Right Sizing HPC Resources

Michael Senizaiz - CTO @ R Systems

What is HPC?

Expensive nationally funded supercomputers?

Large and complex?

Bleeding edge hardware and software?

$$$$$$$$?

What is HPC?

Leveraging big labs expertise

Targeting best performance per $

Targeted HPC

Identify platform for algorithms

- CPU or Accelerator selection
- Footprint

Application feeding

- More than cores - Data movement!
- Scaling = Storage and Network performance

Setting ROI to $/job

- Build scenarios for fastest and good enough
- Include often ‘forgotten’ costs

Your HPC

"What are others doing in my industry?"

Generally, something completely different

Application secret sauce

Functional algorithm defines processor/accelorator

Y=MX+B == Multiply+Add Capabilities

Actual application workflow defines scaling requirements

(Data Stored) -> *(Data Transferred)* -> **(In-Memory) -> [Actual Processing] -> (In-Memory)** -> *(Data Transferred)* -> (Data Stored)

Add your additional steps here. This is an EP example

Scaling lies here

(Storage Performance) *(Network Performance)* **(Node Performance)**

Guidelines

- Make history history
  - Revisit tech that missed the mark years ago
  - Don’t follow old internal ‘per-core’ or ‘per-node’ metrics
- Efficiency over FLOPS
  - Data must be present to be accounted for
  - Ignore if Top500 competitor
- Don’t assume, hypothesize
  - Always test to validate
  - ‘Should’ rarely survives scrutiny
- Profile if available
  - Visualize areas of opportunity
  - See Amdahl's law

Old Standard

- IPC gains == HPC gains
  - Obvious benefits from architecture changes
  - Broad segment application
  - Easily prove replacement ROI
- Relatively Limited Selections
  - 1-4 core counts/socket
  - 7.2k or 10k Disk?
  - Ethernet or Exotic Interconnect
    - Limited ‘Exotic’ software support

New and Confused

IPC gains != HPC gains

- Marginal overall improvements
- Marketing numbers, Top 500 focused
- Largest ROI sometimes utility bill and warranty
- 2x+ FLOPS often get 10% improvement per core or per node!

Too Many Selections

- 10+ core counts/socket
- Disk, SSD, NVDIMM, PCI-E, etc.
- Ethernet (1,10,25,etc), RoCE, Infiniband (40,56,100,etc)
- Accelerator? GPU? PHI? FPGA?

Example CFD Benchmark

CFD vendor assumptions

Processor is lynchpin

AVX2 required for any meaningful performance

10Gb or better network sufficient

‘Cloud’ testing will validate hypothesis

Example CFD Benchmark

CFD Results

144 3.2GHz Sandybridge Cores + IB + Lustre

Walltime: 29,763 Seconds. I/O: 531 Seconds.

$/job = $95 ($0.08pch)

21 jobs/week @ $1,995

144 3.3GHz Haswell Cores + 10Gb + SSD NFS

Walltime: 77,231 Seconds. I/O: 1,335 Seconds.

$/job = $256 ($0.083pch)

8 jobs/week @ $2,051

Resource Diversity

- Target most important workflows with distinct node sets
  - Requires high quality forecasting
  - Doubly so for mixed architectures
  - Single app? Great!
- Or meet in the middle
  - Flexible resource pool
  - Simplified development environment
  - Possibly over-built for some periods
- Focus on what matters
  - User expectations
  - Move non-priority work (Dev/Test/etc.) to ‘The Cloud’

Benchmarking- Node Configuration

Start tests small - Use a single node when possible

- Move to multi-node to validate same number of cores
- E.g. 16x2 vs 32x1, 16x4 vs 32x2

Storage and Networks (and code) control scaling efficiency

Run the CPU Gamut

- Get the biggest processor available
- Use BIOS to test CPU varieties
- Disable cores, features, set clock, w/ and w/o HT, etc.

Test local I/O impact

- If it fits, RAMDISK
- SSD, BBU RAID, Standard Spindles, etc.

Storage

Scratch Space

Density restricts performance

- Core density greatly outpacing local I/O options
- Dense rack configurations further restrict
- Multi TB requirements restrict further
- SSD/NvDIMM great for small datasets

Transition to Parallel FS

- Move disks away from node, possibly free up floor space
- Scale benefits random I/O and aggregate bandwidth
- Temporary nature allows islands, no need to be global

Storage

Global Shared Storage

NFS

- No single namespace requirements (generally)
- Not using MPIIO
- Shared input files
- Tiny ‘sync’ writes can be async

Parallel FS

- Single namespace
- Great I/O scaling
- Shared output files - no waiting
- Reduced dataset hot-spotting

Storage Sourcing

Name Brand Appliances

Pro's

- Turn-key installation
- High-quality support
- 'Free' systems architecture planning

Con's

- Last years technology
- Licensing & Support = ½ - ⅔ of price

Storage Sourcing

BYOD

Pro's

- Lowest $/TB = Lower $/Job
- More technology options

Con's

- Requires internal experts
- Multiple support channels, Hardware vendors and Open Source community
- Much longer lead-time for testing, tuning, and validation

Storage Testing

- Don’t rely solely on googled benchmarks
  - Useful for validation
  - Unlikely to be similar to production
  - Many FIO > IOR for many workloads
- Test at scale whenever possible
  - Best single-job performance might fall over at 10 jobs
  - What about 100 jobs?
  - Work with developers on test cases that minimize core requirements
  - Parallel FS generally scales medium as it scales extra large.
    - E.g. 4 OSS of 20 OSS for 20% I/O scale testing
    - Similar for NFS ‘pools’.

Networks

Network goals

- Simplify management, out of band, and provisioning
- Understand impacts to application
- Understand importance of cable selection at scale
- Not too over-provisioned - $$$/job
- Correct network allocation on run

Network Types

Ethernet

- 1/10Gb - relatively cheap - 25/40/100Gb available. Node and Aggregation layer
- Protocol overhead precludes low-latency applications

Ethernet - RoCE

- Performance closer to Infiniband that Ethernet
- May require changing OFED/MPI stack, runtime arguments, etc.
- Requires HCA and switch support and complex configurations
- Great for Parallel FS

Infiniband/Omnipath

- Lowest latency
- Highest Bandwidth (56/100/200Gb+)
- Most expensive

Small Packet (Latency Sensitive) Example

Tested 256 Cores MPI communication. 8 Byte Packets

Network	**8 Byte Latency**	**Slowdown**
FDR	0.0029ms	
10Gb RoCE*	0.0034ms	17%
10 Gigabit Ethernet	0.088ms	2934%
	8 Byte Bandwidth	**Slowdown**
FDR	2.75MB/s	
10Gb RoCE*	2.3MB/s	19%
10 Gigabit Ethernet	0.09MB/s	2956%

*RoCE used Infiniband HCA and Switch, but 10Gb DAC

Large Packet (Bandwidth Sensitive) Example

Tested 256 Cores MPI communication. 2 MByte Packets

Network	**2 MByte Latency**	**Slowdown**
FDR	10.8ms	
10Gb RoCE*	40.9ms	377%
10 Gigabit Ethernet	98.9ms	915%
	2 MByte Bandwidth	**Slowdown**
FDR	191MB/s	
10Gb RoCE*	48.8MB/s	391%
10 Gigabit Ethernet	20.2MB/s	945%

*RoCE used Infiniband HCA and Switch, but 10Gb DAC

Walltime Effect - RDMA Messaging

Small messages:

1 Billion messages would take

FDR: ~45 Minutes
10 Gig RoCE: ~1 Hour
10 Gig: ~22 Hours

Large Messages:

1 TB Data would take

FDR: ~1.5 Hours
10 Gig RoCE: ~6 Hours
10 Gig: ~14.5 Hours

*RoCE used Infiniband HCA and Switch, but 10Gb DAC

Infiniband - Best Fit

- Fastest isn't always faster
  - Latency is usually king over bandwidth
    - Remember what we said about assumptions. YMMV
  - Latency differences are marginal between IB generations
  - Client can set port-generation speed for test cases
  - HCA Can also be used as RoCE with DCB ready network switch
  - Blocking factor should take largest MPI jobs into account
    - Smart schedulers can prefer jobs on same switch or sets of switches
    - Use a multiple of job sizes to select cores per low-blocking segment

Infiniband - Real Applications

Network	OpenFOAM Mesh	Ansys Fluent
10Gb	8H 32M	DNF (WALL 48H)
DDR Infiniband	2H 30M	N/A
QDR Infiniband	2H 28M	N/A
FDR Infiniband	2H 24M	8H 50M
EDR Infiniband	2H 15M	4H 46M

Analysis Paralysis

Put it all together - Part I

Nodes

- Price out top 2 picks for node configurations
- Best wall-time and Best Value
- Include support costs per *N* nodes

Storage

- Appliance or BYOD
- Select configuration that doesn't impact walltime at scale
- Include additional internal BYOD storage expertise staff

Network

- Include redundancy where important - storage, management, etc.
- Don't forget the right cables!

Analysis Paralysis

Put it all together - Part II

Facilities

Get rack and power consumption for each overall configuration

Not in your budget? Find a way to share for the right scenario

Licensing

Can be more expensive than the hardware

Per node/socket/core? Per accelerator?

Sizing

$/Job is good, how many jobs per day, week?

Account for utilization periods and turnaround availability.

All day? Process over nights and weekends? Mixed?

Conclusion

Less is more

- Improve backend performance to get more work to the CPU
- Address bottlenecks before optimizations
- Used cycles > Available cycles

Leverage ‘The Cloud’

- Internal cluster for routine and priority workloads
- Less complex workloads or easily moved jobs
- Special projects, peak shaving, testing and development
- Lessons learned help select reasonable providers

Questions?