<table>
<thead>
<tr>
<th>HPC</th>
<th>HTC</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Science</td>
<td>Business-Social-Security</td>
<td></td>
</tr>
</tbody>
</table>

Scientific, Engineering, Medical, Security, Societal, Business Applications
Application Challenge Areas

• **Experimental facilities**
  – Large data, streaming, some real-time
  – Many experiments or few large ones

• **Observational data from sensors (IoT)**
  – Streaming, continuous, noisy

• **Data fusion with multimodal analysis**
  – Cancer, graphs, ...

• **Science at boundary of simulation and observation**
  – “Steering” experiment with simulation
  – Refining simulations with observations
  – Iterative science workflow (inference)
What is HPC?

• High Performance
• Computing

• Does not necessarily presume tight coupling
• Does not assume high flops
Math, Algorithms, (Software) Challenges

- Fusion, Coupling, Multi-modal, inverse problems
- Noisy, sparse, (systematically) biased
- New data types: Images, Genomes, Graphs, Time series
- Sharing, Security, Privacy
Application Wish List

- Run complex software stack on demand
- Handle job stream failures
- Complex workflows with changing “width”
- Data-aware (intelligent) networks
- Whole system view of compute/data/storage ecosystem for inference
- Performance and usability
- Edge services: security, resource flexibility, etc.
Given a clean slate, what particular I/F and/or defining problems should we target Operations
Big Data (Analytics, Machine learning)

Simulation (classical HPC)
Big Data (Analytics, Machine learning)

Simulation (classical HPC)

co-location vs. convergence, co-design, co-finance
Big Data (Analytics, Machine learning)

Simulation (classical HPC)

co-location vs. convergence

Data Archive
Big Data (Analytics, Machine learning)

Simulation (classical HPC)

co-location vs. convergence

Data Archive

On demand, real time data in/out
We touched on these dimensions (1):

• What is the benefit of convergence, where are the advantages?
• What can you do that you could not do before?
• What is converged: Hardware, Management, Software stacks ....?
• Dynamics of tools and software evolutions much faster in BD
• Are there financial advantages (OPEX, APEX)?
• Difference of use and operations modalities (different user groups/experience)
• Security requirements differ and need acceptance
• Predictability differences (Data sources, formats, data flow peaks, etc)
• Data and machines use differences (1 user, static, vs many users, dynamic)
  • Need to handle complexity of jobs and workflows in converged mgmt.-structure
• Utilization differences
• Recovery/restart differences (especially in shared environment)
We touched on these dimensions(2):

- Is “operations” of a converged infrastructure becoming more complex (one team) or not (another team) ?
- Cost of convergence on application level much higher than on hardware level
- Tradeoff between operational cost and development cost
- Tradeoff between operational cost and operational efficiency
- Points of view: benefits from user’s point of view vs. resource operator’s view
Best options (for now):

Homogeneous (=converged) management on top of heterogeneous hardware
Summary of breakout - Software

Summary

Mission
- Not clear what problem we are trying to solve, the problem is ill-defined
- The 300K for 12 months question

Low-level services and capabilities
- Containers (deployment -- ease of use, consistency, reproducibility, extensibility)
- Programmability, QoS, SDN, SDX – expose finer control of capabilities
- Resource management
  - On-demand
  - Multi-dimensional, look beyond nodes
- Understanding of monitoring systems data (complexity)

Higher-level capabilities
- Build a platform that adapts itself to application
  - We are usually adapting application to platform, but it should go the other way around as well
  - Auto-tuning, adaptive behavior
- Support for fine-grained data streams and services (geo locating, etc.)
  - Edge computing
  - Workflows (better interoperability, fine-grain management)
  - Interoperability of data formats
- Programming models, efficient data analytics for HPC platforms
- Provenance
- Content Distribution Network for scientific data

Economics and Incentives

- Economics of software: “Big Data” part of the world is much larger than “HPC”, they will drive a lot of systems software that is not compatible with the way we run HPC systems – HPC will end up as “the 1%”, luxury end of computing

- Creating incentives: instead of charging users per service hour, charge them per data usage (i.e., create a charging model that will incentivize convergence)