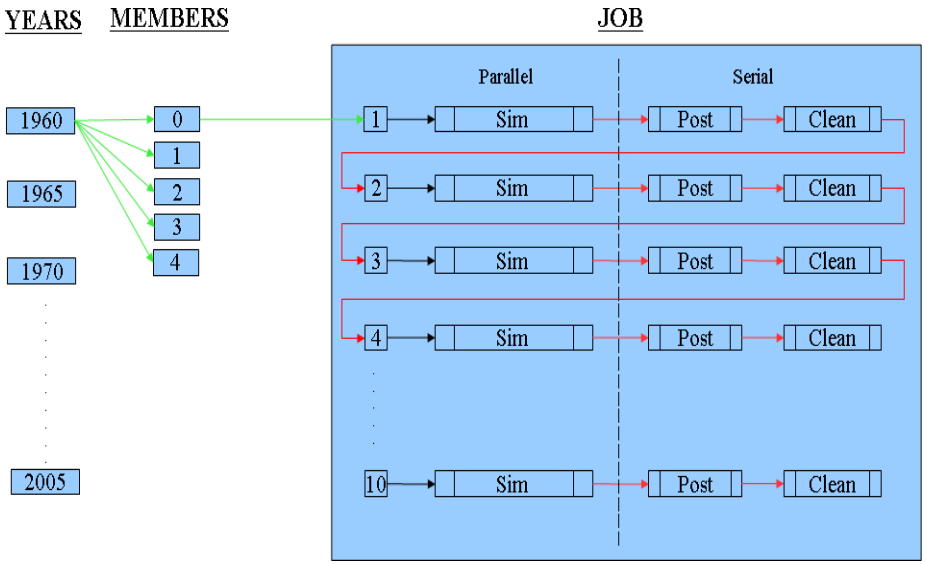
Wrapper to Run Ensemble Climate Forecast Experiments under Autosubmit on Tier-0 Machines

Climate Forecasting Unit

Autosubmit

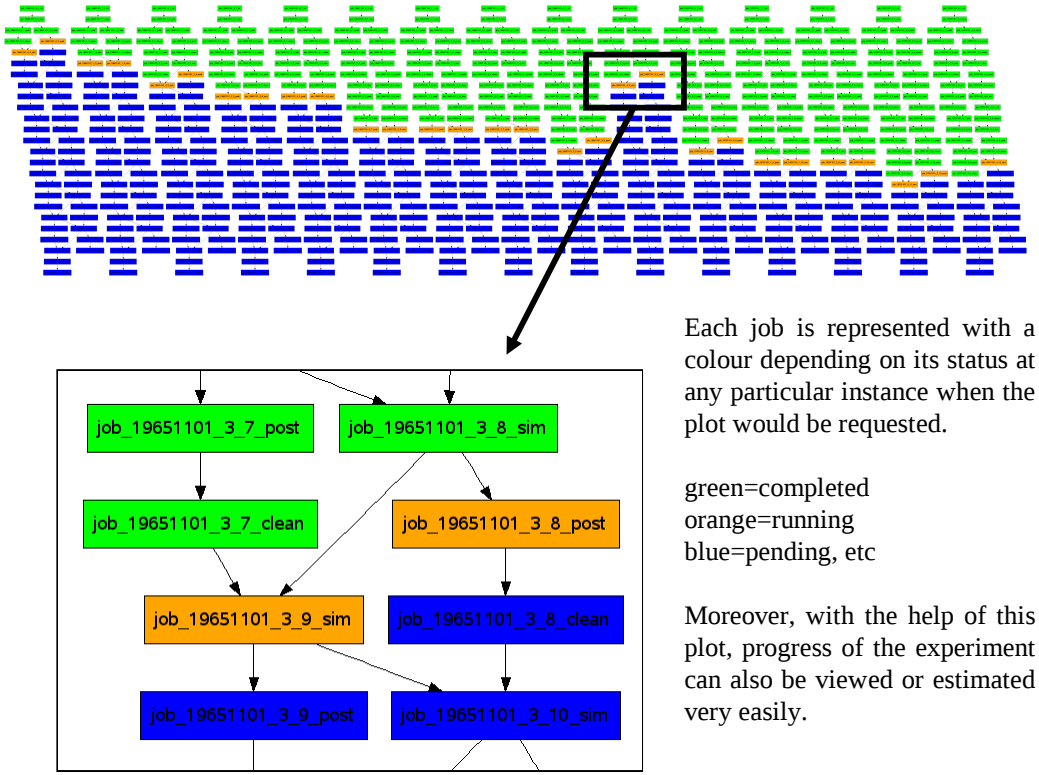
Autosubmit is designed by using Python suite of scripts. It can manage number of experiments with arbitrary number of jobs on any HPC platform.

A typical Expriment: The following schematic experiment consists of 10 starting dates from 1960 to 2005. Each start date contains 5 different members. Each member will be running for 10 years with 10 chunks of one year simulation length. Moreover, every single chunk has 3 different type of jobs: a simulation (Sim), a post-processing (Post) and an archiving and cleaning job (Clean). So, with this typical scenario, one start date with one member will be running 30 jobs and in total for the complete experiment, there will be running 1500 jobs. Therefore to accomplish this task manually requires a lot of time and continuous effort.



The method: Autosubmit acts as a wrapper over the queue system and HPC scheduler. Autosubmit defines the experiment as a sequence of jobs (list of jobs) and after resolving dependencies among different types of jobs, it submits the jobs one by one on queue system. Thereafter it starts continuous monitoring of the queue system after some defined time intervals and perform further necessary actions accordingly.

Live Monitoring: Climate forecast experiments could be of different time length and could be finished in a few days or in a few weeks. Therefore it is extremely essential to have a bird eye view on experiments frequently and conveniently. Autosubmit contains an interesting feature to monitor experiment with ease which can be seen below:



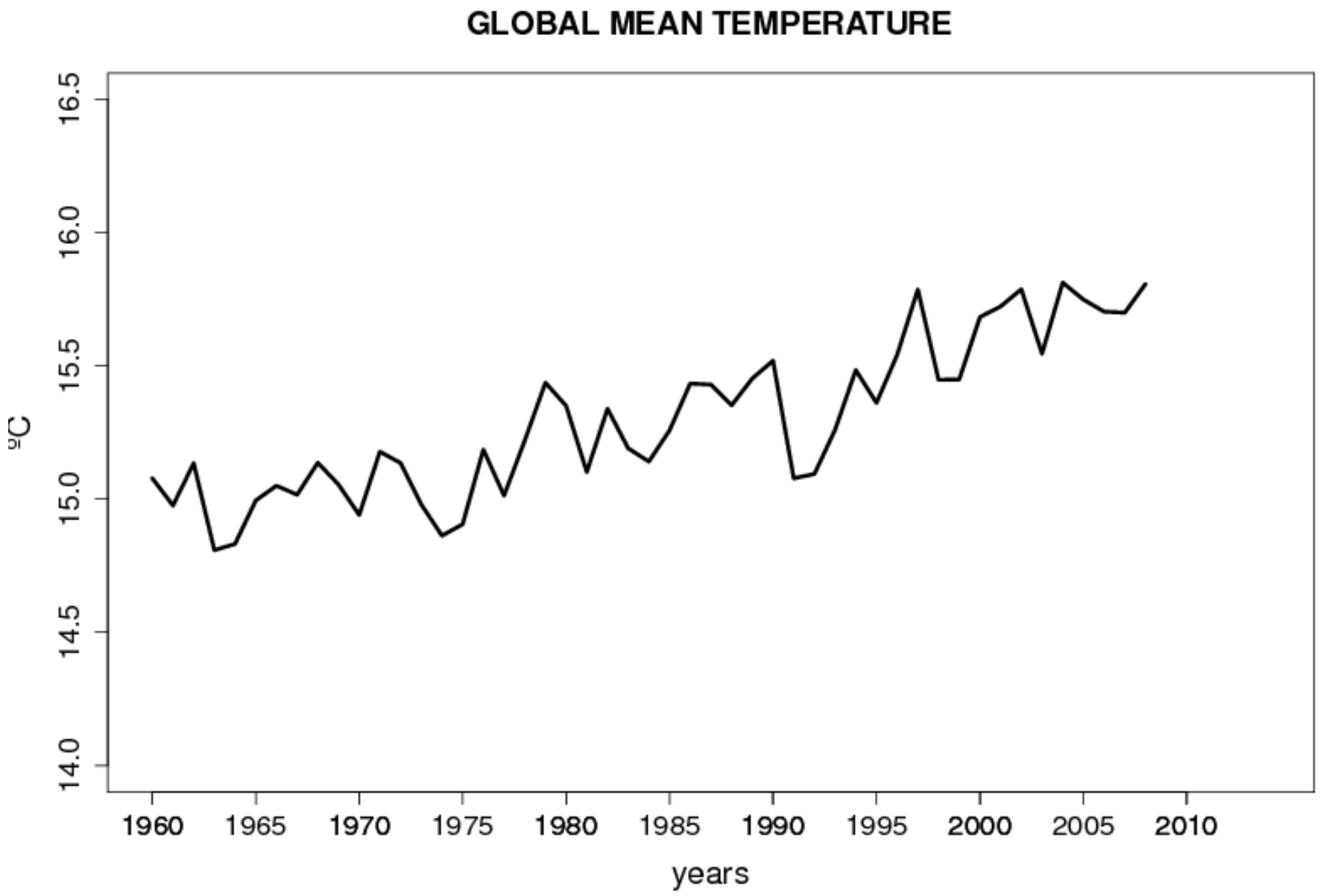
Each job is represented with a colour depending on its status at any particular instance when the plot would be requested.

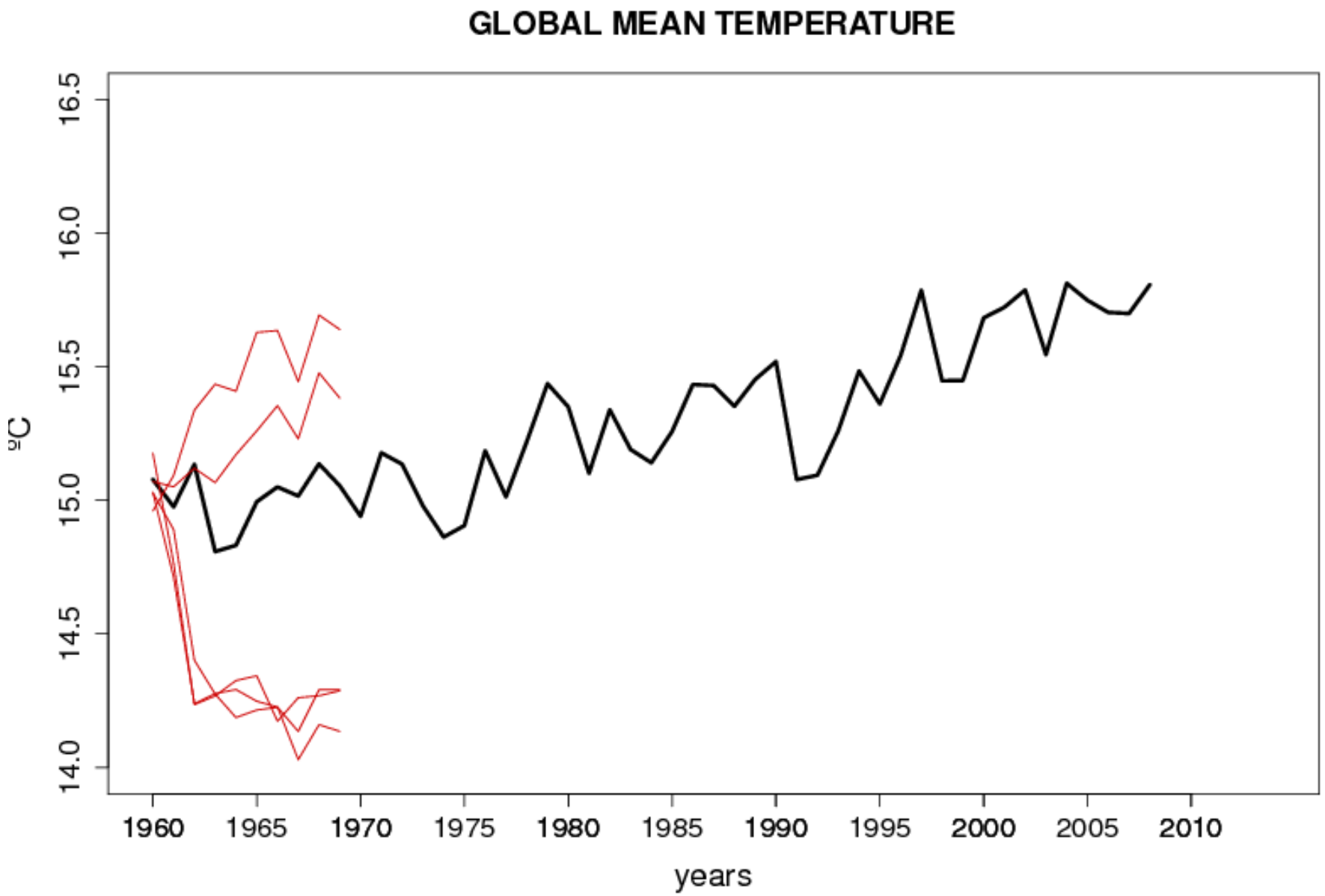
green=completed
orange=running
blue=pending, etc

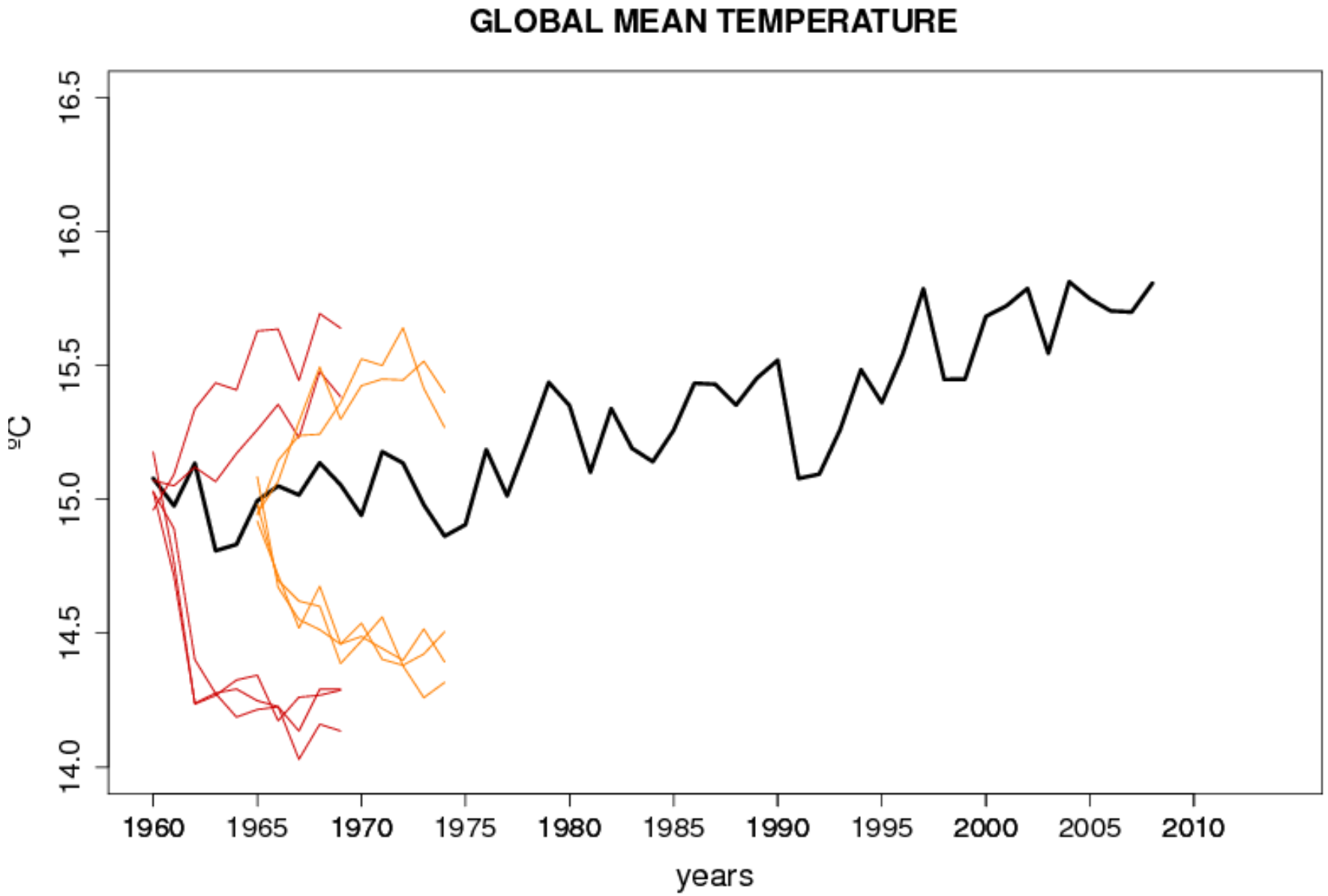
Moreover, with the help of this plot, progress of the experiment can also be viewed or estimated very easily.

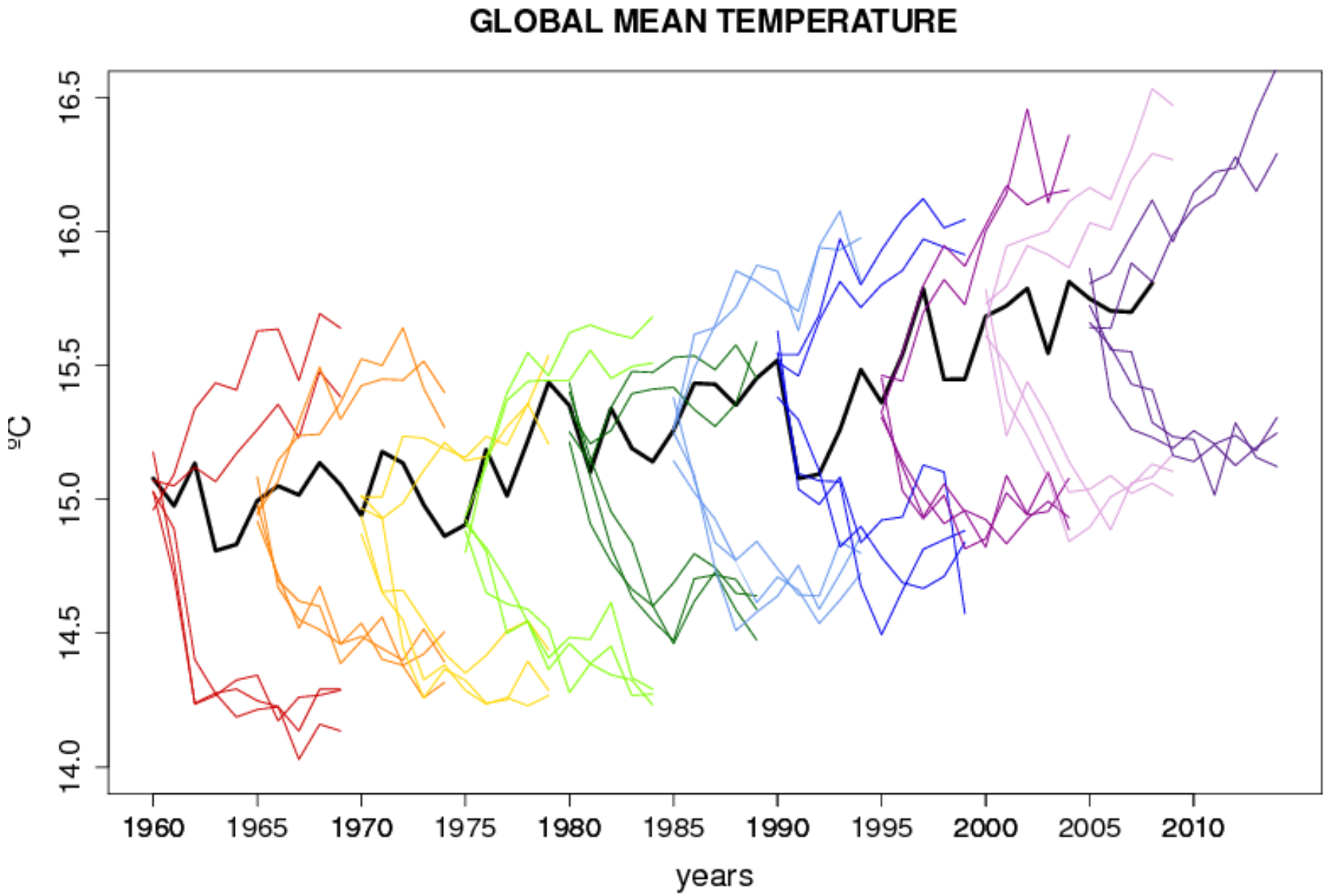
Conclusion: Autosubmit could be used for any typical experiment on any HPC platform to achieve the following goals:

- Efficient handling of highly dependant jobs
- Optimum utilization of available computing resources
- Ease of starting, stopping and live monitoring of experiments
- Auto restarting the experiment or some part of experiment in case of failure
- Ability to reproduce the complete experiment or a part of experiment.


GLOBAL MEAN TEMPERATURE
°C
16.5
16.0
15.5
15.0
14.5
14.0
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010
years


GLOBAL MEAN TEMPERATURE
°C
16.5
16.0
15.5
15.0
14.5
14.0
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010
years


GLOBAL MEAN TEMPERATURE
ºC
16.5
16.0
15.5
15.0
14.5
14.0
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010
years


GLOBAL MEAN TEMPERATURE
ºC
16.5
16.0
15.5
15.0
14.5
14.0
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010
years

- We run EC-Earth (IFS/NEMO) on **45 cores** (IFS: 28, NEMO: 16, OASIS3: 1)
- For a decadal experiment: The model runs one start date and one member for 10 years and produces approx. 650GB data (10 start dates and 5 members approx. 32 TB)
- The table shows the number of cores that we could use simultaneously

No. of Start Dates	No. of Members	No. of Independent Jobs (simulations)	No. of Computing Cores
5	5	25	1125
10	5	50	2250
10	10	100	4500
20	10	200	9000
20	20	400	18000

- We are looking for (and working on): A wrapper over any queue system and HPC scheduler to manage many simultaneous jobs to participate on a Tier-0 machine (where in a Tier-0, one single job should be running with more than 4,000 cores). Specifically:
  - “Python threads”, where each thread corresponds to one autosubmit job: currently testing it but...
  - “Job coupler”, i.e. something like OASIS, but to couple multiple jobs and run as a single executable
  - “MPI wrapper”: we need it to manage and control an arbitrary number of simultaneous jobs
  - “Tier-0 machine account”: to be requested to test all the above