Optimizing Seismic HPC in the Cloud for Performance and Cost

Society of HPC Professionals
Lunch & Learn
25 June 2020

Branden Moore
HPC & Benchmarking Manager

Numerical Algorithms Group

NAG

Founded 1970

Started from UK academia

Offices in UK, US & Japan

NAG Products

NAG Library

Fortran Compiler

Algorithmic Differentiation

NAG Services

HPC Managed Services

Technology evaluation & Benchmarking

Accelerator development

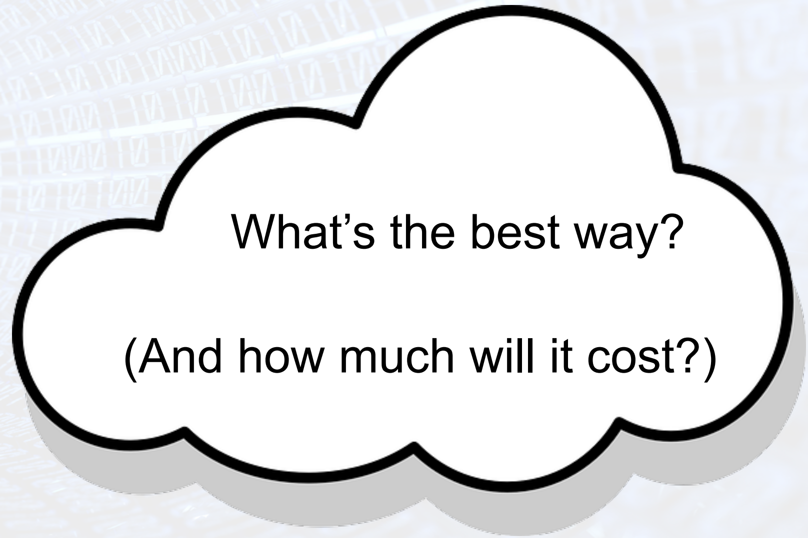
Code Porting / Optimization

Cloud HPC Migration

HPC in Cloud?

Companies ARE moving HPC workloads to the Cloud

- Even Oil & Gas companies



This talk: Seismic HPC from the cloud

- Hypothetical Seismic workload (RTM)
  - (Extremely) Brief introduction to RTM
  - A few things to consider optimizing, and dials to turn

- Explore moving an RTM workload to cloud
  - What cloud resources to use?
  - How to pick the best resource for the workload?
  - Cost-to-Solution vs Time-to-Solution

RTM in a tiny nutshell

- Basically doing an ultrasound of the earth
  - Emit shock waves into the earth, they bounce back - reveals subsurface image
- Reverse-Time-Migration (RTM) is taking this data, producing a volumetric image of the the subsurface
  - (–) Large amounts of input data (TB)
  - (–) Huge amounts of temporary data (PB)
  - (–) Expensive computation as well
  - (+) Each “shot” is computationally independent
- Adding physics/fidelity → Even more computation

Hypothetical Baseline – in-house RTM

On-prem cluster

- 480 Nodes (~1PF)
- Dual socket Intel Broadwell,
  16 cores/socket, 128GB RAM
- InfiniBand – 100Gb/s

Lustre-based storage

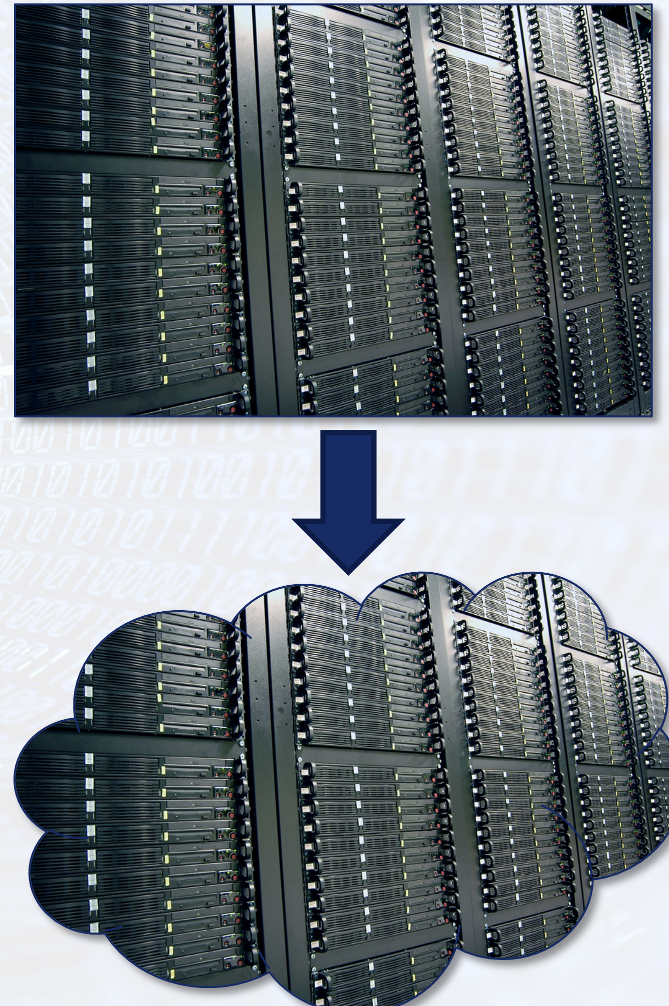
- 8PB storage, 256GB/s bandwidth

Full Workload

- In-house developed RTM code
  - Isotropic acoustic model
- ~10,000 shots per campaign
  - (~350 TB raw seismic data)
- Each Shot
  - Reconstructing to 2048^3
  - MPI / domain-decomposed to 4 nodes
  - 50TB snapshot data
- Full cluster, ~60 days

Target environment – "The Cloud"

- "Lift & Shift"
  - Replicate our nodes
  - Replicate our storage (and how it's used)
  - Replicate our scheduler, queues, etc.
- Very difficult to match exactly
  - VM instance types differ
  - Interconnect may be different
  - Storage totally different
- Almost NEVER cost effective



Lift & Shift - AWS

Compute nodes

- On Prem - Dual socket Intel Broadwell, 16/32 cores, 4GB RAM/Core (no HT)

VM Instance	vCPU	RAM	Networking	Architecture	List Price / hour	$ / HW Core-Hour	$/hr @ 15,360 cores
m5.24xlarge	96	384	25Gbps	Skylake/ Cascade Lake	$4.608	$0.096	$1,474.560
m5n.24xlarge	96	384	100Gbps EFA	Cascade Lake	$5.712	$0.119	$1,827.840
c5n.18xlarge	72	192	100Gbps EFA	Skylake	$3.888	$0.108	$1,658.880
c5.18xlarge	72	144	25Gbps	Skylake	$3.060	$0.085	$1,305.600
c5.24xlarge	96	192	25Gbps	Cascade Lake	$4.080	$0.085	$1,305.600
c5a.24xlarge	96	192	20Gbps	Rome	$3.696	$0.077	$1,182.720

Using on-demand Pricing

Storage

- Build your own?
- Amazon FSx for Lustre – managed service – various performance tiers
  - 256GB/s @ Persistent 50 – 5120TB @ $0.14/GB-month -> ~$1k/hr

Lift & Shift - AWS

On-Prem

- Compute
  - 480 Nodes (~1PF)
  - InfiniBand – 100Gb/s
- Lustre-based storage
  - 8PB storage, 256GB/s bandwidth

- ~60 days compute

AWS version

- Compute
  - 480 Nodes c5n.18xlarge - $1,658/hr
  - EFA-based 100Gb/s
- Storage
  - 8PB Lustre FSx @ 400GB/s - $1,555/hr

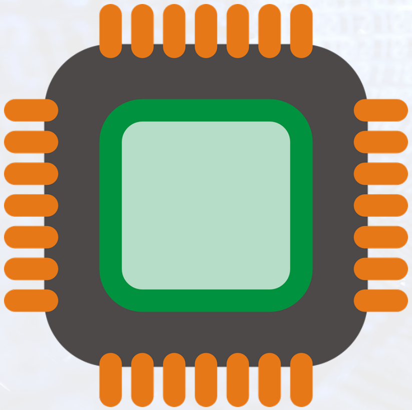
- ~47 days compute
- Total job cost: ~ $3.9M

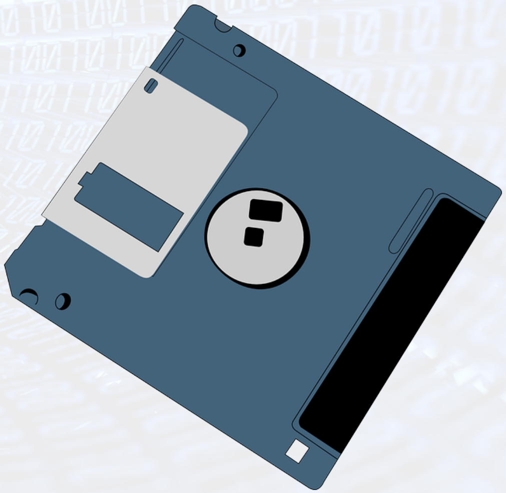
How can we do better?

- Change the application?
- Change the platform?

Change the application

- RTM has two major sources of cost:
  - Time-stepping wave equations (computational cost)
  - Image condition (storage)





Change the application: Time-stepping

- Time-stepping often (but not always) driven by wide stencil calculations
- Highly sensitive to vectorization and caching, but… driven by the system's memory bandwidth
- Thus common strategies for optimizing this:
  - Accelerate time-stepping with GPUs (high bandwidth devices)
  - Domain decomposition over MPI (more than one node per shot)
  - Optimize cache utilization in the stencils

Change the application: Image Condition

- The image condition is usually computed in two passes:
  - Forward pass: periodic snapshots made
  - Reverse pass: convolved with forward pass by retrieving snapshots, accumulated into image
- The primary difficulty is dealing with snapshots, we can improve this in the following ways:
  - Store fewer snapshots, but recompute them in reverse pass (recomputation strategies)
  - Domain decompose enough that aggregate storage of all nodes can store all snapshots

How can we do better?

- Change the application?
- Change the platform?

Change the platform

What all is available?

- Instance types
  - Memory / CPU Speed / # cores / Storage ratios
  - CPU architecture
    - What about AMD? ARM?
    - Lots of instances masquerade – do you know what you're running on?
  - Accelerators
  - Are we "full machine" size, or some partial segment?
    - If you aren't at least full-socket, you're sharing resources
- Interconnect
  - Some platforms have limited support for advanced networking (ie: IB, EFA)

Picking the hardware

Let your application be your guide

- Do a good profile of your application
  - Be aware of what you can profile on the cloud
- Know your bottlenecks
  - CPU-bound? (Rarely, but possible)
  - Memory-bound?
  - Interconnect? - Latency or Bandwidth?
  - I/O?

Addressing bottlenecks

- **Compute-bound**
  - Domain-decomposition
  - Accelerators
  - Higher core-count nodes
- **Memory bandwidth**
  - Explore alternate architectures / instance sizes
  - Accelerators
  - AMD EPYC, AWS's Graviton (ARM) have higher bandwidth than Xeon
- **Interconnect Bandwidth**
  - Often scales with size of VM instance
- **Interconnect latency**
  - Some clouds have HPC-focused networking – IB, EFA.
  - Placement groups
  - Reduce node count?
  - Async communications?

Addressing bottlenecks – I/O

- Evaluate different services & service levels
  - Filesystems: NFS, Lustre, BeeGFS, NetApp, ClusterStor
  - EBS volumes / Managed Disks / Persistent Disk
  - Direct to Object store
- Use of local SSDs
  - I/O Optimized Nodes with many large SSDs
- Add I/O nodes to create a "private" filesystem
- Explore Asynchronous I/O options
  - Less pressure on bandwidth

Build a cost model

- Too many options to keep track
- Explore options available
  - Make sure you know what you're getting!
- Understand how options interplay
  - Increase I/O bandwidth, decrease runtime
- Build a spreadsheet
  - I/O, Compute, Network
  - Bandwidths, latencies, FLOPS, etc.

Focus on Cost-to-Solution

- Cannot focus on $/hr for a resource
- Does including it speed up the job enough to cover the cost?
- If a job runs faster by using a more expensive resource, the total cost may be lower

- Azure example: HC vs HBv2
  - HC - $3.49/hr, 44 cores
  - HBv2 - $3.96/hr, 120 cores

RTM Workload Exploration

- 2048x2048x2048 reconstruction volume
- 7500 Timesteps, Snapshotting every 10 steps
  - ~50TB of snapshot space / shot
- Total RAM / Shot: ~500GB
- 10,000 Shots

- Assume raw input data lives in S3
  - no extra filesystem required for it

Options

- Storage
  - Lustre FSx – various performance levels 50MB/s/TiB -> 200MB/s/TiB
  - EBS – up to 500MB/s/Volume (Can attach multiple volumes)
  - On-node NVMe/SSD/HDD
- Compute Options
  - c5n – 36 core “Skylake”, EFA, no disk
  - c5d – 48 core “CascadeLake”, 3.6TB NVMe
  - [cmr]6g – 64 core ARM “Graviton”
  - c5a – 96 core AMD Epyc “Rome”

Try different Storage options

Lift & Shift: $3.9M

Use 4x c5n.18xlarge for compute / shot

- Lustre FSx Scratch-level - 6PB (120*50)
  - 200MB/s/TiB @ $0.14/GB-month
  - Shared across entire cluster

$1.97M

- EBS – 1 Volume
  - 13TB for 500MB/s @ $0.045/GB-month

$3.62M

- EBS – 2 Volumes
  - 26TB for 1000MB/s @ $0.045/GB-month

$2.66M

Try different Compute options

Lift & Shift: $3.9M

Use 6PB Lustre FSX Scratch, solve 120 shots together

- 4x c5.24xlarge - $4.08/hr (each)
  - Newer Xeon, 48 cores

$1.77M

- 4x c6g.16xlarge - $2.17/hr (each)
  - AWS Graviton, 128GB, 64 cores

$1.10M

- 1x r6g.16xlarge - $3.2256/hr (each)
  - AWS Graviton, 512GB, 64 cores

$1.66M

Try different Concepts

Lift & Shift: $3.9M

- Large local SSD
  - 1x i3en.24xlarge – 60TB local SSD storage
  - 48 Skylake cores
  - $10.848/hr

$1.64M

- Combine Compute + Storage
  - 4x c6g.16xlarge + 1x i3en.24xlarge

$1.16M

- All In-Memory
  - 2x u-24tb1.metal – 24TB RAM
  - 224 Skylake cores
  - Unknown price - $50/hr?

$1.74M

- Over-decompose w/ SSD
  - 16x c5d.24xlarge - ~3TB SSD
  - 48 CascadeLake cores
  - $4.608/hr

$2.52M

Takeaways

- Understand options available
  - Know what you're getting – and what you're not
- Understand your application
  - Know its bottlenecks
- Understand how options interplay
  - Increase I/O bandwidth, decrease runtime
- Build a spreadsheet cost model
  - Interplay and balance not always obvious

Price-Per-Hr vs Cost-to-Solution

Real world challenges

- Integration with existing infrastructure
  - Job schedulers, File Systems, etc.
  - Policies, exposing infrastructure-as-code to users
  - Some providers have a ”cloud scheduler” that might work for you
    - AWS Batch, Azure Batch
- Availability of resources
  - Cloud really isn’t infinite
- Traditional global filesystems are expensive
    - Move data from cold storage to hot storage during a job – return it later.
- Everything constantly changes

NAG Cloud HPC Migration Service

- New service offering from NAG
  - HPC Cloud migration
  - Cost-to-Solution focused platform & application optimization
  - Partnering with the “Big 3” providers

https://nag.com/cloud-hpc

Experts in High Performance Computing, Algorithms and Numerical Software Engineering

www.nag.com | blog.nag.com | @NAGtalk