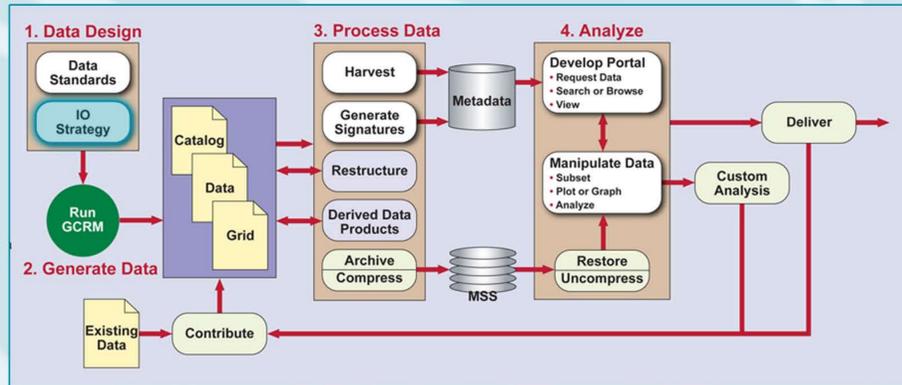


# IO Strategies for a Global Cloud Resolving Model

BJ Palmer, AS Koontz, KL Schuchardt • Pacific Northwest National Laboratory

<http://climate.pnl.gov> • [bruce.palmer@pnl.gov](mailto:bruce.palmer@pnl.gov)

The Global Cloud Resolving model, with spatial resolution of approximately 2-4 km over the entire globe and on the order of 100 vertical layers, is paradigm changing. With data output rates of approximately 1 TB per simulated hour, all aspects of data design, generation, processing, and analysis must be examined. Key components of this effort are shown in the figure. In this poster, we focus on our work in evaluating IO strategies for efficient data output.



## IO Strategies

### Objectives

- Write files in standard, platform-independent format that supports data analysis.
- Maximize bandwidth from compute servers to file system.
- Minimize blocking of main computation by IO.
- Minimize memory requirements for IO.

### Configuration I

- Easiest to implement; current IO model for many codes.
- Not expected to scale to very large processor counts; bandwidth contention for IO may slow down computation.
- IO blocks computation.

### Configuration II

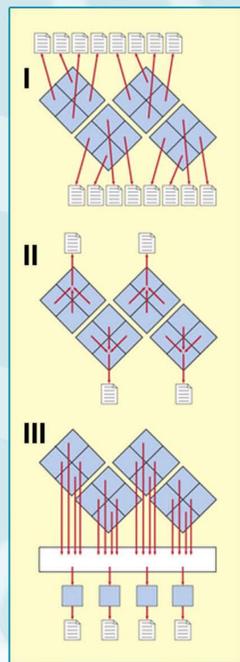
- Collect data on a subset of processors for IO.
- Optimize bandwidth by using more than one processor for IO but avoid saturating the system by not using all processors.
- Will still block processors from doing computation until IO is complete.
- Uses 'faster' communications bandwidth to reduce IO contention.

### Configuration III

- Split off a subset of processors dedicated exclusively to performing IO.
- Size of the IO group can be selected to optimize IO bandwidth and configuration can be programmed so that IO does not block main computation.
- May require significant amounts of extra memory to implement.

### Variations

- Avoid the need for post-processing by structuring data so that each variable is in its own file.
  - reduces amount of data needed from off-line storage for typical time-series analysis
  - grid needs to be in separate file to avoid excessive duplication
- Use Parallel NetCDF (and MPI\_IO) rather than moving data over communications channel.
  - hides detail of organizing IO access across processors
  - may result in more segmented writes
- Experiment with file OS striping options.

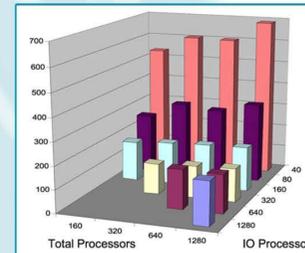


## Approach

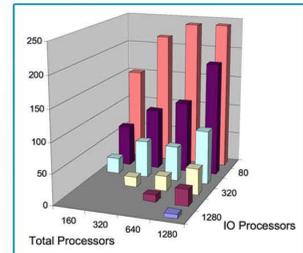
IO routines from BUGS5 superparameterization code were appropriated for these tests to preserve IO format. Data is output in multiple files representing a square portion of the geodesic grid. Unless otherwise noted, the results reflect a combination of Configurations I and II with spatially divided data sets and using Global Arrays (GA) toolkit to manage internal data layout and interprocessor communications. GA provides a higher level API that substantially simplifies parallel code development. Initial tests indicate that GA performs as well as MPI in this application.

## Preliminary Results

Average Time per Processor Spent on IO (MPP2)

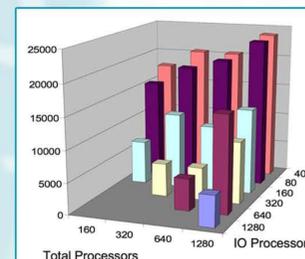


Average Time per Processor Spent on IO (Jaguar)

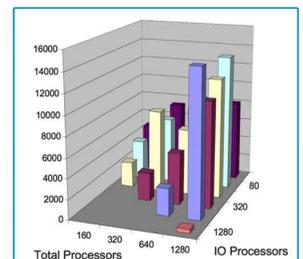


Both MPP2 and Jaguar show scaling up to large numbers of processors. Jaguar exhibits scaling all the way up to 1280 processors for R=10 test case. MPP2 appears to show optimum IO performance at about 320 IO processors, after that performance degrades slightly.

Aggregate Time Spent in Communication (MPP2)

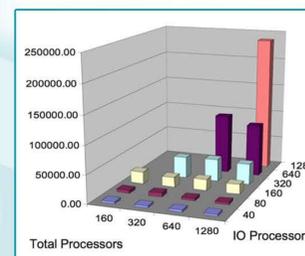


Aggregate Time Spent in Communication (Jaguar)

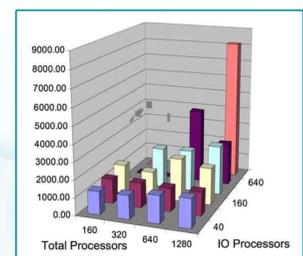


Both MPP2 and Jaguar show scaling behavior with respect to number of IO processors but not with respect to total number of processors. Aggregate time spent in communication increases with number of processors, possibly due to contention on the switch.

Aggregate Time Spent in Writes to Disk (MPP2)

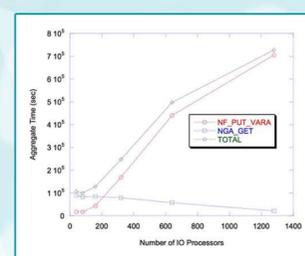


Aggregate Time Spent in Writes to Disk (Jaguar)

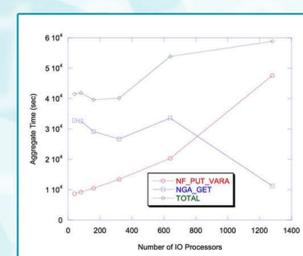


Both platforms show decreasing bandwidth as number of IO processors increases. Behavior is relatively flat with respect to total number of processors.

IO Benchmarks for 1280 Processors on MPP2

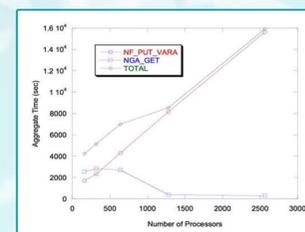


IO Benchmarks for 1280 Processors on Jaguar

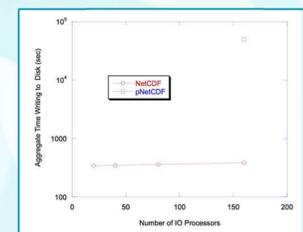


Communication (NGA\_GET) scales on both platforms with respect to number of IO processors, writes to disk (NF\_PUT\_VARA) do not. At low IO processor counts communication dominates but at high IO processor counts, writes are dominant. This is much more pronounced on MPP2 than on Jaguar.

All Processors Perform IO (Jaguar)



Comparison of NetCDF and Parallel NetCDF



Except at small processor counts, writes (NF\_PUT\_VARA) dominate communication (NGA\_GET). Decreasing bandwidth in writes results in non-scaling behavior.

The bandwidth achievable with NetCDF in this test appears to be orders of magnitude faster than for Parallel NetCDF. This may be due in part to the fact the NetCDF test is probably writing much larger contiguous chunks of data than the parallel NetCDF test. Further investigations are ongoing.

## Conclusion

Preliminary results indicate that communication scales well to high IO processor counts but writes to disk do not, suggesting that for individual processors writing to separate files, there is a point of diminishing returns at which it is no longer productive to have more processors engaged in IO. Communication associated with moving data to the IO processors is a significant cost with both MPI and GA communication libraries. Initial tests of the Parallel NetCDF library with multiple processors writing concurrently to the same file suggest that this results in a significant drop in effective IO bandwidth, but these results are very preliminary.