

SDTPC Workshop

August 1-2, 2007

Washington, DC

Development Environment Infrastructures Working Group

Craig Rasmussen,

Rod Oldehoeft

The times they are a changin' [Bob Dylan 1964]

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

Working Group Charge

1. Integrated development environments
2. Design tools
3. Remote development support
4. Compiler support for tools and development environments
5. Program transformation and refactoring tools
6. Mixed language support
7. Build and runtime environments: configure, make, libraries
8. Dynamic linking issues

Unaffected by Petascale, or In-Hand

Design tools

Source code control and bug tracking tools

Database services

Policy issues in project management

Not Address Because of Time Constraint

Fault-tolerant software development

Topology-aware development support

Working Group Deliverables

Status: Work-in-Progress or Don't-Know-How

Petascale requirements

Findings: define specific challenge

Recommendations regarding challenge

Priority of addressing challenge: high, med, low

Challenge: technical, funding, policy, training

Impact of challenge: high, medium, low

Probability of challenge: high, medium, low

Analysis: IDEs

Status: WP

Petascale requirements: Large data, control, viz requirements; tools usable in frameworks + standalone; development productivity

Findings: IDEs are not as ubiquitous as CLIs, rarely used in HPC, potential for higher productivity, now projects have their own development mechanisms

Recommendation: Pilot projects should show advantages of use

Priority: 2/4/4

Challenge: Technical, Policy

Impact: 3/4/3

Probability

Analysis: Remote Development

Status: WP

Petascale requirements: Few petascale systems mean more remote development, seamless migration among petascale systems

Findings: All petascale computing is remote, but distance matters.

Recommendation: Shared infrastructure for client-server based approaches should be explored, with improved communication efficiency.

Priority: 7/2/0

Challenge: Technical

Impact: 4/5/0

Probability

Analysis: Compiler Support

Status: WP

Petascale requirements: Users and tools need to know about estimated costs at source level, including IO; users need to understand compilers (in)actions; more program correctness feedback

Findings: HPC compilers address some costs, but not IO costs.

Recommendation: Develop infrastructure for cost models, extensive static analysis; vendor help.

Priority: 7/2/0

Challenge: Technical, Policy

Impact: 5/4/0

Probability

Analysis: Mixed-Language Support

Status: WP

Petascale requirements: Yet more support for mixing with current and new (PGAS, HPCS) higher-level languages

Findings: Increased use of mixed-language applications, tending beyond F/C/C++. Prototypes are built in lower-performance languages.

Recommendation: Tool development for advanced language targets, increased automation. Prototype transformation support.

Priority: 4/4/0

Challenge: Technical

Impact: 5/3/0

Probability

Analysis: Build/Configure Support

Status: WP

Petascale requirements: Need to build apps across multiple systems. Parallel build/configures.

Findings: Multiple compilers, OS's, libraries, versions; no common options, CLIs; and shared library unavailability make huge complexity.

Recommendation: New tools (make is still broken), improved tools, interoperability.

Priority: 9/0/0

Challenge: Technical

Impact: 9/0/0

Probability

Analysis: Linking/Library Issues

Status: WP

Petascale requirements: Scalable dynamic linking, inter-library compatibility, order of library access.

Findings: Library order is difficult and manual; dynamic linking does not scale, and not always available.

Recommendation: Vendors need to supply dynamic linking capability; automated tools for library access.

Priority: 5/1/1

Challenge: Technical

Impact: 4/2/1

Probability

Analysis: Program Transformations

Status: WP

Petascale requirements: Dissimilar architectures need to be used efficiently.

Findings: Cannot afford to rewrite apps @ \$100M each.

Recommendation: Explorations of architecture-dependent transformation tools are needed.

Priority: 1/3/2

Challenge

Impact: 2/2/2

Probability