# *Attempt at an overview of the CASA* “HPC” features mentioning the ALMA pipeline and the ALMA QA2 script generator

D. Petry (ESO)

# Motivation

**- ALMA datasets are large**
**typical sizes of a single dataset with high spectral res. will be 500 GB**
**(MS format, 0.5 h obs time = 1 SB, 40 antennas)**

**- CASA (e.g. v3.4, official ALMA version in Cycle 0) is too slow:**
**already processing a ca. 50 GB dataset took ca. 2.5 h on a 3 GHz machine**
**=> processing one typical SB execution from cycle 1 will take one day at this rate**

**- Study shows not all processing tasks are I/O limited, especially not when a fast filesystem is available (e.g. SSD or Lustre)**
**=> can gain from higher computing speed, e.g. by parallelisation**

**- Up to CASA 3.4, the individual CASA instance runs on a single processor except where OpenMP is used (regridding in imaging, table sorting).**

**- HPC effort started after CASA 3.4 release. Goal: accelerate CASA by parallelisation**

# CASA parallelization

**- non-trivial parallelization**

**→ already implemented in some parts of the imaging code and the table module**

**→ achieved on the C++ level using OpenMP**

**→ completely transparent to the user**

**→ not applicable to most of the processing tasks**

**→ time-consuming to develop and debug**

**- trivial parallelization**

**→ idea: split the input data into independent chunks, process them in parallel**

**→ design by J. Kern:**

**- task *partition* splits the data along the time and frequency axes into a *Multi-MS (MMS)* - *looks like a single MS but is many sub-MSs underneath***

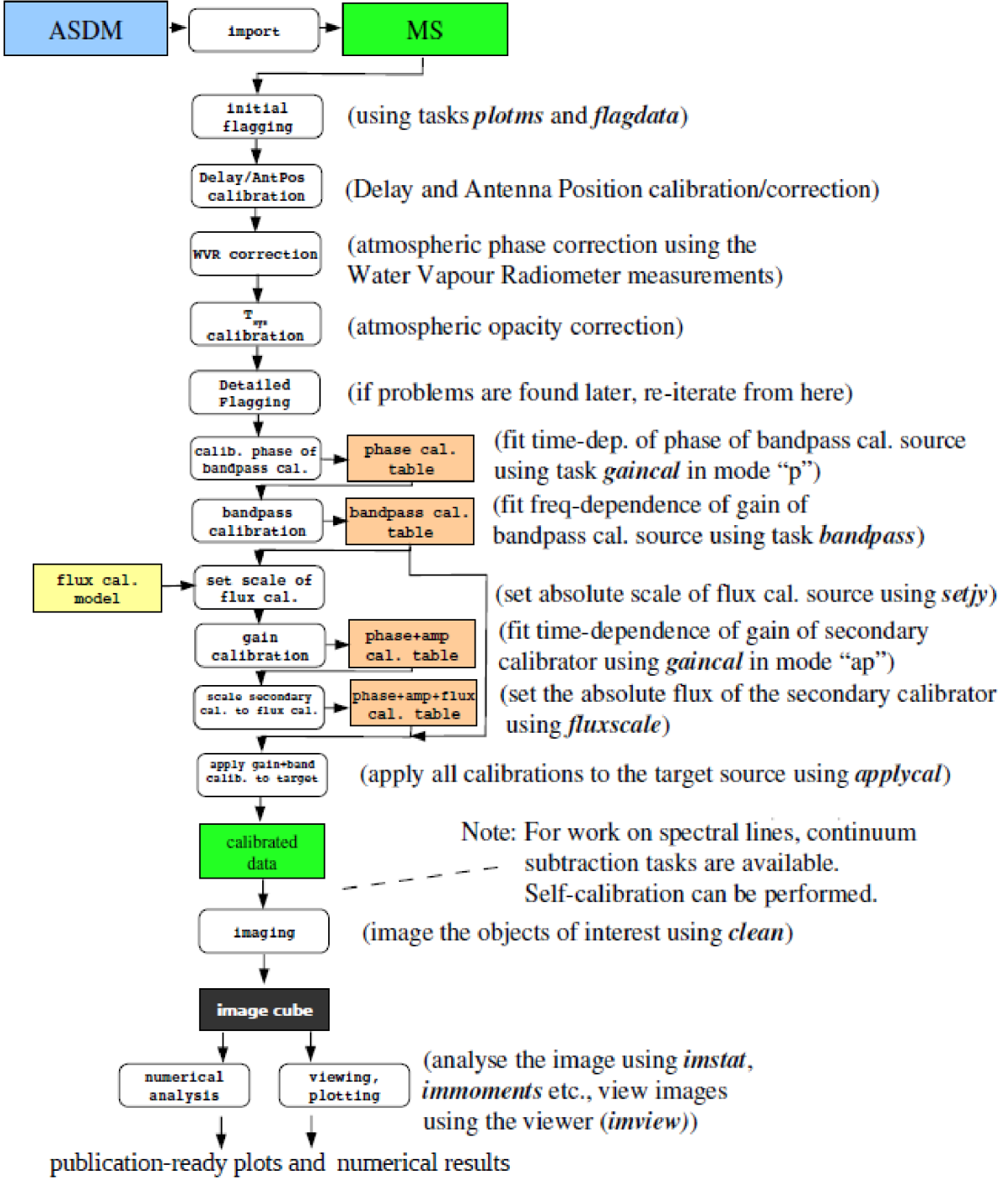
**- all subsequent tasks recognize the MMS and *parallelize themselves* to process the sub-MSs accordingly.**

**→ mostly transparent to the user**

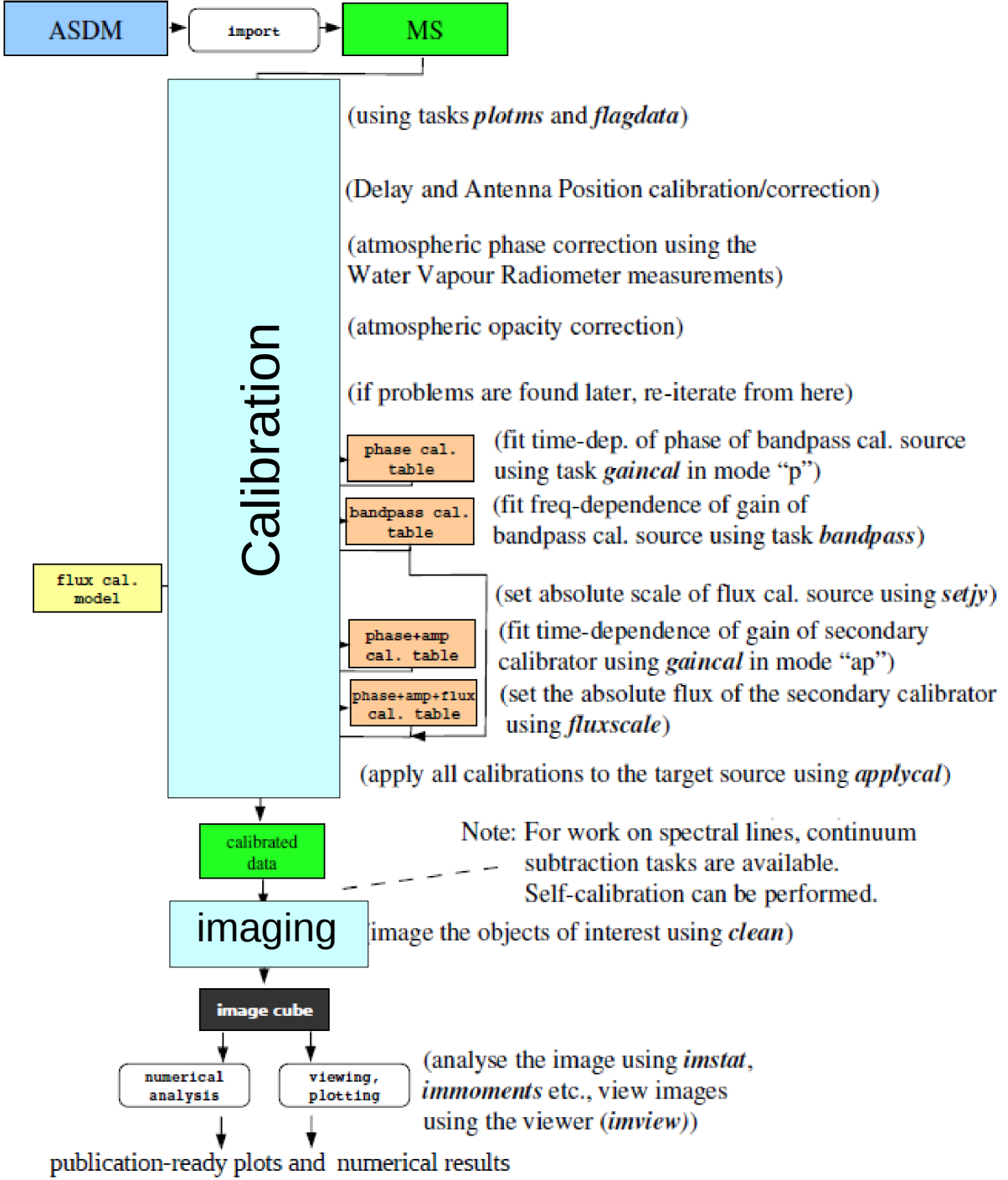
**→ applicable to many of the processing tasks, more simple to test and debug**

**→ proven overall speed-up by factors > 2 depending on the file-system used**

# CASA parallelization



# CASA parallelization



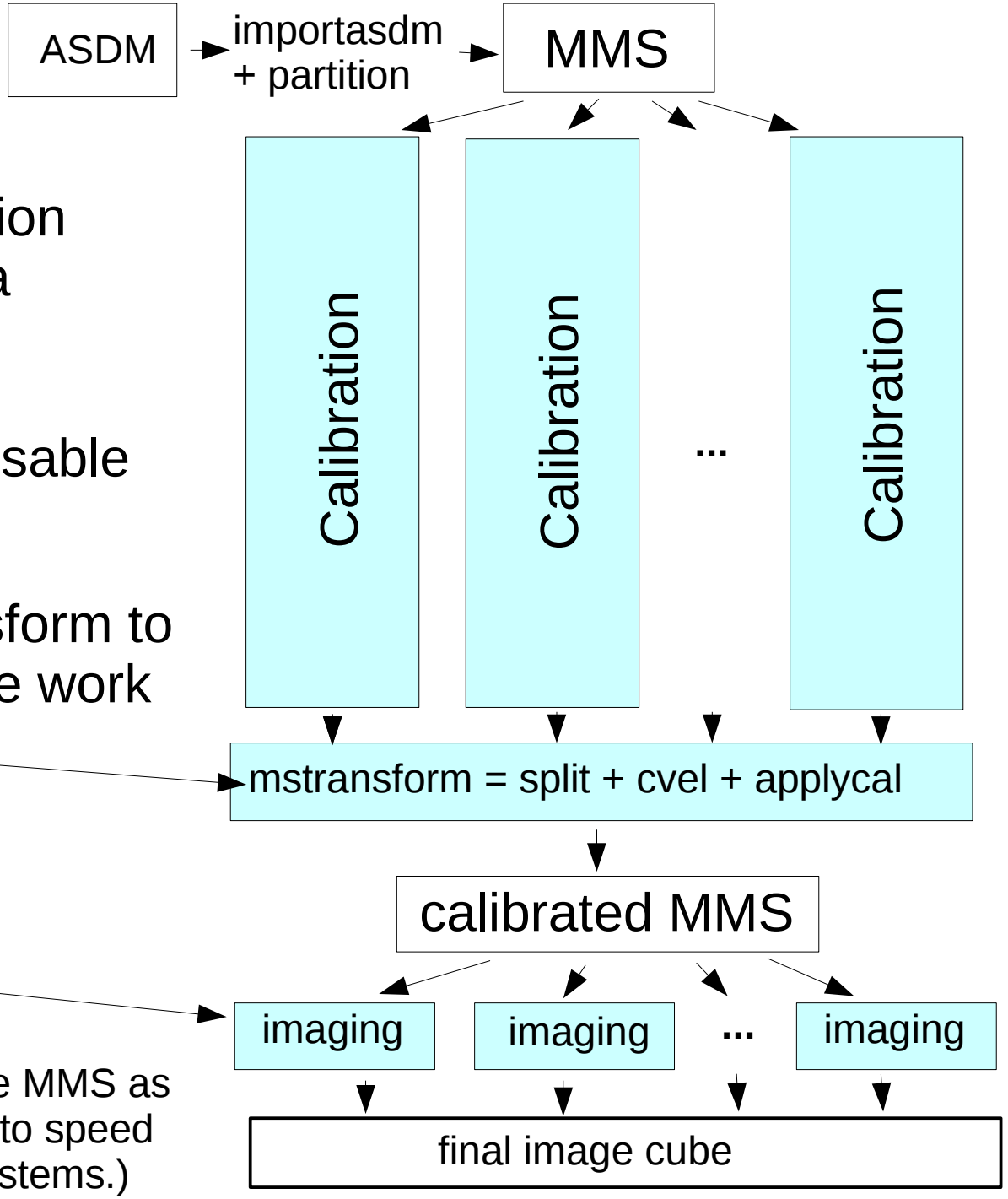
CASA parallelization
- the general idea

Partition to create separately processable chunks of data

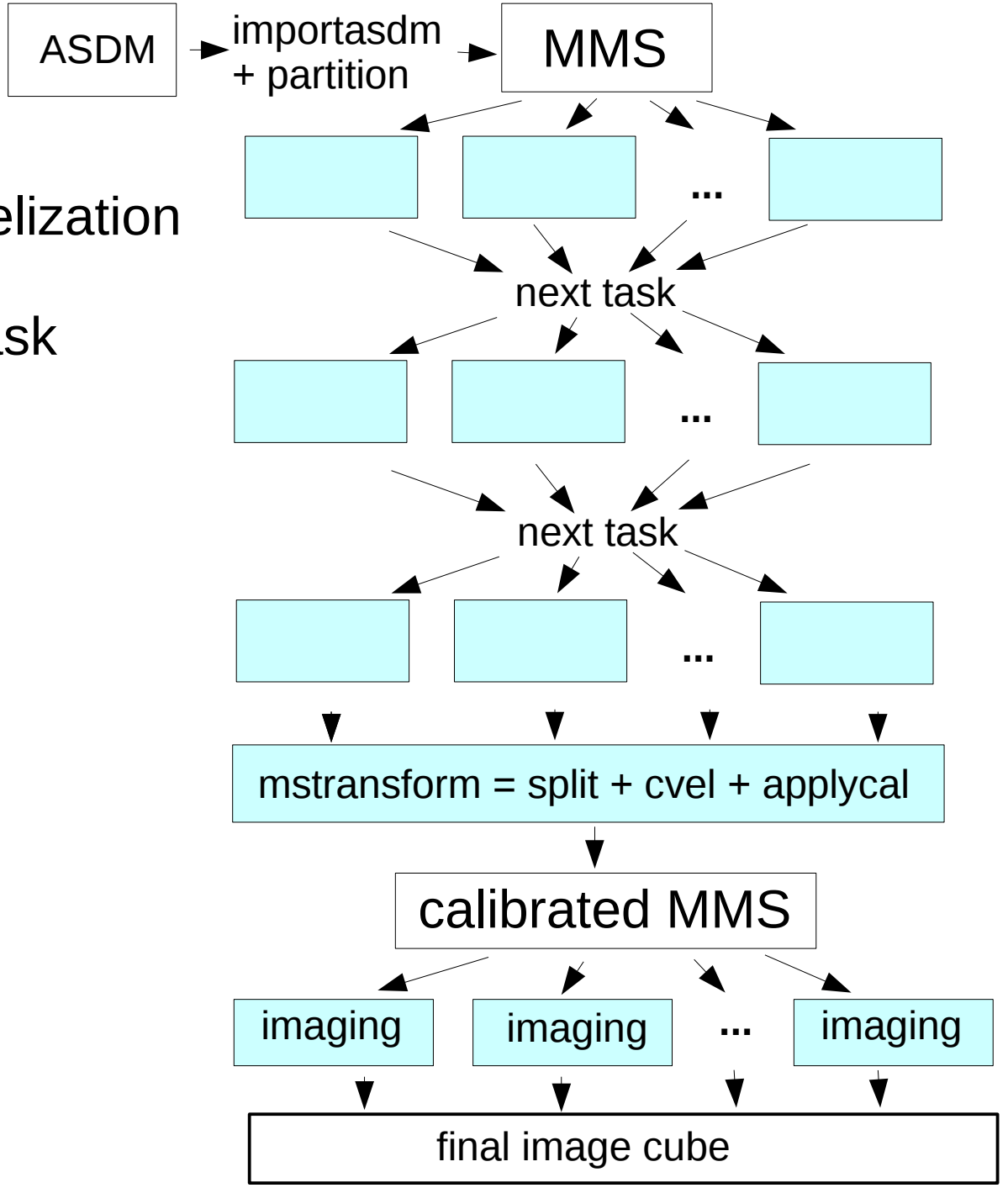
New task mstransform to do all I/O-intensive work at once

Parallel imaging

(pclean does not require MMS as input but it is supposed to speed up I/O on parallel file systems.)



CASA trivial parallelization
in v4.0 and v4.1
happens task by task



# “HPC” work at ESO for 4.0 dev. cycle

**- Work in the CASA 4.0 development cycle (mostly at ESO, May – October 2012)**

**- results:**

**1: created monitoring and assessment *utilities* for the parallelization framework and the MMSs**

**2: completed the *partition task* and the *parallelization framework* (created proper unit tests, debugged the existing prototypes, documented)**

**3: went through nearly all *parallelizable tasks* and made them work in the parallization framework (extended unit tests accordingly, documented), imaging parallelization to be done by K. Golap in Socorro**

**4: created an *end-to-end regression tests* which uses the new framework and measured speed-up**

***See new chapter in the CASA 4.0 cookbook.***

# “HPC” work at ESO for 4.0 dev. cycle

**Results of the 4.0 dev. cycle**

**- speedup measurements based on ca. 30 GB ALMA datasets were *disappointing (only ca. 10-20% speedup overall)***

**- presently assume our dataset too small to demonstrate the speedup, but even on a SSD system the speedup is small.**

**- Discussion on parallel file systems vs. solid state disks necessary**

**- Advantages of a parallel file systems *for the individual user running one analysis session still have to be demonstrated.***

# “HPC” work at ESO for 4.0 dev. cycle

**- Work in the CASA 4.1 development cycle (since end of 2013)**

**1) Creation of the *“mstransform” task (mostly at ESO)*,**
**a combination of split, cvel, *applycal, and partition***
**functionality (recurring pattern in analysis scripts)**
**to achieve further speed-up by avoiding I/O on intermediate data**
**(at least partially in release 4.1)**

**2) Ger, Michel, and DP developed “lazy” import in importasdm:**
**new asdmStMan storage manager enables direct access to the ASDM**
**via the DATA column (which is read-only anyway!)**
**=> reduces data volume on disk by nearly 50%,**
**speeds up import by ca. 80%,**
**speeds up access (due to compressed storage in ASDM) by 20%,**
**=> overall speedup equivalent to elimination of import(!)**

**3) DP developed “virtualconcat”: concatenation of MSs into MMS**
**speedup factor 5 to 10 w.r.t. concat**

# “HPC” work at ESO for 4.0 dev. cycle

**...**

**2) Ger, Michel, and DP developed “lazy” import in importasdm:**
**new asdmStMan storage manager enables direct access to the ASDM via the DATA column (which is read-only anyway!)**
**=> reduces data volume on disk by nearly 50%,**
**speeds up import by ca. 80%,**
**speeds up access (due to compressed storage in ASDM) by 20%,**
**=> overall speedup equivalent to elimination of import(!)**

**3) DP developed “virtualconcat”: concatenation of MSs into MMS**
**speedup factor 5 to 10 w.r.t. concat**

**The last two points show:**
***Speed can not only be gained by brute-force parallelisation!***
***Deep understanding of the data structure and clever optimisation of time-consuming processes can give large improvements also without using more than one processor!***

# The ALMA Pipeline

**- No time to go into the decade-long history of the ALMA pipeline**

**- Status now:**
- **calibration section nearing completion for cycle 0 features**
- **imaging section still rudimentary**

**- Ideas:**
- **Separated Heuristics and Context routines**
- **Analysis in “stages”**
- **User submits “Pipeline Processing Request” (PPR), pipeline does the rest and takes all decisions automatically including reiterations etc.**
- **To be embedded in CASA and distributed with it, anybody can rerun it**

**- Expect to switch ALMA QA2 calibration to pipeline late this year**

# The ALMA QA2 script generator

**- For commissioning and quality assurance, JAO and the ARCs developed the *QA2 Script Generator* (lead: E. Villard)**

**- general idea: given an raw ALMA MS, the generator creates analysis scripts which do the entire data analysis**

**- separate script for calibration and imaging**

**- scripts meant to be run by the QA2 expert analysts AND the science end user**

**- initially the expert analysis runs the scripts step by step (special stepping mechanism implemented)**
**and makes corrections if necessary (e.g. additional flagging)**

**- when the script is OK, it can be run in one go or in groups of steps**

**- final scripts are part of the delivery to the science end user**