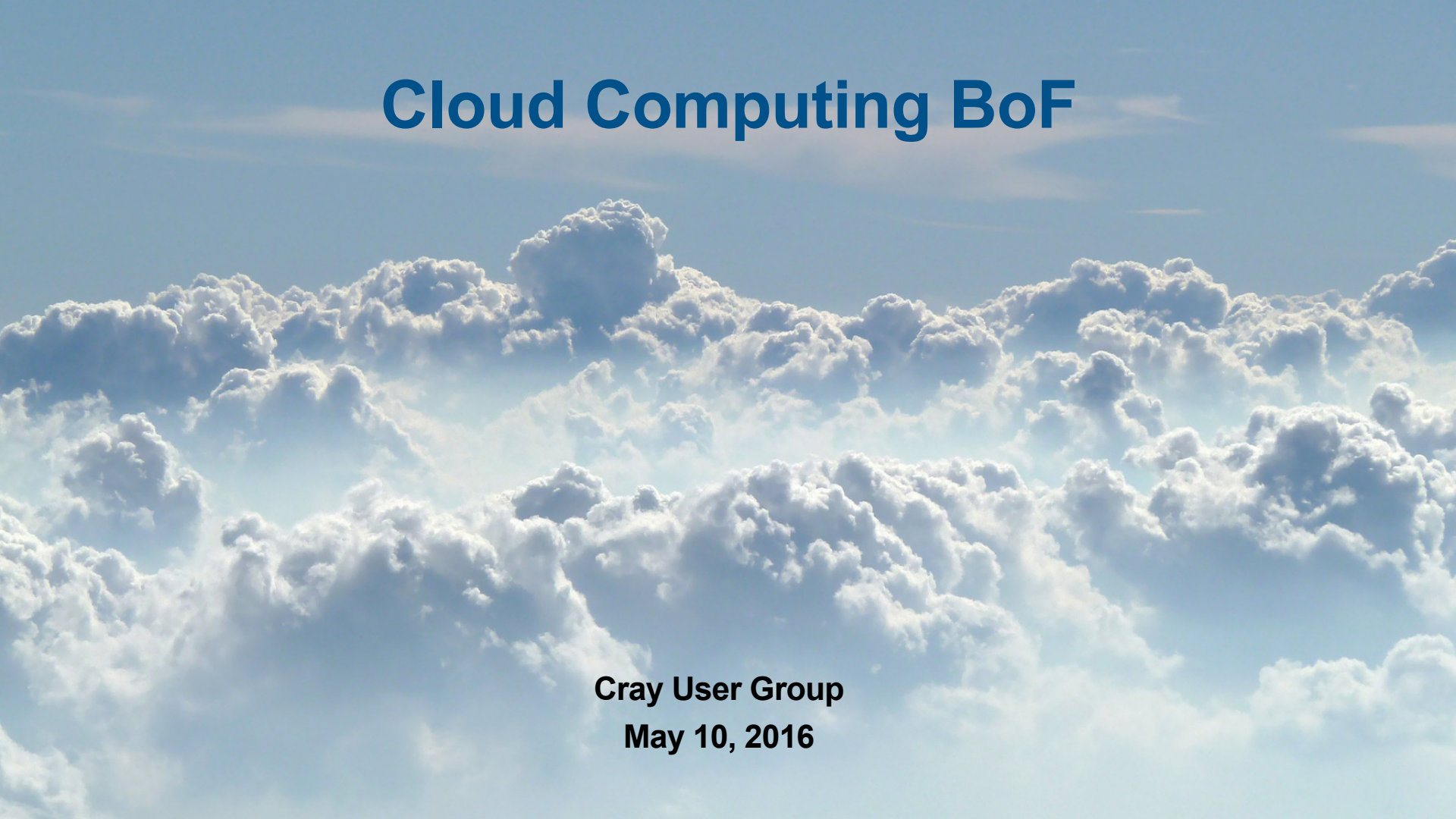
Cloud Computing BoF

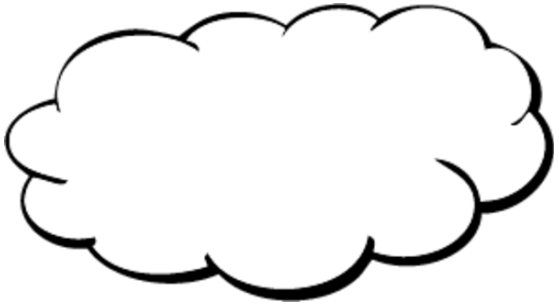


Cray User Group

May 10, 2016

Introduction

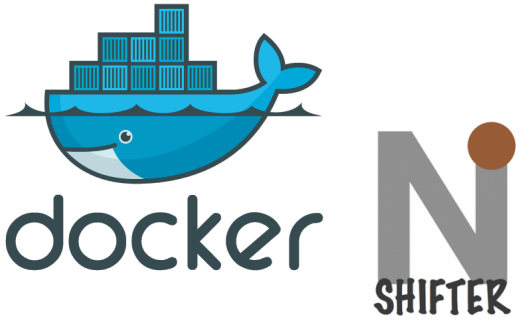
The Future Will be Partly Cloudy



- Ease of use, managed by others
- Quick to deploy
- Elastic
- Continuously updated software infrastructure
- Pay as you go

- More expensive than (well utilized) in-house
- Not good for capability HPC
- Time and expense to move data off/on premises
- Security concerns

Cray's Initial Response to Cloud



Build with certified software stacks

Bundle libraries and dependencies

Build a consistent environment from desktop to supercomputer

Common, standards-based interfaces for system management and provisioning

Phasing in across Cray products over time

Deloitte.

Cloud-based service delivery

Some Cloud-Related Questions

- **What do HPC centers need to compete with the public cloud?**
- **Is running on-premises systems as a *private cloud* important?**
  - If so, what capabilities are needed? User self-service portals?
- **How to make on-premises systems more useful vs. the cloud?**
  - Bursting? Full virtualization for selectable OS kernels? Compatibility with public cloud storage APIs or job launch APIs?
- **Would having *capability* systems in a public cloud be useful?**

Cloud computing survey (ask us, or your Cray account exec)

Remainder of this session…

Working Session

Working session on customer requirements

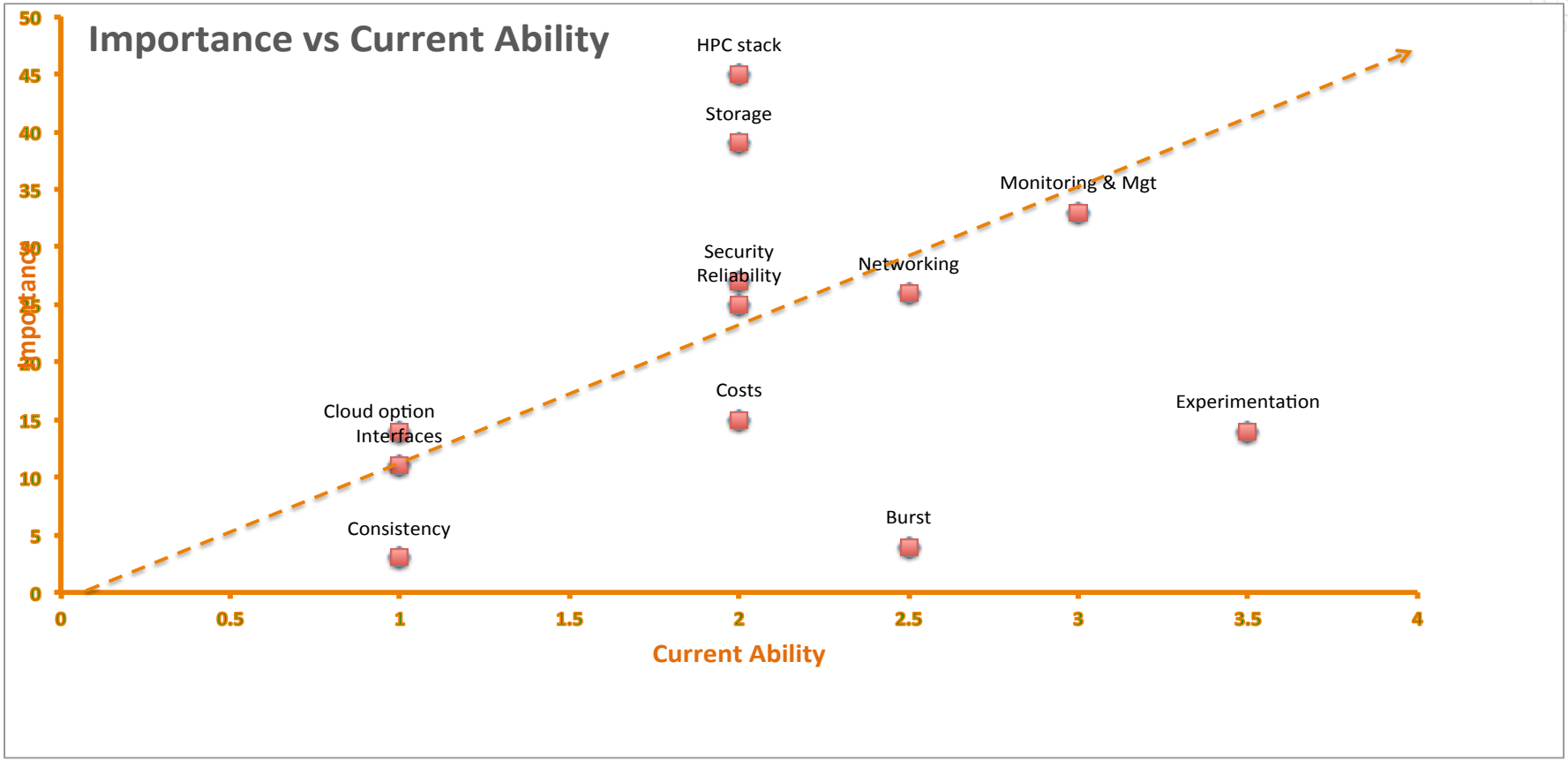
- **Seed Question:**

"In the 2018-2020 timeframe, in response to cloud computing trends, and to better compete or interoperate with cloud computing, Cray users and/or administrators will need the ability to…. <fill in the blank>"

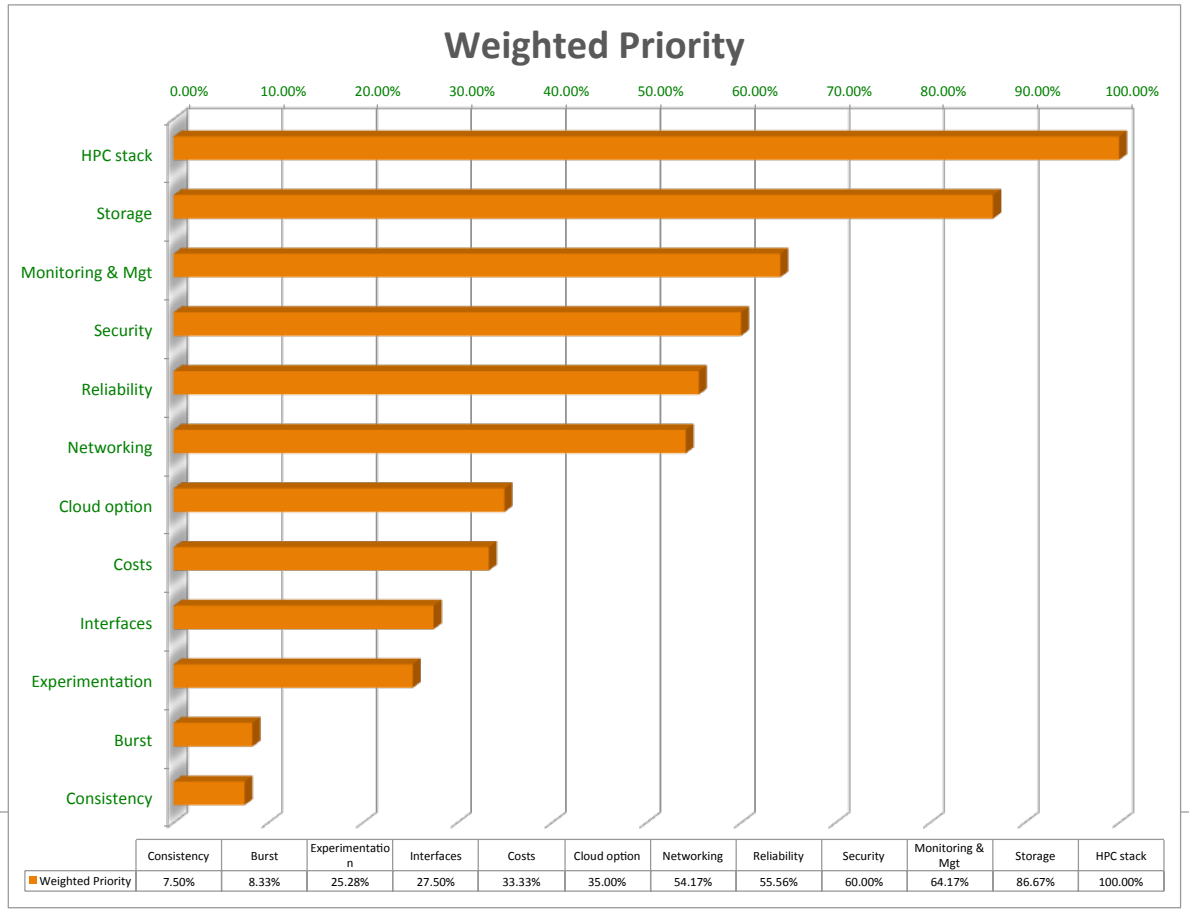
Prioritized Requirements

Requirement (full text)	Requirement (short name)	Importance	Current Ability	Weighted Priority
Complete_flexible_hpc_stack	HPC stack	45	2	100.00%
Storage	Storage	39	2	86.67%
Monitoring_and_management	Monitoring & Mgt	33	3	64.17%
Security	Security	27	2	60.00%
Reliability	Reliability	25	2	55.56%
Networking	Networking	26	2.5	54.17%
Have_the_option_to_use_the_cloud	Cloud option	14	1	35.00%
Understand_and_control_costs	Costs	15	2	33.33%
Interface_easily_with_the_cloud_and_others	Interfaces	11	1	27.50%
Experimentation	Experimentation	14	3.5	25.28%
Burst	Burst	4	2.5	8.33%
Consistency	Consistency	3	1	7.50%

Importance vs Current Ability



Prioritized Gaps



	Consistency	Burst	Experimentation	Interfaces	Costs	Cloud option	Networking	Reliability	Security	Monitoring & Mgt	Storage	HPC stack
Weighted Priority	7.50%	8.33%	25.28%	27.50%	33.33%	35.00%	54.17%	55.56%	60.00%	64.17%	86.67%	100.00%

Your words…


APIs database
Easily Simulate
launch multiple
spare
secure?
Catalog node
Simplify downstream PFS distributed burst
workloads
sequential Price
move web
virtual
remote
important consumption infrastructure justify analytics
networking
Explore
integrity purchases
option
new ability
resource provenance
know
storage
run-time
capacity
fast
limit
use
see
Share
software
real-time
system
jobs
cloud
cycles
extreme
energy
customers/users
points
VMs
Cray
quickly
Profile
Different
private
available full
coordinate tasks
native
convinced
services
Fluid
OS
encryption tools
deploy
sell application scalability
compute
minimal
stability
data
potential
environment cost images manage
provision
operational
maintenance
instantly
accounting Leverage
need
failure
HPC
block integrate
Stream
Train
HTC
sources
reliability
power
hard platforms
results
handling
between
Evaluate
Stage
Track
Admin
stack Change scheduler
models containers access consistent
machine
local
tuning Performance upgrades object

How can you Cray help you learn about the cloud?

- **Data movement –** Once you move the data does the compute stay? What is the scale of the data that you would need to move? Are they partial sets or full sets? Is their an opportunity that the base data may be shared with other organizations? What do you need from Cray to help you be successful?
- **Set up of complex environments** – What do you need to customize the environment? Do you expect software infrastructure automation to build your bursting capability, what monitoring capabilities/tools do you need? How reactive does your scaling need to be? What do you need from Cray to help you be successful?
- **Performance Tuning** - Cloud offers abstractions to its resources, optimizations will be required when developing or deploying complex applications. How do you see handling this? What do you need from Cray to help you be successful?
- **Dynamic and automated management** - what are the challenges for you to manage auto scaling and cloud bursting. Some cloud resources are more expensive than others with different availability -- automatically and dynamically selecting resources according to application and quality-of-service needs, which can change over time is key. What do you need from Cray to help you be successful?
- **Security and privacy -** These are two of the main concerns for Cloud adoption, how big is this obstacle for you? What do you need from Cray to help you be successful?
- **Programming interfaces and tools –** which of your existing software and service components need to reside on the cloud? What application Programming Interfaces will you require? What do you need from Cray to help you be successful?

Detail of Requirement Groupings

Complete Flexible HPC stack

Complete_flexible_hpc_stack

- Be able to deploy web services/databases/analytics after HPC computing runs for data handling
- Change OS for application
- Have a cloud/HPC integrated scheduler(E.g. coordinate distributed database on cloud with HPC compute… Different launches)
- Have a relevant software available
- Integrate analytics and HPC and HTC
- Perform minimal maintenance for software upgrades
- Support VM's and containers
- Update and integrate native software stack on cloud(Docker, Etc.)

Storage

Storage

- Access multiple data sources(possibly remote)
- Do data handling/analyzing from HPC workloads
- Have access to multiple data sources
- Have fast access to high-performance storage
- Move to/from the cloud
- Share block an object and PFS storage
- Track data provenance

Experimentation

Experimentation

- Do the "out there" stuff that is hard to justify
- Explore new computational models
- Simulate downstream load
- Train machine learning system
- Try out new tools
- Understand the potential effect of new loads on operational systems

Security

Security

- Admin's will need the ability to securely provide access
- Ensure data security and integrity
- Know where the date is
- Leverage encryption at rest and in transit with performance
- Provide users secure environment
- Securely manage private cloud infrastructure
- Support multiple authentication and authorization services
- vet user provided system images

Understand and Control Costs

Understand_and_control_costs

- Ability to know what the cloud will cost me and my customers/users
- Provide caps on expenses(burst and tell Price limit reached)
- Sell their spare cycles to cloud users– Need accounting
- Understand power consumption and energy efficiency of use of the cloud

Monitoring and Management

Monitoring_and_management

- Catalog points of failure
- Profile job performance or resource utilization tuning will be important
- See what's running on compute node
- Understand extreme scalability

Burst

Burst

- Deploy diverse workloads instantly
- Fluid computer tasks when local machine is at full capacity
- Know jobs will be returned/run in a timely fashion
- Quickly launch jobs
- Run on -- and move between -- heterogeneous platforms
- Stage data between sequential usage of HPC and the cloud
- Stream work between systems

Have the option to use the cloud

Have_the_option_to_use_the_cloud

- Cray user will need to be convinced to see the benefits to use the cloud
- Evaluate potential software and system purchases
- know the cloud can be an option

Reliability

reliability

- Ensure consistent results
- Ensure stability for operational workload

Networking

Networking

- Simplify virtual networking with reliability
- Stream in real-time data

Consistency

Consistency

Easily sell the idea of cloud to users

Know the results are reliable and consistent

Interface easily with the cloud and others

Interface_easily_with_the_cloud

Have APIs for system and user services

Annex

Requirements generated