Dealing with high calculation performance and low data storage computing systems.

Akkermans, T., Brisson, E., Van Lipzig, N.P.M.
CLM Community Assembly, WG SUPTECH, 01/09/2010, Berlin
1. Introduction

- Climate models require multiprocessing with high performance computing (HPC) systems.

- These should be composed of both **computational power** and **data storage** capacity.

- However, the two components are not always equally developed, as other disciplines require the latter less than the first.
  - e.g. 112 nodes x 2 quad processors per node = 928 cores, 9.28 Tflops, processors: Intel Xeon L5420 2.5GHz; memory: 8GB/node; **OK!**
  - e.g. personal storage scratch: (only) **1.5TB**

- The resulting lack in data storage capacity is then a limiting factor in climate simulations.
The solution proposed here consists of transferring input and output data while the model is running.

- Only **input data** which is needed by the model in the near future will be copied from local data sources (e.g. external hard disk) to the storage server of the HPC.

- Directly after they are produced, **output data** is copied from the HPC storage server to the local data source.

- In this way, the total storage demand is limited to a minimum.

- A possible disadvantage is the relatively high bandwidth demand.
2. conceptual example

- Running five months: January – May

  **Situation:**
  - All necessary INT2LM output (= CCLM input) stored on local hard disk(s)
  - CCLM model installed on remote HPC
  - Low storage capacity on remote HPC, e.g. only enough for 6 months of data (input and/or output).

- From the moment that the model is started, it is running continuously without any interruptions.

- A script on the local machine controls the **data transfers** and **removal of data** which is not necessary anymore.
  - The script checks every x minutes if the next month has begun, by looking for the presence of a CCLM output file on the remote HPC (lffd2005MM0100.nc, with MM being the next month)
2. conceptual example

INT2LM output = CCLM input

Copy January input
2. conceptual example

INT2LM output = CCLM input

Start model run and producing CCLM output for January

Copy February input

Use input data from January

LOCAL DISK

HPC STORAGE
2. conceptual example

INT2LM output = CCLM input

CCLM output

LOCAL DISK

Use input data from February producing CCLM output for February

copy March input

copy January output

HPC STORAGE

Maximum 4 months of data stored on HPC!

delete input JAN

producing CCLM output for February

JAN  FEB  MAR  APR  MAY

JAN  FEB  MAR
2. conceptual example

INT2LM output = CCLM input

CCLM output

LOCAL DISK

copy April input

HPC STORAGE

Maximum 4 months of data stored on HPC!

copy February output

Use input data from March

producing CCLM output for March

delete input FEB

delete output JAN

Mar

Apr

FEB

JAN

Feb

Mar

Apr

May
2. conceptual example

INT2LM output = CCLM input

- JAN
- FEB
- MAR
- APR
- MAY

Copy March output

Copy May input

Maximum 4 months of data stored on HPC!

Delete input MAR

Delete output FEB

Use input data from April

Producing CCLM output for April
2. conceptual example

INT2LM output = CCLM input

CCLM output

LOCAL DISK

 deletion of APR input

HPC STORAGE

copy April output

producing CCLM output for May

Use input data from May

delete output MAR
2. conceptual example

CCLM output

INT2LM output = CCLM input

copy May output

delete input FEB

delete output APR
2. conceptual example

CCLM output

| JAN | FEB | MAR | APR | MAY |

INT2LM output = CCLM input

| JAN | FEB | MAR | APR | MAY |

LOCAL DISK

HPC STORAGE

delete output MAY
3. conclusion

- The script is located and executed on the **local machine**, because on the remote HPC it is not possible to retrieve data from an external source, e.g. local hard disks (firewall/security limitations).
  - From the local machine one can have a 2-way communication with the HPC cluster.
- Script is available on demand, however, it's not flexible (yet) and should be made more user-friendly.
- In the conceptual example, all the CCLM input (INT2LM output) is already located on local disks. However, in many cases it is not, and a similar script/program should exist to run also INT2LM with low storage needs.
Vielen dank!