

HPC-HDF5 - Feature # 598: Enable starting 'core' VFD from existing buffer,

| | | | |
|-----------------------|---|---------------------|------------|
| Status: | New | Priority: | Normal |
| Author: | Mark Miller | Category: | |
| Created: | 02/10/2011 | Assigned to: | |
| Updated: | 02/10/2011 | Due date: | 03/09/2011 |
| Impact: | 3 - Medium | | |
| Expected Use: | 3 - Occasional | | |
| OS: | All | | |
| Support Group: | Any | | |
| Subject: | Enable starting 'core' VFD from existing buffer, | | |
| Description: | <p>Steve Langer pointed out that this request regarding something like what Vamsee referred to as 'htpurl vfd' (a slight swizzle on core vfd), would also be good for a particular parallel I/O setting; read WHOLE file on one processor and broadcast to all others. Proc 0 would use core vfd to 'read' it, get the pointer to the 'core' file, and broadcast that whole chunk of bytes. All other procs would just sit in broadcast, and upon getting the bytes, open the 'core' driver on those bytes.</p> <p>So, I am thinking there would be a good 'new' vfd to define for parallel HDF5 too for this. Of course, it would be for read-only setting I think. Maybe read-write could be made to work but that is not a requirement/need and if it was ever attempted, I think it would be ok to limit to...</p> <ul style="list-style-type: none"># overwrites of data already existing in file# a small number of writers and writes <p>For parallel HDF5, I am not sure what you'd call this beast of a vfd but I am certain it would be very useful to parallel codes using HDF5's parallel interface in this way.</p> <p>Of course, you could stay away from parallel HDF5 (and the MPI part of it) and NOT have HDF5 able to do all that 'magic'. Then, application is responsible for acquiring buffer on proc 0, broadcasting and then calling (serial) HDF5 on each proc to 'open' that buffer of bytes. In fact, that might be best just to leave it, and then just add a parallel use case test for this mode as well as a tiny HDF5 tutorial on how to do it.</p> <p>Of course, you could do BOTH; define the 'open these bytes in core' vfd for serial HDF5 and define the special broadcasting variant of it for parallel HDF5.</p> <p>Here are the basic requirements from our end.</p> <ul style="list-style-type: none"># the ability to 'open' an HDF5 'file' existing as a buffer of bytes already in memory.# the ability to obtain a pointer to a buffer of bytes in memory after having 'opened' an HDF5 file with the core vfd.# functionality cannot be 'bound' to parallel HDF5 only. That is, 1) & 2) should be possible with a serially compiled HDF5. <p>The other options suggested here (making a convenient parallel vfd, supporting limited writes) are just some nice icing on that cake.</p> | | |

History