WHAT CAN HPC LEARN FROM CLOUD COMPUTING

Dale Southard, Salishan 2016
AGENDA

- Vis and Post Processing Overview
- State the Problem
- Tell a Joke
- Look at Possible Solutions from Hyperscale
- Take question from Michael Wolfe
HPC POST PROCESSING
Technology Toolkit, 2016

Remoting Technologies
1. Move the Data
2. Move the Triangles
3. Move the Pixels
4. Move the RGB

but three of them are not very useful at scale

Scaling Technologies
• Contract Based IO
• Distributed Rendering
• Parallel Compositing
• Pervasive Acceleration
BUT BARRIERS REMAIN

Visualization throughput limited by ingest

MB of Imagery
I/O ISN’T (ONLY) A HW PROBLEM

We are Already Deploying a Lot of HW

Scalable Storage Units have lots of bandwidth

But individual MPI ranks see only a fraction of that

“Storage doesn’t pay the bills, it is a tax on the center”
Shlemiel is hired to paint centerlines on the road.

- On Monday Shlemiel paints 300m of the road
- On Tuesday Shlemiel paints 130m of the road
- On Wednesday Shlemiel paints 100m of the road
- On Thursday Shlemiel paints 80m of the road
- On Friday Shlemiel paints 70m of the road

“Sorry boss, but it takes me longer to walk back to the bucket.”
DALE, THAT’S STUPID
Our Computer Scientists Would Never Use Shlemiel’s Algorithm

How is `strcat()` implemented?

What happens when you `ls -l` in a Lustre directory?

Shlemiel may be hiding behind leaky abstractions
“Don’t buy stuff you cannot afford”

Saturday Night Live, Season 31
DBSYCA EXAMPLES
HOW DO BIG DATA SITES SCALE IO?
WHY DOES NETFLIX RELY ON S3
when a real parallel filesystem is better?

99.999999999% durability

99.99% availability

Effectively infinite storage

RESTful API
WHAT IS RESTFUL?

Representational State Transfer using hypermedia as the engine of application state

- Client-Server
- Stateless
- Cacheable
- Layered
- Uniform Interface
- Code on Demand (optional)
WHAT ABOUT BANDWIDTH?
Data locality is still the key

https://openconnect.netflix.com/
WHAT IF....
Could a REST approach work for HPC?  
And would it make life better?

Can we evolve this...

<table>
<thead>
<tr>
<th>App</th>
<th>App</th>
<th>App</th>
<th>App</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functions</td>
<td>Libraries</td>
<td>fread/fwrite/mmap</td>
<td>Storage</td>
</tr>
</tbody>
</table>

Into this?

<table>
<thead>
<tr>
<th>App</th>
<th>App</th>
<th>App</th>
<th>App</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functions</td>
<td>Libraries</td>
<td>http get/post</td>
<td>Storage</td>
</tr>
</tbody>
</table>
HYPERSCALE EXPANDABILITY
GOING FURTHER
Leveraging resource management

RESTful User Storage Proxy Apps
ACCELERATING HPC VIZ
Move Accelerated Post Processing to the Data

Ideal intercept for in-transit visualization

• Volume visualization
• Photorealistic rendering
• Deep Learning
• etc

Many tools compatible with minimal changes
In Summary:

• We have a good set of building blocks
• But we are bandwidth limited
• Others are dealing with similar bottlenecks
• Watch out for Shlemiel