

# Data Processing and Archiving of COSMOS experiments

COSMOS meeting 2008  
Heinrich Widmann  
M&D/MPI-M

19. Mai 2008

# Outline

- 1 Status of Millennium experiments
- 2 COSMOS workflow and data processing
  - *Synchronous* data base filling
- 3 Data access
- 4 Summary

## Status of Database Filling

- Four experiments are running (run), post-processed, stored on data server and archived on tapes
- Scripts for *asynchronous* filling ready and in use
  - Due to delays (technical and performance problems) asynchronous DB filling necessary for current experiments
  - For mil0001 and mil0002 first partition almost filled (continue to check ...)
- New workflow with *synchronous* filling implemented and will be used for next experiment, but manageable only by
  - decreased "data pressure",
  - synchronisation between model run and data storage
  - improved load balancing between the data processes

# Millennium experiments

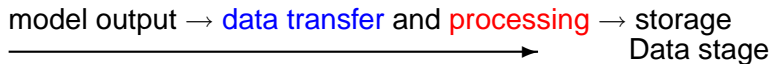
## All COSMOS-ASOB T31L19-GR30L40

ExpId	Description	time integr. (years)	run and on tape	filled in DB
mil0001	Control exp.	800-2800 (2000)	ca. -2600	6h(200y)/ mm(1000y)
mil0002	+ veg. maps (b.g.)	800-1700 (900)	finished	6h(200y)/ mm(1000y)
mil0003	+ veg. maps (max.l.u.)	800-1700 (900)	finished	
mil0004	Control (for en- semble)	1860-2360 (500)	finished	
mil0005	scenario ??	??	-	fill syn- chronous !

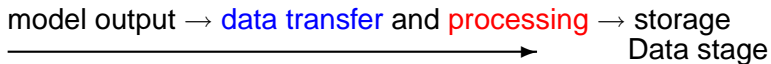
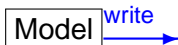
# Data flow and storage

Model

model output → data transfer and processing → storage  
Data stage



# Data flow and storage



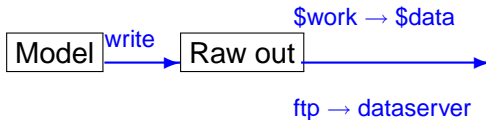
# Data flow and storage



model output → data transfer and processing → storage  
Data stage

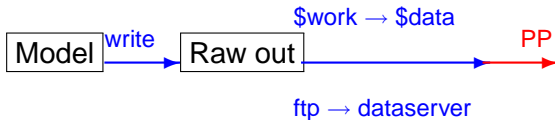


# Data flow and storage



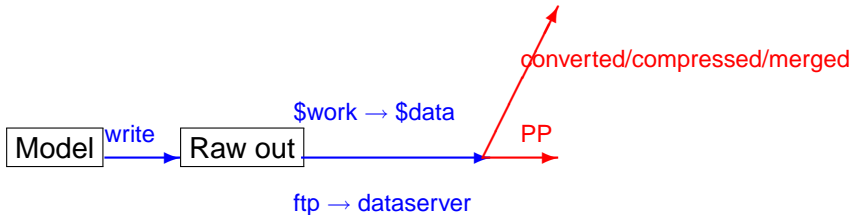
model output → data transfer and processing → storage  
Data stage

# Data flow and storage



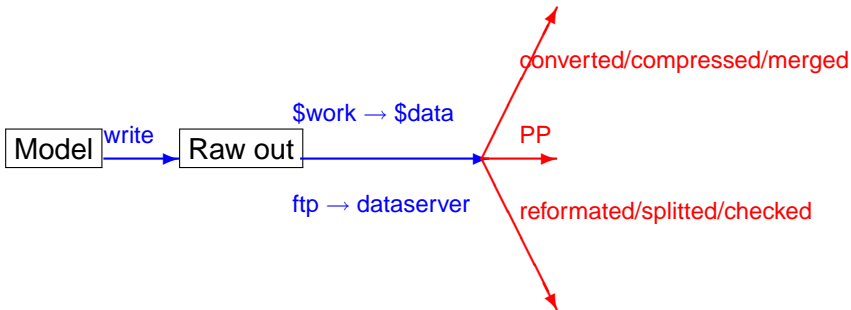
model output → data transfer and processing → storage  
Data stage

# Data flow and storage



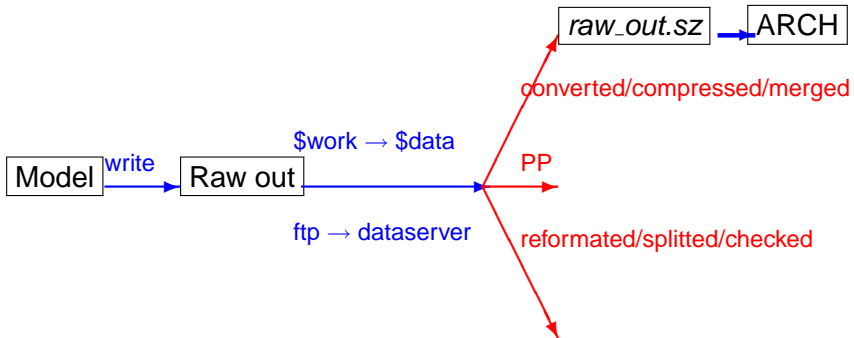
model output → data transfer and processing → storage  
Data stage

# Data flow and storage



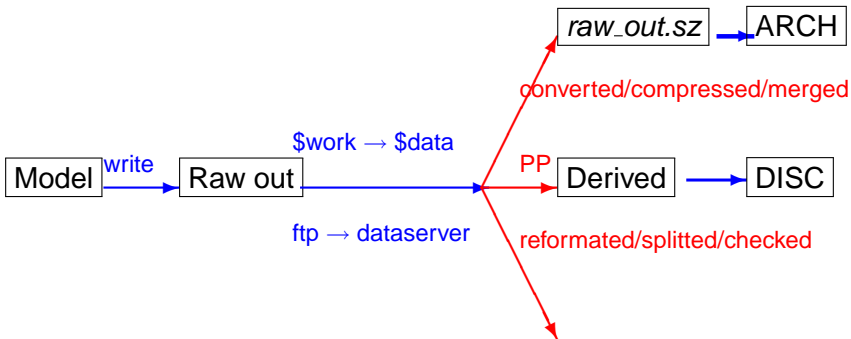
model output → data transfer and processing → storage  
Data stage

# Data flow and storage



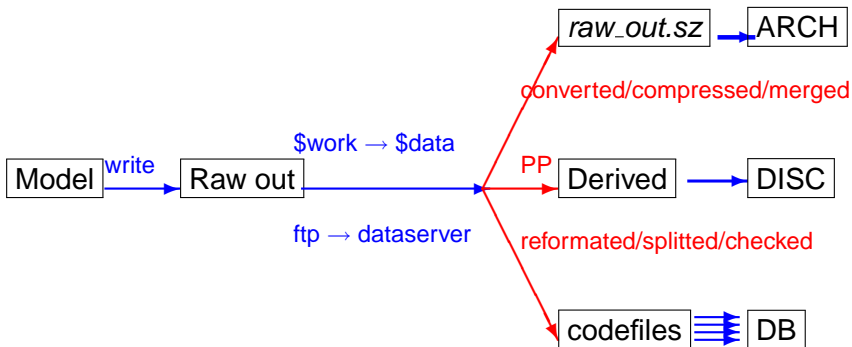
model output → data transfer and processing → storage  
Data stage

# Data flow and storage



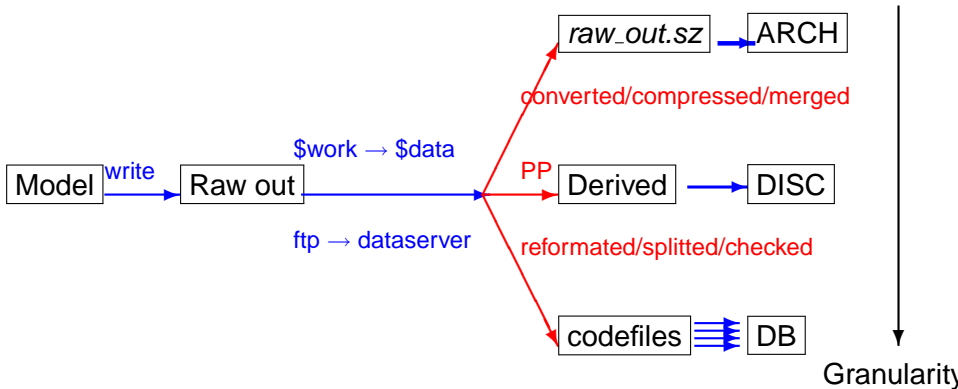
model output → data transfer and processing → storage  
Data stage

# Data flow and storage



model output → data transfer and processing → storage  
Data stage

# Data flow and storage



model output → data transfer and processing → storage  
Data stage



# Synchronous data base filling

*expid.run* (30 min.)



**cross (SX-6 + GFS)**    **mil00**



→ model time



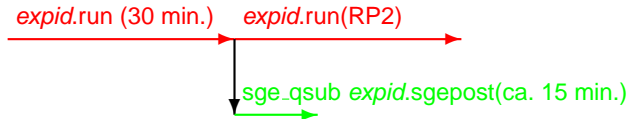
# Synchronous data base filling

*expid.run (30 min.)* → *expid.run(RP2)*

**cross (SX-6 + GFS)**    **mil00**



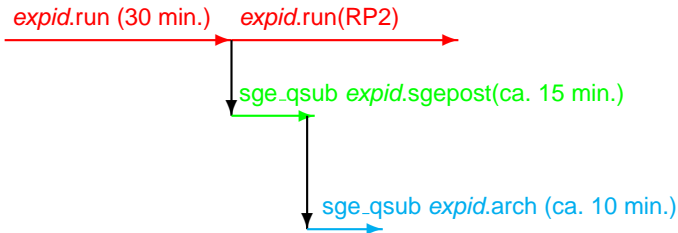
# Synchronous data base filling



**cross (SX-6 + GFS) mil00**



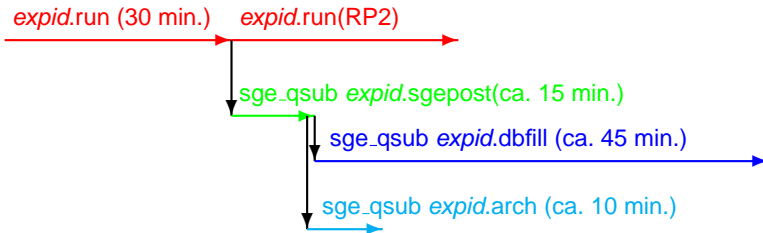
# Synchronous data base filling



**cross (SX-6 + GFS) mil00**



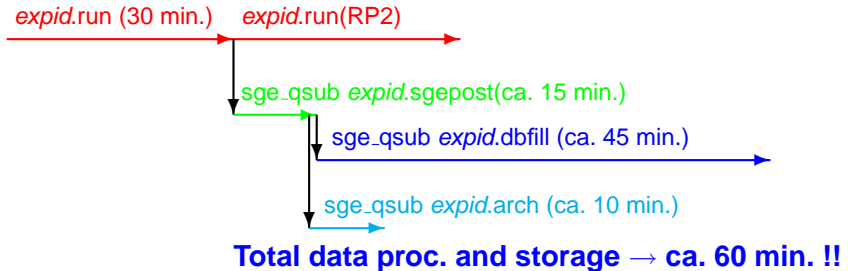
# Synchronous data base filling



**cross (SX-6 + GFS) mil00**



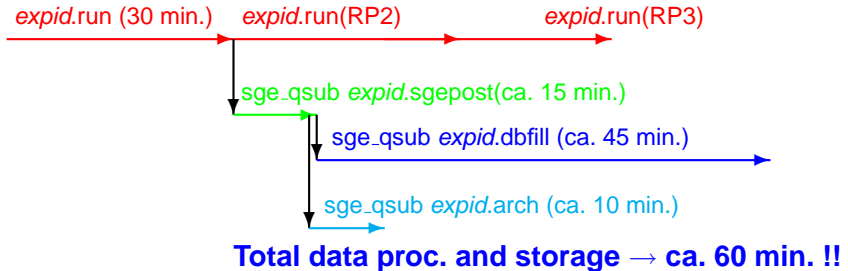
# Synchronous data base filling



**cross (SX-6 + GFS) mil00**



# Synchronous data base filling



**cross (SX-6 + GFS) mil00**



## Turn over time and parallelism of data storage processes

- Data storage (*expid.arch* and *expid.dbfill*) is slower than model integration (*expid.run*) ⇒
  - synchronicity between model run and data storage get lost
  - data jam and bottleneck on data server
- Competing parallel processes on data server ⇒
  - load too high
- Possible solutions
  - runsript (model integration) must "wait" until dbfill is again synchronous or
  - e.g. only part of data can be filled (only monthly and daily means, but not 6 hourly values)

# Data access

Data stored on disk

Archived data

Temporary data

Thumper mil00.zmaw.de

Raw data

*\*.grb* → *\*.sz*

Derived data

*ATM\*, BOT\*, means, ...*

Splitted data

*timeseries (single code)*

DKRZ archive (tapes)

*rawdata.sz*

(merged and compressed)

WDCC data base

*timeseries*

(blobs)



# Data access

DKRZ archive (tapes)

Data stored on disk

Thumper mil00.zmaw.de

Archived data

Temporary data

Raw data

*\*.grb* → *\*.sz*

Derived data

*ATM\*,BOT\*,means,...*

Splitted data

*timeseries (single code)*

*rawdata.sz*

(merged and compressed)

WDCC data base

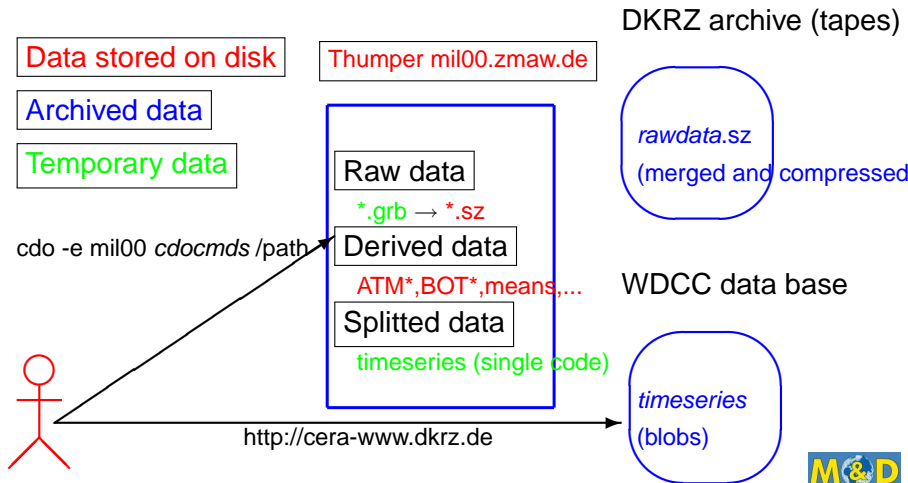
*timeseries*

(blobs)

`cdo -e mil00 cdocmds /path`



# Data access



# CERA access → <http://cera-www.dkrz.de>

The screenshot shows a web browser window displaying the CERA website. The page header includes the M&D logo (Modells & Daten), the text "World Data Center for Climate, Hamburg", and the CERA logo. Below the header, there is a navigation bar with links like "CERA UI Home", "WDC Home", and "Impression". The main content area is titled "Entry information for experiment: mil0002".

**General Information**

Acronym	mil0002
Name	MPI-M Earth System Modeling Framework (mil0002): millennium forced by vegetation maps (best guess)
Type	experiment
Progress	metadata only
Project(s)	<a href="#">MILLIENNIUM project aims at ensemble simulation of the last millennium (800-2000 AD) using the comprehensive COSMOS Earth System Model (MILIENNIUM)</a>
Summary	The involved models are abbreviated and are based on model grids as follows: ECHAM5 (EHS): T31L19 MIPCOM ISIRACH (ISR) HAMMOCC Output from the model run: climate simulation, COSMOS
Keyword(s)	climate simulation, COSMOS
Creation date	09/04/2008

**Quality**

Accuracy report	model data
Completeness report	not filed
Consistency report	as consistent as the model are
Historical accuracy	not filed

At the bottom of the page, there is a search bar with the text "Find: ALLCACC" and several checkboxes for search options: "next", "previous", "Highlight all", and "Match case".

# Open

- Load balancing of data processing in *synchronous* workflow !
- CERA access policies for Millennium ?
- T63L19-G15L40 runs ?
- Fill from compute node (IBM or tornado cluster) ?
- Update documentation and define responsibilities !