# Economic and management challenges and needs of computational resource providers and industry partners

Chair: Dan Reed (Microsoft Research)
Secretary: Jean-François Lavignon (Bull)

IESP Workshop 2, June 28-29, Paris, France

# Working Group Participants

- Patrick Aerts
- David Barkai
- Sanzio Bassini
- Taisuke Boku
- Iris Christadler
- Hugo Falter
- Alan Gara
- Jean Gonnord
- Andrew Jones
- Kimmo Koski
- Bill Kramer
- Jean-Francois Lavignon
- Dan Reed
- Christian Saguez
- Makoto Taiji
- Peggy Williams

# Working Group Outbrief (Reminder from Yesterday)

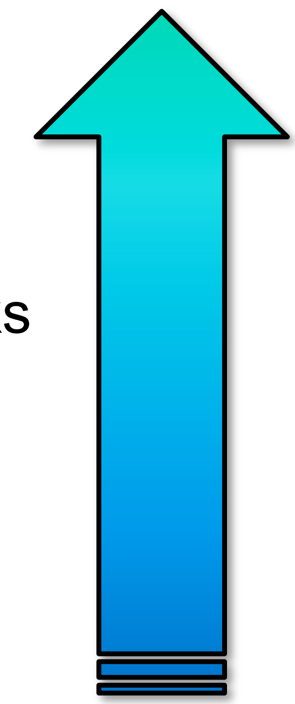
- Discussion focus
  - Metrics define outcomes (choose wisely)
  - Collaboration models and horizons
    - $N \times X \mathrel{!=} X \times N$
  - Regional differences identified
    - U.S., Europe (e.g., PRACE), Japan
  - Strong interest in international collaboration
- High level issues
  - Pre-competitive partnerships desired
    - Horizon should be 5+ years if involving provider competitors
    - Procurement winner(s) known early
  - Co-development implications
    - Shared risk, funding and outcomes
  - Vendor types/sizes have different constraints
    - Risk, time horizon, funding fungibility

# White Paper Outline (Draft in Two Weeks)

1. Introduction/charge (**Reed**)
    - Summary of the working group agenda
2. Lessons from large-scale projects (**Reed,** Taiji, Gonnord, Bassini, Gara)
    - Computing at petascale (U.S., Japan, Europe)
    - Other large S&T projects (ITER, LHC, …)
    - Metrics define outcomes (systems and collaborations)
3. Market size/type and implications (**Williams**, Barkai, Jones, Bassini)
    - Limited experimental facilities or larger commercial market (assume expansion)
    - Software scaling semi-invariant across evolution of Top500 (Jones)
    - Interaction models on issues
        - Specifications, implementation, lifecycle support
    - Decision criteria (feasibility, cost, go/no-go decisions)
4. Collaboration approaches (***Aerts***, Christadler, Kramer, Koski, Boku)
        - Funding profile shapes collaboration options (time and hardware/software balance)
    - Collaboration and relationship (tight to loose) models (Kramer)
    - Links to software types and implications (Aerts)
    - Defining limits of collaboration/competition/creating an IP rights process
        - Vendor, center and region
    - Impact and implications of open source approaches

# Collaboration Scenarios (4)

- Tightly coupled collaboration
  - International governance and funding structure
  - Multi-company development teams
- Collaboration with standardization
  - Definition of standards, test suites, and benchmarks
- Loosely coupled collaboration
  - Focused workshops on software activities
  - Comparison of technical milestones
- Little collaboration
  - Periodic workshops, status reports of regions
  - Voluntary and ad hoc usage of project products



# Relationship Models

- Definition - Organizations - could be
  - Industry providers
  - National labs
  - Universities
  - Consortiums
- Relationship Models
  - Funded Investigation
  - Fully defined purchase
  - Design and Development – no solution
  - Co-design and co-development of a solution
  - Base plus value add
- Multiple models can be used in a program

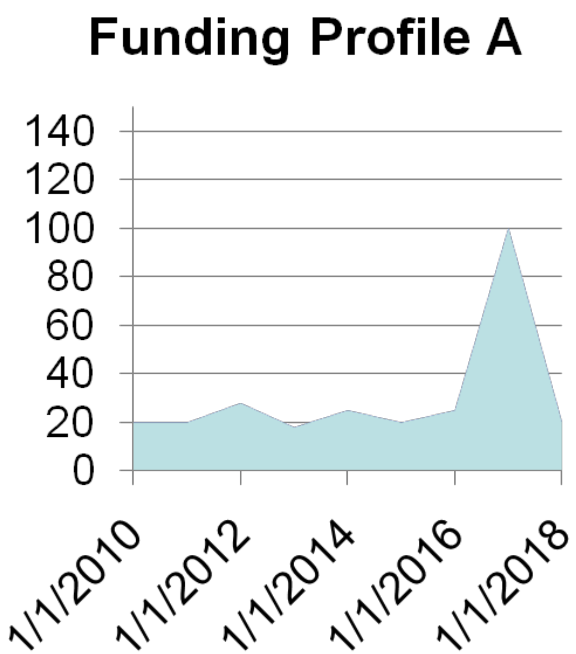
# Summary Relationship Attributes

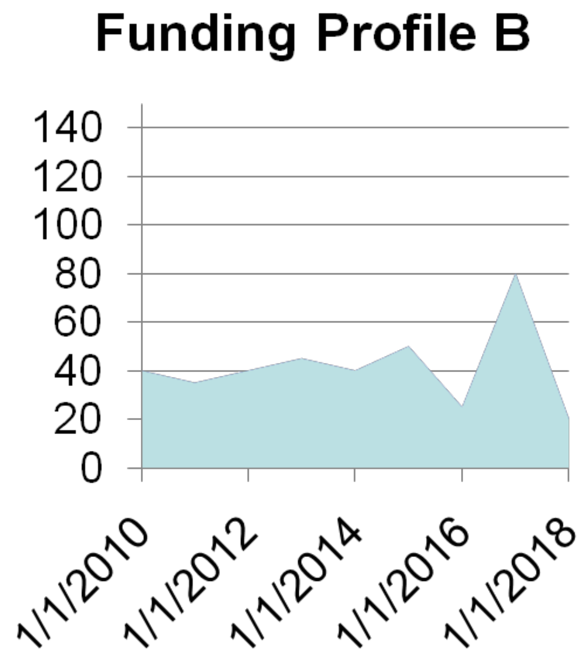
| Model | Obligation | Expectation | Benefit/ Reward | Risk | IP | Metric | Example | Scope, Schedule, Cost |
|---|---|---|---|---|---|---|---|---|
| Funded Investigation | deliver insight, knowledge and opinion | Good work | Information that informs future direction | None | Flexible | Publication, Peer Review | SciDAC, UK eScience | Flexible, Fixed, Fixed |
| Fully Defined Purchase | A solution | Solution works completely | Profit to solution provider, simplicity to purchaser | All solution provider | All solution provider | PERCU | DODmod TI Hector | All Fixed |
| Design and Develop – no solution | prototypes, subsystems, demonstrations | Demonstrated progress toward a solution | Early technology to use | Shared but limited | Flexible – within all performing organizations | Demos, models, etc. | DARPA HPCS | Flexible Scope, Fixed Cost and Sched |
| Co-Design and Co-Development of solution | A solution | Solution working solutions | Early Solution, Future Tech use | Shared by all | Flexible | Working solution | Red Storm, HPSS, Earth Simulator (?) | All Flexible |
| Base plus value add | Min system + additions | Min solution works, at least some of the value add works | Some profit, at least min solution | Min for performing organization, shared | Base for solution provider, other flexible | Minimal working solution, number of value adds | Blue Waters | Fixed Min scope, flexible value added, fixed cost |

## Provider Collaboration Priorities/Models (4)

- Traditional hierarchy
  - Applications, libraries, runtime systems, system software
- Current horizontal software process (white paper)
  - Potentially not extensible to exascale
- Vertical integration/collaboration process
  - Software funding and collaboration implications
  - Open source community leverage
  - Provider staff continuity/training for collaboration
- Engage the broader software provider community
- Lowest software levels most difficult for collaboration
  - Hardware and vendor-specific issues

# Funding Profiles Shape Collaborations (3)




$$\int_0^t f(t)dt = F(t)$$

# Products from Heroes? (3)

- Look at characteristics of software requirements of Top 500:N~500 compared to when same performance was required for N~1
- In other words, is there a reasonable expectation, based on history, that the software requirements of *product* Exascale systems will be similar to *hero* Exascale systems?
  - … and thus can the investments made by HPC product developers/providers and funding bodies can expect an ROI beyond just the first few Exascale systems?
- Example 1 (spot case, need to look at more examples)
  - N=400-500 (09June): ~ 20 TF HPL from ~ 20-50 TF peak; ~ 2-5K cores
    - $O(10^3)$ MPP with multiple cores/CPUs per node
  - N=1 (01June): 7 TF from 12 peak; 8K cores
    - $O(10^3)$ MPP with multiple cores/CPUs per node
  - N=1 (02June): 36 TF from 41 peak; 5K cores
    - $O(10^3)$ MPP with multiple cores/CPUs per node

# Energy Efficient Infrastructure (2)

- Lessons from industrial experience
  - Balance of system versus software focus (IESP)
  - Holistic system design

- Decoupling exascale elements
  - Facilities, hardware, operations
  - Enabling software, applications

# Roadmap/Milestone

| | 2009 | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 |
|---|---|---|---|---|---|---|---|---|
| Software/ Language Issues | | | | | | | | |
| Sustainability | | | | | | | | |
| Collaborative workshops | | | | | | | | |
| Coordinated research | | | | | | | | |
| Educational activities | | | | | | | | |
| Standards activities | | | | | | | | |
| Priorities | | | | | | | | |
| Staffing | | | | | | | | |

- Coordination mechanisms
- Research and development topics

# Background

## Background and Overview

- Experiences and challenges
  - Insights from vendor and center experience
- Technology implications
  - Evolution/revolution
- Industry-community coordination
  - Crosscutting and complementary efforts
- Collaboration scenarios
  - Precompetitive and competitive
  - Economic and political feasibility

# Petascale Lessons for Exascale

- Programs
- Process
- Mechanisms
- Outcomes

- Good
- Bad
- Ugly


Would we do it the same way again?

# We Applied The Fundamental Axiom of Computing …

- … All problems can be solved via another level of indirection
- Which is to say, we discussed how to collaborate and how to create roadmaps

# Exemplar Technical Issues Affecting Software

- Parallelism scale
- Component heterogeneity
- Communication
  - bandwidth/latency
- Memory models
- Storage system structure
- Component reliability
- Energy management

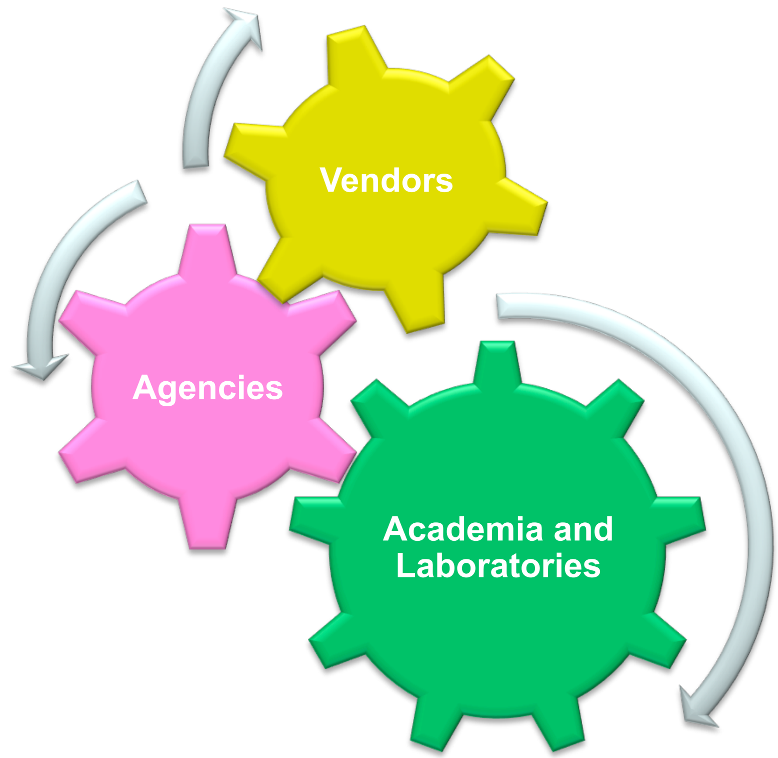
- Design options
  - Evolutionary
  - Revolutionary

- Baseline identification
  - Strengths/weaknesses
  - Available resources

- DARPA architecture reference
  - Evolutionary strawmen
    - “Heavyweight” strawman”
      - Commodity-derived microprocessors
    - “Lightweight” strawman”
      - Custom microprocessors
  - Aggressive strawman
    - “Clean Sheet of Paper” silicon

# Interaction Modalities and Motivations

- Commercial provider issues
  - Profit
  - Differentiation
  - Market share and sweet spots
  - Customer loyalty
  - Interoperability
  - Continuity/sustainability




- Type-specific issues
  - ISVs, software, hardware, integrators
- Commercial and open source software

# Hypothesis: Timelines and Processes Really Matter

- Procurement-driven research and development
  - Rewards incrementalism and product evolution
  - Punishes revolutionary innovation
  - But, it is our historical model
- Short timelines reduce collaboration
  - Create vendor competitive pressures
  - Lessen information sharing
- Competitive advantage and compatibility
  - Differentiation and interoperability
- Implications
  - Define strategic, not tactical roadmaps
  - Enable pre-competitive industry collaborations

# Vendor Exascale Software Roadmap

- The roadmap should
  - Specify ways to re-invigorate the international computational science software community
  - Include computational science software activities across industry, government & academia
  - Be created and maintained via an open process that involves broad input
  - Identify quantitative and measurable milestones and timelines
  - Be evaluated and revised as needed at prescribed intervals
  - Specify opportunities for cross-fertilization of activities, successes and challenges
- Agency strategies for computational science
  - Shaped in response to the roadmap
- Strategic plans
  - Recognize and address roadmap priorities and funding requirements.

# Issues

- Funding level needs/haves
  - Hardware, system software, user software
- Vendor and resource provider needs
  - Testing/development on current systems
- Societal benefits

# Relationship Models

- 1) Funded Investigation (research)
  - Funding to organizations to explore problems to understand issues, explore solution spaces and better define the problems
  - Performing organizations have obligation to deliver insight, knowledge and opinion
  - Expectation
    - People do good work
  - Metrics
    - Peer Review, Publication,
  - Rewards
    - Information that informs future direction
    - IP/royalties
  - IP Ownership
    - Flexible – depends on approach
  - Flexibility in scope – schedule and cost fixed
  - No risk
  - Examples
    - SciDAC
    - Phase 1 of HPCS

# Relationship Models

- 2) Fully defined purchase
  - Funding to industry deliver a solution, possibly a product
  - Performing vendor delivers a solution that works according to a specific set of requirements for a certain cost
  - Expectations
    - A completely working system
  - Metrics
    - Specified performance, RAS, …
  - Rewards
    - Profit to vendors, low risk to funding organizations
  - IP Ownership
    - All industrial partner
  - Fixed scope, schedule and cost
  - Most Risk on industry partner
  - Examples
    - DODmod TI sequence
    - Hector??

# Relationship Models

- 3) Design and Development – no solution
  - Funding to organizations to develop certain technologies and methods that are necessary for the program
  - Performing organizations have obligation to deliver prototypes, subsystems, demonstrations
  - Expectations
    - Demonstratable progress toward a solution
  - Metrics
    - Demonstrations, analytical models, …
  - Rewards
    - Early technology to move to product/use
  - IP Ownership
    - Flexible – Within some performing organizations
  - Fixed schedule, cost – flexibility in scope
  - Risk shared – but limited
  - Examples
    - Darpa HPCS

# Relationship Models

- 4) Co-Design and Co-Development of a Solution
  - Funding to organizations to develop a solution for a set of requirements
    - Roles and Responsibilities flexible in a long term project
  - Performing organizations have obligation to deliver a working solutions
  - Expectation
    - A solution that is probably usable
  - Metrics
    - Working solution
  - Rewards
    - Early solution
    - Shared IP
  - IP Ownership
    - Shared by performing organizations
  - Flexibility in scope, schedule, cost
  - Risk completely shared
  - Examples
    - Red Storm
    - HPSS

# Relationship Models

- 5) Base plus Value Added
  - Funding to organizations to develop a certain base solution that meets a limited set of requirements.
  - Performing organizations have obligation to deliver base system, and to collaborate on value added attributes that will expand the impact of the solution
  - Expectations
    - Solution that meets basic need – and exceed basic need in some areas
  - Metrics
    - Base requirements met, value added in some areas succeed
  - Rewards
    - Science community gets a minimally working solution
    - Industry partners get a certain profit
    - Science community may get a much better system
  - IP Ownership
    - Industrial partner  with maybe some sharing
  - Fixed minimum scope, Larger potential scope, fixed schedule, cost may be fixed or flexible
  - Risk shared but not entirely
  - Examples
    - Blue Waters

## Another Topic – The Vertical Approach

- Resilience (reliability & fault tolerance)
- Performance
- Programmability
- Computational model
- I/O
- Consistency and verification
- Resource Management
- Power Management/Total Cost of Ownership