

SOS 13

System Software Panel

(We don't suck as bad as memory)

Barney Maccabe

March 11, 2009



Panelists

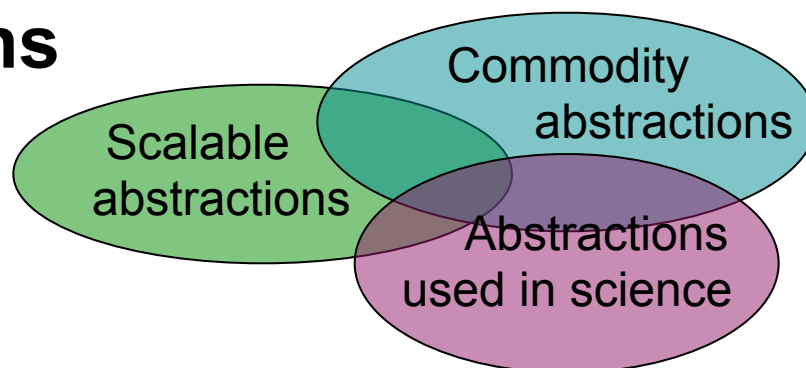
- **Pete Beckman, ANL**
- **Ron Brightwell, SNL**
- **Christine Morin, INRIA**
- **Stephen Scott, ORNL**

System Software

- **Basic I/O System**
 - provides a basic abstraction of the I/O devices
- **Operating system**
 - provides abstractions and mediates resource conflicts
- **Runtime system**
 - software needed to support a programming model
- **Virtualization**
 - resource abstraction, typically encompasses multiplicity (e.g., processes, threads, windows)
 - multiplicity enables deferred management decisions

Abstractions

- In the end, it's about building the right set of abstractions
- Commodity abstractions
 - processes, demand paged virtual memory, dynamic linked libraries, signals, map reduce, ...
- Abstractions used in science
 - MPI, global arrays, ...
- Scalable abstractions



Panel questions

- **Transparency** (exposing resources) versus **Simplicity** (masking complexity): Are these in conflict? How much overhead is too much?
- System **software lifecycle**: This is a niche market, how do we sustain system software? Why is system software different than HPC applications?
- **Fault tolerance/resilience**: is this an OS, runtime, or application issue?
- When do we get to stop supporting **MPI**? From a system software perspective, what is the biggest challenge in supporting MPI?
- For which questions (not limited to the ones above) is **virtualization** not the right answer?