



Brandt Belson <belson17@gmail.com>

Some questions concerning modred

6 messages

Tobias Leibner <l_tobi01@uni-muenster.de>
To: belson17@gmail.com

Fri, Jun 3, 2016 at 3:09 AM

Dear Brandt Belson,

I am trying to use modred to perform a MPI-parallel POD on a cluster. Using VecHandleTextArray or VecHandlePickle both for the snapshots and the modes works as intended, but I am not quite sure how to use VecHandleInMemory.

If I am using VecHandleInMemory for the snapshot vectors, does every MPI rank have to pass all the snapshots handles in the PODHandles.compute_decomp(...) method? That is, do I have to copy all the snapshots to all processes or is it enough that every snapshot is passed to the compute_decomp function in some process?

And using VecHandleInMemory for the modes, it seems like the modes are scattered across the processes and not collected on rank 0, so I will have to collect them manually after the POD?

Kind regards,

Tobias Leibner

Brandt Belson <belson17@gmail.com>
To: "Tu, Jonathan H CIV NSWCCD, BANGOR, 7102" <Jonathan.Tu@navy.mil>

Fri, Jun 3, 2016 at 9:59 AM

Hey Jon, Is this a good answer to his email?

Hello Tobias,
I'm glad you're using modred, and I hope you find it useful. I'm also CC'ing Jon Tu, co-author.

VecHandleInMemory has the same interface as the other VecHandles you mentioned. You should write your code to use modred as if there is only one processor. When run in parallel, each processor will have identical copies of all the VecHandleInMemory instances. To use VecHandleInMemory in a parallel setting, it is required for all processors to have all of the VecHandleInMemory instances in memory. If you have memory limitations, then it is best to use one of the other VecHandle classes that are designed to work with larger datasets. As a general note, if all of your data fits in memory, then it might be the case that parallelization is not needed. You could also consider the "compute_POD_matrices..." methods.

When you call PODHandles.compute_decomp(), you should do so on all processors and each processor should have all of the VecHandleInMemory instances.

The resulting modes are not scattered across the processors when they are returned. All of the modes are returned to every processor. You will not need to collect them manually after POD. In general, you should need to do nothing with MPI yourself other than call the script with "mpiexec" or similar.

I hope that helps. Let us know if you have any questions.

Best,
Brandt

[Quoted text hidden]

Tu, Jonathan H CIV NSWCCD, BANGOR, 7102 <Jonathan.Tu@navy.mil>
To: Brandt Belson <belson17@gmail.com>

Fri, Jun 3, 2016 at 10:17 AM

I think that makes sense, yeah.

Jonathan Tu
[Quoted text hidden]

 **smime.p7s**
6K

Brandt Belson <belson17@gmail.com>
To: Tobias Leibner <l_tobi01@uni-muenster.de>
Cc: "Tu, Jonathan H CIV NSWCCD, BANGOR, 7102" <Jonathan.Tu@navy.mil>

Fri, Jun 3, 2016 at 11:47 AM

Hello Tobias,
I'm glad you're using modred, and I hope you find it useful. I'm also CC'ing Jon Tu, co-author.

VecHandleInMemory has the same interface as the other VecHandles you mentioned. You should write your code to use modred as if there is only one processor. When run in parallel, each processor will have identical copies of all the VecHandleInMemory instances. To use VecHandleInMemory in a parallel setting, it is required for all processors to have all of the VecHandleInMemory instances in memory. If you have memory limitations, then it is best to use one of the other VecHandle classes that are designed to work with larger datasets. As a general note, if all of your data fits in memory, then it might be the case that parallelization is not needed. You could also consider the "compute_POD_matrices..." methods.

When you call PODHandles.compute_decomp(), you should do so on all processors and each processor should have all of the VecHandleInMemory instances.

The resulting modes are not scattered across the processors when they are returned. All of the modes are returned to every processor. You will not need to collect them manually after POD. In general, you should need to do nothing with MPI yourself other than call the script with "mpiexec" or similar.

I hope that helps. Let us know if you have any questions.

Best,
Brandt

Brandt Belson, Ph.D.
Data Scientist
[MOCAP Analytics](#)

On Fri, Jun 3, 2016 at 3:09 AM, Tobias Leibner <l_tobi01@uni-muenster.de> wrote:
[Quoted text hidden]

Tobias Leibner <l_tobi01@uni-muenster.de>
To: Brandt Belson <belson17@gmail.com>
Cc: "Tu, Jonathan H CIV NSWCCD, BANGOR, 7102" <Jonathan.Tu@navy.mil>

Mon, Jun 6, 2016 at 5:08 AM

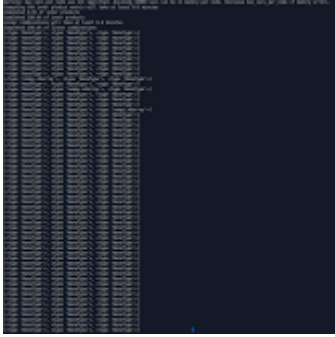
Hello Brandt,

thanks for your fast reply! I guess I will be using the "compute_POD_matrices..." methods or the file-based distributed "compute_decomp" method (which are both working for me) anyway but for comparison it would be nice if the VecHandleInMemory code would be working, too.

I attached a code example and a screenshot of the output from the example using VecHandleInMemory. The modes are only stored in three processes and not broadcasted to all of the 125 processes. I am using modred 2.0.1 with python 2.7. Of course I can gather the modes manually on rank 0. I think in that case the result is the same as when using VecHandleTextArray.

Best,
Tobias

[Quoted text hidden]

2 attachments

failing_example_output.png
122K

 **failing_code.py**
2K

Brandt Belson <belson17@gmail.com>

Mon, Jun 6, 2016 at 9:34 AM

To: Tobias Leibner <l_tobi01@uni-muenster.de>

Cc: "Tu, Jonathan H CIV NSWCCD, BANGOR, 7102" <Jonathan.Tu@navy.mil>

Thanks for the email and details, Tobias. We'll look into this and get back to you. Despite our tests, it's possible we have a bug because the InMemory classes aren't used in a parallel setting as much as the other classes.

Best,
Brandt

[Quoted text hidden]