Seismic Laboratory for Imaging and Modeling

Context

GA Tech Seismic Laboratory for Imaging and Modelling

> *The Seismic Laboratory for Imaging and Modelling---SLIM, is a widely recognized world leader in the development of the next generation of seismic acquisition and imaging technology for the oil & gas industry. SLIM's interdisciplinary research team, with direct involvement of faculty from Computer Science and Mathematics, has leveraged recent developments in transformative fields of compressive sensing and machine learning ('big data') to drive innovations in exploration seismology.*

Department Website
Github Repo

Workloads

From Mathias on 8/31:

Our workflow on azure can be summarized with two Julia packages:

> *https://github.com/microsoft/AzureClusterlessHPC.jl*

Which is the MSFT package interfacing azure batch to Julia. And:

> *https://github.com/slimgroup/JUDI4Cloud.jl*

Questions to our team:

- Is there any path for us to implement functionality to port workloads from Azure Batch to Bacalhau?
- Is there any path for us to migrate their Judi4Cloud workloads by implementing the AzureClusterlessHPC libraries?

Storage Estimates

100TB (rough order of magnitude estimate across many workloads)

Phil's Notes

- "If azure doesn't give us any credit, then we won't use it any more."
- They utilize https://github.com/microsoft/AzureClusterlessHPC.jl, which is an Azure library to orchestrate Azure batch jobs.
- Their application code is pretty simple, if you ignore the julia stuff, (https://github.com/slimgroup/JUDI4Cloud.jl/blob/master/examples/modeling_basic_2D.jl) - just a massive map-reduce job.
- "Everything they do is fundamentally parallel. Single jobs don't make sense."
- "Everything they do is an iterative map-reduce, so we need to be able to scale out and orchestrate massive numbers of jobs."

Meeting Notes

9/20 Email from Mathias

On Tue, Sep 20, 2022 at 1:04 PM Louboutin, Mathias <mlouboutin3@gatech.edu> wrote:

> Hi
>
> Thank you for the follow up. Lemme try to give you some more details.
>
> In JUDI (and similarly in the azure batch interface) the map reduce is done in two steps.
>
> - First "map" by spawning multiple tasks. This is done at these lines in JUDI https://github.com/slimgroup/JUDI.jl/blob/b17e60486866783df4d87a703fb1d5df982a5f30/src/TimeModeling/Modeling/propagation.jl#L26
> - Seconnd reduce. This part is done in JUDI as a binary tree reduction to reduce the memory traffic and is implemented at https://github.com/slimgroup/JUDI.jl/blob/master/src/TimeModeling/Modeling/distributed.jl
>
> The bulk of the communication (i.e dispatch arrays and such to the worker) is handled by julia's Distributed package (part of the core of Julia) and in the azure batch part, it is handled through file serialization on azure blob.

Let me know if you need more details and I'll be happy to dig up any detail necessary.

Cheers

Mathias

From: Wes Floyd <wesfloyd@protocol.ai>

Date: Tuesday, 20 September 2022 at 11:47

To: Louboutin, Mathias <mlouboutin3@gatech.edu>

Cc: Phil Winder <phil@winder.ai>

Subject: Bacalhau & GT Slim Followups: Map/Reduce Style Examples

Mathias,

Thank you again for your time last week taking us through your cloud workloads.

In regards to our discussion around "map reduce like capabilities" in your existing system - could you help direct me to a few specific examples of how this capabilities has been built with Azure Batch/HPC and Julia?

For example you shared the modeling_basic_2d and fwi_3d_overthrust examples with us last time. Can you help us drill a layer deeper and call out where you've built the map/reduce style capability - either in those examples or in the broader JUDI library?

--

9/12 with Mathias (Phil's Notes)

- Previous context:
  - Azure HPC - custom Julia stuff
  - https://github.com/microsoft/AzureClusterlessHPC.jl
- https://github.com/slimgroup/JUDI4Cloud.jl
  - Azure batch? This is a natural wrapper for azure batch.
  - https://github.com/slimgroup
  - JUDI4Cloud is the workhorse of everything we do
- Workloads in general? What data sizes? What rough order of magnitude?
  - 2 sizes. Academic size, industry size.

    - Basically a map-reduce problem. E.g. thousand jobs running a map, reducing into a single result, e.g. weighted sum.
    - Usually requires external data, usually between 1GB and 100GB.
    - e.g. full dataset for industry dataset is 10-100TB
    - Reduces to single multidimensional array, about 1GB-100GB max.
- Execution environments. What environments do you have? On-prem? Cloud?
    - 60-70% on-prem. Mainly because we don't have the software stack to run in the cloud.
    - E.g. have a deal with department of energy, sharing supercomputer via grant (10000 compute hours per month)
    - And one small DGX workstation.
- Do you have a simple example? Do you use docker?
    - Very basic example of what they do: https://github.com/slimgroup/JUDI4Cloud.jl/blob/master/examples/modeling_basic_2D.jl
    - This example doesn't have any IO, doesn't pass or use any data
    - If azure doesn't give us any credit, then we won't use it any more.
- How do you pass data?
    - All the comms is passed to blob storage, then the job reads it from blob storage. Azure blob storage.
    - Usually requires to hacking around to copy files onto the node, then reading it on the node.
- Is there an opportunity for us to provide some extra compute capability?
    - Maybe. Let's try. But don't want to pay for it or anything. Need to test mid-sized problem.
    - Trying to figure out what's the best platform to run that simulation?
- Comments
    - Need flexibility, need to be able to port away from clouds. We've already tried to build a compute agnostic UI.
    - I'm looking for a way of providing an interface so that it runs on any cloud provider.
- Is your data publically accessible?
    - Everything we do is public. All the data is public. Have an FTP server, script will download it.
- Do you have an example that demonstrates scale? Use of data? What is an example of your "mid-sized problem"?
    - github.com/slimgroup/JUDI.jl -- examples/scripts/modelling_basic_2D.jl - simple example
    - examples/scripts/modelling_basic_2D.jl - simple example
    - examples/scripts/fwi_example_minConf.jl - mid-size example, uses segy data - read then sent around to the nodes then gathered back together.
    - examples/scripts/fwi_3D_overthrust_spg.jl - mid-size industry example. Test to see how you create a quick dataset.
    - Everything is containerised.

- Did you have to port the workloads to julia? Or the old stuff was already using it?
  - We were using Matlab, but was awful, so started new code/software/etc., was moving to azure/aws, so Julia was a natural choice. Didn’t need old stuff.
- Julia stuff - clustermanagers.jl - interfaces with clusters.
  - AzureClusterlessHPC.jl - same but for azure - but not really - a taskfile that sends something away - dumps a bunch of task files to blob storage, and it monitors azure blob storage to wait for the result.
- Is there any value in doing a single job?
  - There isn’t enough space to to do this, everything is fundamentally parallel.
  - Everything we do is an iterative map-reduce, so we need to be able to scale out and orchestrate massive numbers of jobs.

9/12 with Mathias Wes’s Notes

Recording here

Azure Clusterless is a wrapper for Azure Batch

Judi4Cloud is the majority of the workloads
Invertible cloud is second most popular

Examples to run:
https://github.com/slimgroup/JUDI4Cloud.jl/blob/master/examples/modeling_basic_2D.jl

Everything running as a remote container

Data storage: Azure Blob used as intermediary between desktop and HPC Batch
Serialization to Blob to pass arrays around

We do not have native Azure I/O, copying the file on the node, then reading as a standard file

Data sizes:

- academic problem size
- Real world problem size: talking to Azure and AWS to get the resources for it.

Basic map/reduce type problem, few thousand tasks run. Each task varies between few GB to hundred GBs
Industry workload data size between 10-100 TB

Not Hadoop explicitly, but parallelized

60-70% of what we do run premise

On premise resources are limited, 10k CPU hours per month
One single DGX server

Looking for ways to scale up research
For Industry and Funding: they ask if we can manage large problems
Want to highlight research and bring interest

Judi4Cloud is meant to be open for many workloads

Next steps:

- How do we port some of the code to Bacalhau?

Phil:
Is the data publicly accessible?
Yes, most data lives on an FTP server

We can run a simple job, submit a single job at a time, but maybe a bash script to scale out the workload
Do you have an example that runs on larger scale?

Basic: Judi/examples/scripts/
Medium:
https://github.com/slimgroup/JUDI.jl/blob/master/examples/software_paper/fwi_3D_overthrust_spg.jl
Examples are containerized

Julia code is performing the orchestration
Dumps a bunch of task files on Azure Blob, Batch monitors blob and runs file when they arrive
Julia is writing config file which Batch then reads

Is there value in running the whole thing in a single process/job?
In practice, a single task will use all the resources in one node, no space to have everything run on one node. Everything is fundamentally parallel, followed by a reduction

How does Bacalhau handle parallelism?
We have some functionality to manage sharding of a job

Product gaps:

- Workflow orchestration

Use Case gaps:

- Rewriting I/O to use CID volumes, instead of Object storage (azure blob)

How was the past migration?
Matlab ecosystem (old software stack)
Started moving to Julia, research took a new direction, new software and new code
Started with AWS, then Azure

We didn't rewrite much of the old code

Portability

8/31 Guidance from Mathias

Our workflow on azure can be summarized with two Julia packages:

https://github.com/microsoft/AzureClusterlessHPC.jl

Which is the MSFT package interfacing azure batch to Julia. And:

https://github.com/slimgroup/JUDI4Cloud.jl

8/24 GT EAS Sync with Felix Herrmann

Azure & AWS batch to do massively parallelizable task
Master free workflow
AWS Julia workflow, Azure batch as backend for Julia

Seismic imaging, highly optimized code, multi-threading

HPC and MPI GPU workloads

Clusterless HPC

Also working with Oil and Gas industry

Summit, BermLoader

Derivative datasets need to be made public
Australian

Another group to investigate:
https://osokey.com/