DARPA HPCS Program

FASTOS Recompete

Petascale Tools Workshop

Fred Johnson
08/14/2007

DARPA HPCS Program and the DOE Office of Science

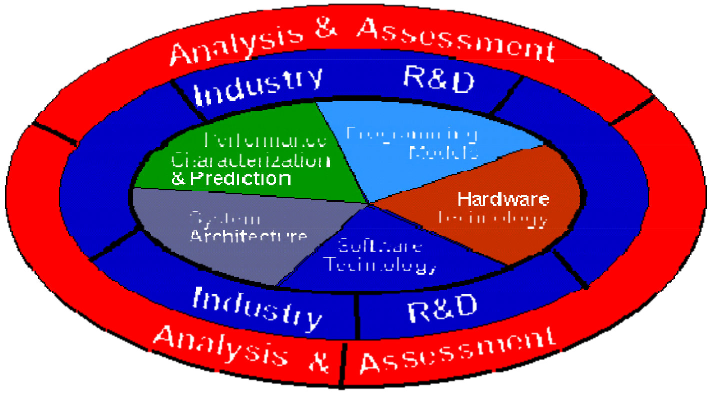
High Productivity Computing Systems

Goal:

- **Provide a new generation of economically viable high productivity computing systems for the national security and industrial user community (2010)**

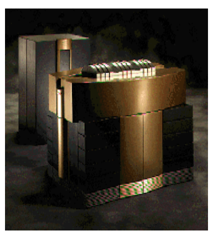
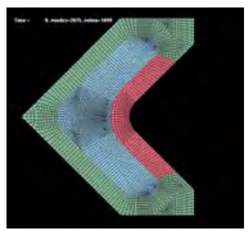
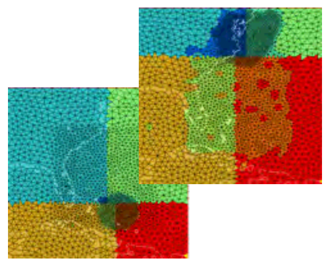
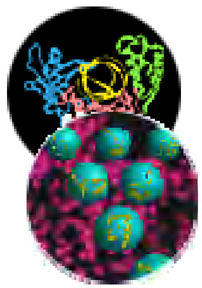
Impact:

- **Performance** (time-to-solution): speedup critical national security applications by a factor of 10X to 40X
- **Programmability** (idea-to-first-solution): reduce cost and time of developing application solutions
- **Portability** (transparency): insulate research and operational application software from system
- **Robustness** (reliability): apply all known techniques to **protect against outside attacks**, hardware faults, & programming errors



HPCS Program Focus Areas



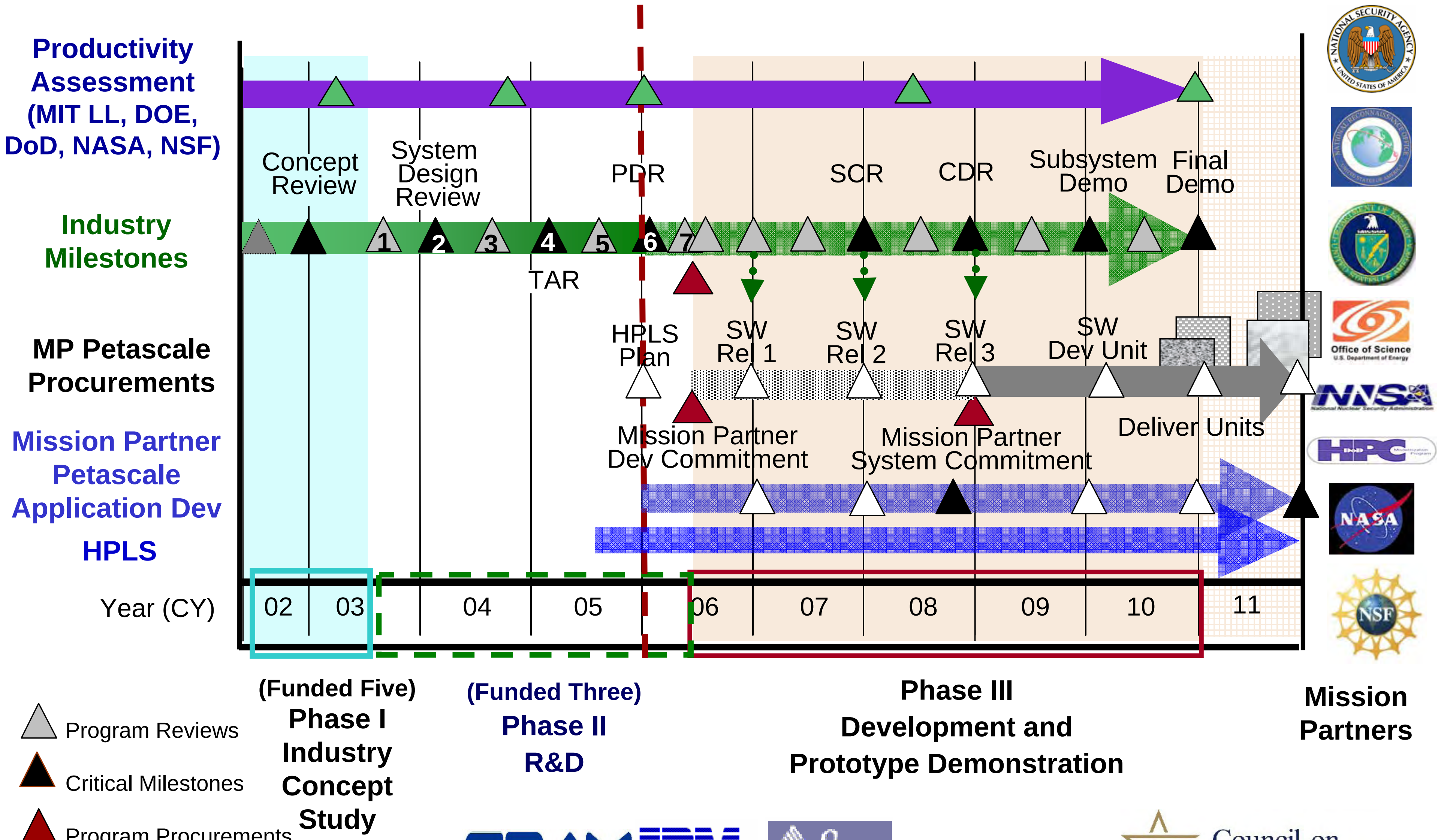
Applications:

- Intelligence/surveillance, reconnaissance, cryptanalysis, weapons analysis, airborne contaminant modeling and biotechnology

Fill the Critical Technology and Capability Gap

Today (late 80's HPC technology)…..*to*…..Future (Quantum/Bio Computing)

HPCS Program Phases I - III



ASCR and HPCS

- Involved from Phase I, day 1
  - Proposal review team
  - Progress review team
- Phase II
  - Proposal, Progress review teams
  - Mission partner
  - Support for Execution time productivity (Bob Lucas, USC/ISI
  - Support for Development time productivity (Jeremy Kepner, MIT/LL)
  - **For further info and HPC Challenge results:**
    - **http://www.highproductivity.org/**
  - Next Generation Programming models—Cray/Chapel, IBM/X10, SUN/Fortress (Rusty Lusk, ANL)
- Phase III
  - Four year $13M/year budget commitment
  - Focus as much as possible on system software ecosystem, details TBD
  - Petascale application development
- Post HPCS
  - Competitive procurement(s)

HPCS Status

- New Program Manager, Charlie Holland

- Phase III Vendors Selected
  - Cray, IBM
  - 90% of Cray funding to come from HPCS Mission Partners (NSA, DOE/SC, NNSA)
  - No separate productivity funding

- FY07/FY08 Senate Appropriations Issue

FASTOS Recompete

Operating and Runtime Systems for Extreme Scale Scientific Computation

- Dates/Participation
  - Announcement posted 3/7
  - Preproposals due 4/6; 58 received, 34 encouraged
  - Final proposals due 6/11; 30 received and reviewed
  - Panel review 7/11-12

- Topics
  - OS framework/kernel/virtualization — 11
  - Input/Output/Storage — 6
  - Fault tolerance/RAS — 6
  - Programming model runtime — 3
  - Misc — 4

Operating and Runtime Systems for Extreme Scale Scientific Computation

- Panel Reviewers
  - Govt (NSF, NSA, DOE, DOD, NASA)-12, Univ/Lab-3, Industry-3
- Status
  - Awards to be made in FY2008
  - Will be funding limited

Software Development Tools for Petascale Computing Workshop — Washington, DC, August 1-2, 2007

The times they are a changin [Bob Dylan 1964]

Background Details

- Community input to planning process for FY08 CS budget increase
- Joint with NNSA/Thuc Hoang
- Steering committee members—Jeff Vetter, ORNL, Bronis de Supinski, LLNL, and Bart Miller, U Wisc
- 55 attendees (invitation only)
  - DOE Lab 25
  - University 10
  - Govt 12
  - Industry 6
  - Europe 2

Workshop Agenda

- Talks by petascale application developers
  - Brian Pudliner, LLNL; Robert Harrison, ORNL; John Daly, LANL
- Platform talks
  - Bob Meisner, Fred Johnson
- Tools issue/challenge overview
  - Bart Miller– We've been here before, but …
- Poster Session

Workshop Agenda

- Breakout Sessions/co-chairs
  - Performance Tools: Dan Reed, RENCI and Bernd Mohr, Juelich
  - Correctness Tools: Susan Coghlan, ANL and Curtis Janssen, SNL
  - Scalable Tool Infrastructure: Jeff Hollingsworth, U Maryland and Al Geist, ORNL
  - Development Environments: Craig Rasmussen, LANL and Rod Oldehoeft, Krell
- Final report with findings and recommendations due 8/31
- More details: www.csm.ornl.gov/workshops/Petascale07

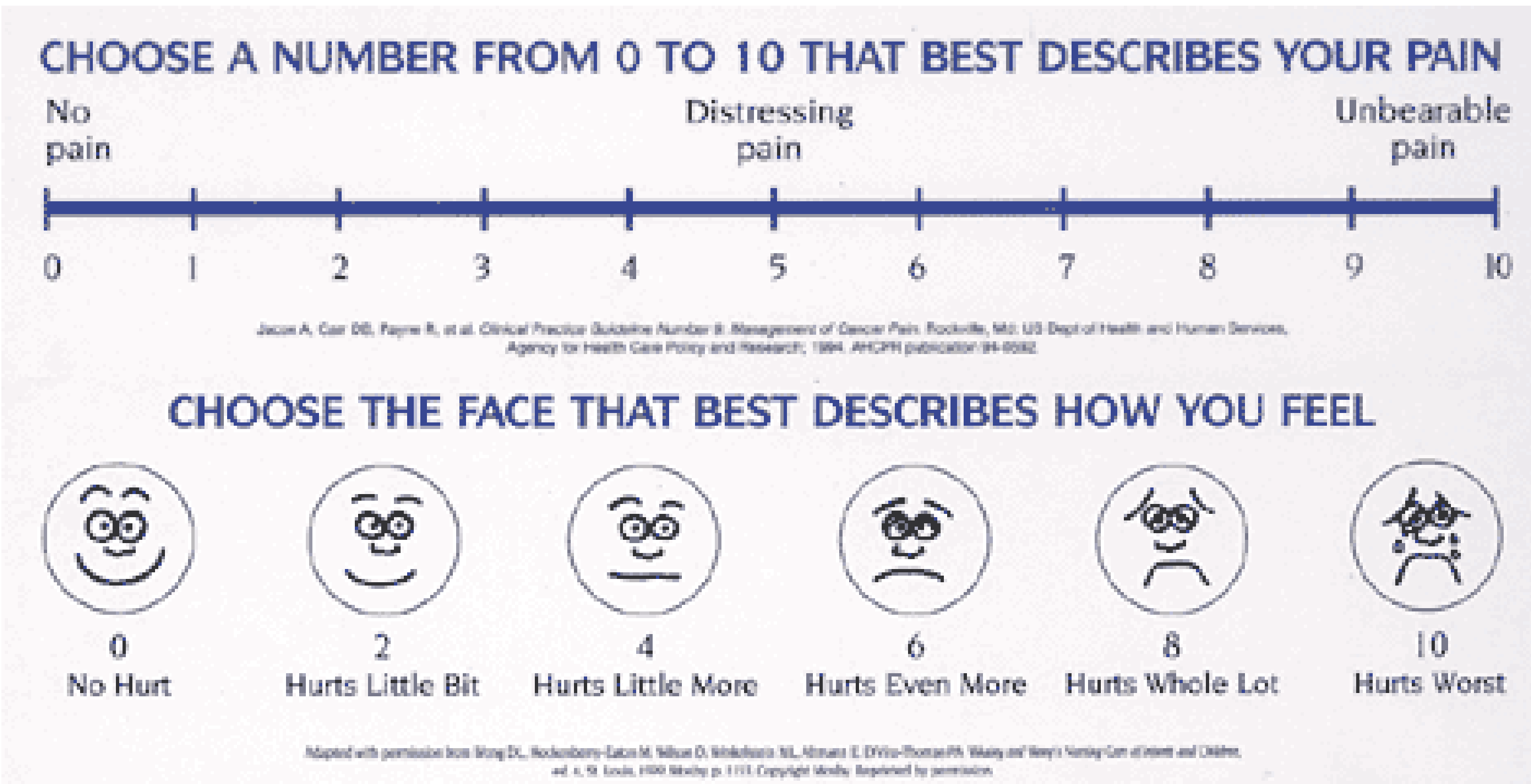
Application Developer Quotes

- John Daly, LANL (41 M hours in Jan-April)
  - Science codes always in development
  - Today's wish list
    - Lightweight massively parallel debugging
    - Low overhead memory debugging
  - Tomorrow's wish list
    - Resource aware job scheduling and task migration
    - Runtime protection against data corruption

- Brian Pudliner, LLNL
  - Most codes rely on timing systems built on top of TAU/PAPI
  - Totalview doesn't cut it at terascale, don't expect to see it at petascale
  - Some codes have 30+ 3rd party libraries
    - Autoconf inline tests drive us nuts when cross compiling
  - Top needs
    - Debugging at scale, memory debugging, memory use characterization
    - Thread correctness, topology characterization/optimization
    - Seriel, parallel, thread performance analysis tools

On Performance Tools …

Sequential **Terascale** **Trans-petascale**

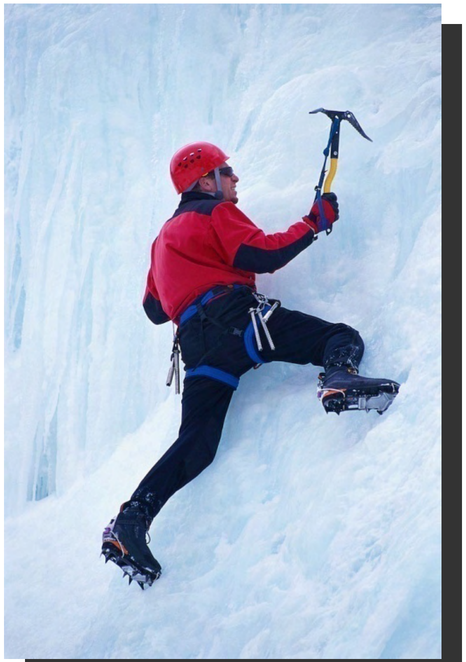


Performance Tool Ecosystems

Petascale Requirements

- Increased automation
  - anomaly detection
  - correlation and clustering
  - data reduction
- Abstraction support
  - detail/complexity hiding
- Runtime adaptation
  - task topologies, …
- Heterogeneity
  - programming models: explicit *and* implicit
  - hardware

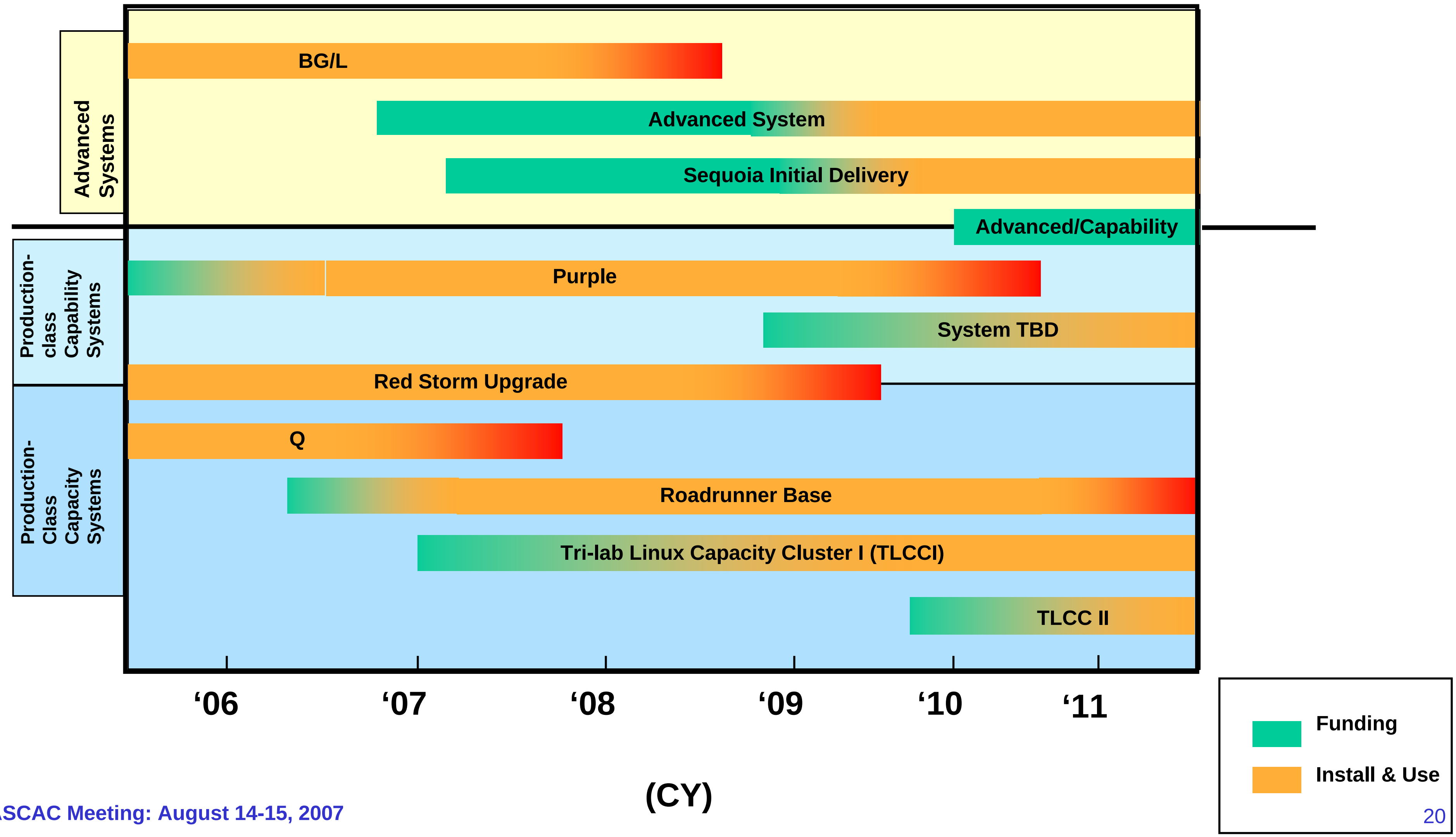
Findings



- **Petascale is *not* terascale scaled up**
  - higher complexity, heterogeneity
- Tool Infrastructure reuse is uncommon, stove piped tools make this hard
- At "peta-scale" tools must handle:
  - 100k cores soon (and up to 1M cores in the future).
  - 2 GB executable & large number of dll's.
  - Support multiple architectures in a single node

Backup Slides

ASC Platform Strategy



ASCR Future Computing Facility Upgrades

- ALCF
  - 100 teraflop IBM Blue Gene/P delivered by end of FY 2007
  - 250-500 teraflop upgrade to IBM Blue Gene/P in late 2008
- LCF – Oak Ridge
  - Cray XT4 upgraded to 250 TF by end of 2007
  - 1 Petaflop Cray Baker system to be delivered by end of 2008
- NERSC
  - 100+ teraflop Cray XT4 in operation by October 2007

Petascale Requirements

- Fault tolerance/resilience
- Education and training
- Multi-level instrumentation
- Memory and I/O analysis
- Performability
  - hybrid/integrated performance and reliability
- Presentation and insight
  - scalable visualization
- Performance modeling and prediction
- Scaling of known methods and techniques
  - million-way parallelism and beyond

Findings

- Crucial interactions
  - users/staff/developers critical
  - education and training
  - feedback
- Insufficient integration
  - among tools
  - component reuse
- No general pathway for release quality tools
  - hardening, documentation, training, support, …

Findings

- Tool Infrastructure reuse is uncommon, stove piped tools make this hard
- At “peta-scale” tools must handle:
  - 100k cores soon (and up to 1M cores in the future).
  - 2 GB executable & large number of dll’s.
  - Support multiple architectures in a single node.
- Applications & systems will be more dynamically adaptable, and tools will need to handle this
- Tools need communication abstractions beyond TCP/IP sockets.
- The costs of supporting tools for multiple platforms and operating systems is straining tool developers
- Going to petascale will increase the need for anomaly detection and (dynamic) data reduction