File Header and Variable File Layout Alignment

From the beginning, PnetCDF has taken an MPI INFO object in the file create and open routines. This object has to this point been used to pass hints down to the MPI-IO layer, but nothing precluded using those hint objects at the PnetCDF layer. We added a new feature in 1.1.0 that uses these MPI info hints to align the space allocated for file header and the starting file offsets of the non-record variables.

Two PnetCDF hints that can be used to control the alignment are "nc_header_align_size" and "nc_var_align_size". The former aligns the header size of a newly created file. The latter, "nc_var_align_size", is used to align the starting file offsets of all non-record variables. (The alignment for record variables is currently in our todo list.)

Usage examples

```
MPI_Info_set(info, "nc_header_align_size", "1048567");
MPI_Info_set(info, "nc_var_align_size",    "4194304");
ncmpi_create(MPI_COMM_WORLD, "filename.nc", mode, info, &ncid);
```

Default values

The performance of PnetCDF can be significantly affected by the alignment, especially when the aggregate I/O request amounts are large. On parallel file systems, setting the hints equal to the file system striping size often results in the best performance. Default values of the two alignment hints are calculated below. If the MPI-IO hint "striping_unit" is larger than zero (set by the user or obtained from the file system) and the total size of all non-record/record variables is lager than N * striping_unit, then the default for both hints is set to striping_unit. (Currently, N=4.) Otherwise, the default is 512 bytes.

The alignment can be disabled by setting both hints to "1".

Why use "nc_header_align_size" hint?

This hint allows some extra space between the end of the header describing the entire file and the first variable. If you have an application that periodically wishes to add more variables to an already existing file, expanding the file header size may result in an expensive move of the entire data file to make room for the definition of the new variables. Hence, setting this hint to a value that is big enough to accommodate any additional variables means you may leave your application code as-is and yet still see tremendous performance improvements.

Why use "nc_var_align_size" hint?

If you are writing to a block-based parallel file system, such as IBM's GPFS or Lustre, then an application write becomes a block write at the file system layer. If a write straddles two blocks, then locks must be acquired for both blocks. Aligning the start of a variable to a block boundary, combined with collective I/O optimizations in the MPI-IO library can often eliminate all unaligned file system accesses.

Why two separate hints?

Note that the two alignment hints can be set at the same time, with different values. When this happens, the starting file offset of the first variable will be set to a number that is a multiple of both hint values. Setting different values

may be useful in two scenarios: 1) when a file has a large header and small-sized non-record variables, and 2) a small header and large-sized non-record variables.

Example scenarios

Alignment for file system: Argonne's BlueGene system has a GPFS file system with a 4MB block size. By setting the "nc_var_align_size" hint to 4MB, PnetCDF will round up the starting offset of each non-record variable to the next 4MB.

Padding header for future growth: An application creates a checkpoint file, but has structured the code so that each component writes its own information to the checkpoint file. Since we don't know 100% of the information at the initial define mode time, this application has to call `ncmpi_redef()` for each component. If this call results in a bigger header on disk, then PnetCDF initiates a very expensive data movement step. However, the application could make use of the header alignment hint to ensure there is enough room for the header to grow and accommodate additional variables.

Limitations

This hint will have no impact on the alignment of record variables. I (robl) tried but could not get the correctness tests to pass. Patches welcome!