Mitglied der Helmholtz-Gemeinschaft

JÜLICH FORSCHUNGSZENTRUM

# Synthesis of the IESP Santa Fe Whitepapers


The Whitepaper Authors

Compiled and summarized by

- Bernd  Mohr, Thomas Lippert, JSC
- Jesus Labarta, BSC


IESP INTERNATIONAL EXASCALE SOFTWARE PROJECT

# Overall Topics Overview

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Productivity / Portability | X | X | | X | | | | X | | | | | X | X |
| Programming / Execution Model | X | X | | | X | X | X | X | | | X | X | X | X |
| Runtime | X | | | | | | X | | | | | X | | X |
| Applications | | X | X | | | X | X | | X | | | | X | X |
| Fault resiliency | | | X | | X | | X | | | | | X | | X |
| Libraries / APIs | | X | | | | | | | | | | | | |
| Development Environments | | X | X | | | | | | | | | | | X |
| Power / Complexity | | X | | | | | X | | | | | X | | X |
| Cultural / Organization | | X | | X | X | | | | | X | X | | X | |

# Programming / Execution Model

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Programming / Execution Model | X | X | | | X | | | X | | | | X | X | X |
| Address space | | | | | X | | | X | | | | X | | X |
| Expression of Parallelism | X | | | | | | X | | | | | X | | |
| Tasks | | | | | X | | | | | | | | | X |
| Asynchrony | | X | | | | X | | | | | | | | X |
| Hierarchy / mixed | X | X | | | X | | | X | | | X | | | X |
| Heterogeneity | | | | | | | X | | | | | | | X |

# Development environment / Runtime

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Runtime | | | | | | | | | | | | | | |
| Dynamic resource management | | | | | | | | | | | | X | | X |
| Load balance | X | | | | | | | | | | | | | X |
| Latency hiding / BW minimization | X | | | | | | X | | | | | X | | X |

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Development environments | | | | | | | | | | | | | | |
| Compilers (single core performance) | | X | | | | | | | | | | | | |
| Debugging | | | X | | | | | | | | | | | |
| Performance tools | | X | | | | | | | | | | | | X |

# Applications

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Applications | | | | | | | | | X | | | | | |
| Data models and analysis | | | X | | | | X | | | | | | | |
| Algorithmic developments | | X | | | | X | | | | | | | X | X |
| Porting/tuning effort | | | | | | | | | X | | | | | |
| Benchmarks | | | | | | | | | X | | | | X | |
| Industrial, multi-physics/scale | | X | | | | | X | | X | | | | | |
| Preprocessing: meshing, … | | X | | | | X | | | | | | | | |
| Visualization, post processing | | X | | | | | | | | | | | | |
| Memory wall (volume per core) | | | | | | | X | | | | | | | |

# Cultural / Organization

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Cultural / Organization | | | | | | | | | | | | | X | |
| Info sharing and aggregation | | | | X | | | | | | | | | X | |
| Collaboration | | | | X | X | | | | | | | | | |
| Standardization | | X | | X | | | | | | | | | | |
| Coordinated funding support | | | | | | | | | | X | X | | | |
| Commercialization | | | | | | | | | | | X | | | |
| Training | | | | | | | | | | | | | X | |

# Libraries / Portability / Power

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Libraries, APIs | | X | | | | | | | | | | | | |
| Code coupling | | X | | | | | | | | | | | | |

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Productivity / portability / ease of use | X | X | | X | | | | | | | | | X | |
| Migration path | | X | | | | | | X | | | | | | X |

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Power management | | | | | | | X | | | | | X | | X |
| Management complexity | | X | | | | | | | | | | | | |

1

# The Application Perspective: Seeking Productivity and Performance

**David Barkai**

**Intel Corporation**

- The programming model for exascale computing needs to address both productivity and performance
- Both are helped by a model that maps the application design to cluster architecture with multi multicore processor nodes
- This may be achieved via a hierarchical MPI-based model
  - Layered by  inter-node, intra-node, intra-socket
  - Exascale system also divided in groups or “gangs” of processes (Gropp et al) – only ‘flat’ component
- Companion topic: much performance is left untapped.
  - Focused effort on defining “computational operators” and  optimizing hardware and software to support them may result in huge ROI
  - Methods for expressing parallelism need to comprehend such operators

2

# EDF White Paper

## J.Y. Berthou, J.F. Hamelin, EDF R&D

**Challenge:** addressing multi-physics, multi-scale simulations on massively parallel heterogeneous architectures, combining parallel software components developed independently from each other and dealing with constantly evolving legacy codes

**Priority research themes:**

- Programming massively parallel computers : languages/compilers/performance analysis tools for achieving mono-processor high performance, fault tolerant implementation of Libraries/Languages for mixed parallelism (MPI/OpenMP/"cuda like" language), algorithm/solvers and data structures adapted to heterogeneous/hybrid, multilevel and hierarchical massively parallel machines
- Unified generic interface for High Performance Solvers
- Unified stochastic HPC computing tools and methods for uncertainty and risk quantification
- Unified Simulation Framework and associated services adapted to massively parallel simulation:
  - Common data model and associated libraries for mesh and field exchange that enable interoperability and the coupling of independent parallel scientific software
  - Meshing tools: parallel meshing, mesh healing, CAD healing for meshing, dynamic mesh refinement
  - Parallel visualization, remote and collaborative post-treatment tools
  - Supervising and code coupling tool for tightly coupling schemes

# 3 Software and Exascale Computing

∑ BM

**Bill Camp**
**Intel Corporation**

- Major software issue in developing a robust exascale computational economy: **scalability**

- Comes in numerous disguises
  - **Programmability, debug-ability, optimization** for $10^9$ thread
  - **Interpretability**: exascale apps will produce yoddabytes data
  - Will need integrated (complex) HW and SW **reliability**
  - **Performance**: load imbalance, aggressive overlap of communication and computation, …
  - **Energy cost of software**

4

# A Proposal for a Capability Centers Consortium

**Bill Gropp, Mark Snir**
**NCSA and UIUC**

- Addressing challenges in HPC software and systems requires collective effort
  - High-end HPC users have specific and demanding needs from software, including (trans)portability between high-end systems
  - Increasing user productivity will require collaborative efforts in software
  - Activities proposed include
    - Information sharing and aggregation
    - Collaborations
    - Standardization

5

# Software Challenges of Extreme Scale Computing

**Michael Heroux**
**Sandia National Laboratories**

- Parallel Programming Transformation:
  - New layer: Manycore, shared memory, underneath MPI.
  - No standard, portable manycore API.
  - Preparation: All computation in stateless functions.
- Beyond the “Forward Problem”:
  - Make current apps into subroutines.
  - 100-1000x increase in parallelism.
- Fault-resilient Application Environment:
  - Vertically integrated solution, application in charge.
- Hierarchical, Multi-organization Software Engineering:
  - Collaborative effort: physically and culturally diverse orgs.

6

# PDE-based Applications and Solvers at Extreme Scale

∑ BM

**David Keyes**
**Columbia University and SciDAC TOPS Project**

- The availability of high capability architecture makes algorithms more, not less, important
  - All algorithms can only be log-linear at worst!
  - Needs synchronization-free (or -minimized) algorithms
  - Needs performance analysis beyond flops
- Scalability needed beyond solvers
  - Tools to manage meshes, fields, particles, …
  - Probably needs new generation of codes!

7

# Major CS Challenges at Exascale

**Al Geist, Paul Messina, & Robert Lucas**
**ORNL, ANL, & USC/ISI**

Numerous studies in last two years have examined Exascale

- **Three DARPA studies and many DOE meetings**
- **Their findings are remarkably consistent**

The major CS challenges to achieving Exascale:

- **Accelerating growth in concurrency** (finding billion-way parallelism)
- **Memory wall** (latency, BW, and volume of memory seen by a core)
- **Continuous faults, often silent** (requires a paradigm shift in SW)
- **Heterogeneity** (programming nodes with different types of cores)
- **Knowledge discovery** (data volumes, multiple formats and types)
- **Increasing complex application software** (multi-scale multi-physics)
- **Power management** (power is a limiting factor to reach Exascale)

# 8 Slouching Towards Exascale

Rusty Lusk
Argonne National Laboratory

## The programming model problem:

- Where we are now
  - MPI provides a robust, portable, effective standard for communication among separate address spaces
  - We require, but do not have, a similarly effective standard for expressing parallelism within an address space that interacts in a defined way with MPI
  - A hybrid approach (MPI + X for some X) is most likely to provide a migration path for applications to the million-core regime
- What we need to do
  - Eschew ritualized denigration of MPI
  - Recognize the need for a shared-memory programming model
  - Understand the difference between end applications and libraries
  - Don't abandon the HPCS language ideas

9

# Application Analysis and Porting in the PRACE Project

Peter Michielse
Netherlands National Computing Facilities Foundation (NCF)

- **Foundation of initial PRACE Application Benchmark Suite (PABS) in spring 2008**
  - Applications with broad usage in Europe
  - Coverage of main scientific areas that require HPC
  - Scalability potential
  - Foundation for PRACE optimisation and petascaling tasks
  - Horizontal approach: one person responsible for porting, optimisation and petascaling (with collaborators)
- **Review of PABS in spring 2009:**
  - Focus on scalability and licensing issues
  - Adapt initial PABS accordingly

10

# NSF IESP Whitepaper

∑ BM

**Abani Patra, Rob Pennington, Ed Seidel**
**Office of Cyberinfrastructure, National Science Foundation**

- NSF 's Vision for Cyber infrastructure for $21^{st}$ Century Discovery
  - Comprehensive program for supporting the national cyber infrastructure for science and engineering
  - NSF wants to pursue global partnerships with other organizations and agencies
- Exascale challenges will drive innovation in may areas
  - Integration between software, applications, hardware
- Whitepaper outlines inputs desired by NSF in order to understand requirements and to define details of this program

11

# A Collaboration and Commercialization Model for Exascale Software Research

**Mark Seager, Brent Gorda**
**Lawrence Livermore National Laboratory**

- We recommend a coordinated strategy between Research & Development (R&D), Development & Engineering (D&E) and Productization and Service (P&S). With migration path towards commercialization.
- Keep current focus areas and funding agents for R&D, D&E and P&S as they currently are and add stake holders from next stage in the process.
- Keep the model flexible as possible to encourage development and competition.
- Multiple iterations required to get to exascale.

11

# The Case for A Hierarchal System Model for Linux Clusters

**Mark Seager, Brent Gorda**
**Lawrence Livermore National Laboratory**

- HPC pyramid investment model requires we pull up the rest of the pyramid while pushing to exascale or the model breaks down.
- Hierarchal systems model developed for petascale systems is a good starting point, with possibly more than one level in the hierarchy, for exascale systems research
- The current “Flat” Linux cluster systems model can be turned into a hierarchal systems model and scale up to 10K to 100K nodes.
- A change to both hardware (simpler compute nodes) and software are required.
- We can mine existing petascale systems efforts and combine it with readily available commercialization paths.

12

# The Biggest Need: A New Model of Computation

Thomas Sterling
Louisiana State University

- HPC is in a phase change (VI$^{th}$)
  - Exascale heterogeneous multicore systems
  - Power & reliability constrained with billion-way parallelism
- Model of Parallel Computation – What is it?
  - Supports co-design of all system layers
  - Governing principles guiding management, naming, control
- Towards a New Exascale Model of Computation
  - Work-queue dynamic allocation of cores for high utilization
  - Message-driven for system wide latency hiding & limiting
  - Active Global Address Space for Programmability
  - Lightweight object oriented synchronization for low overhead
  - Threads as 1$^{st}$-class objects for dynamic adaptive scheduling

13

# Developing a High Performance Computing/Numerical Analysis Roadmap

JÜLICH FORSCHUNGSZENTRUM

**Anne Trefethen, Nick Higham, Ian Duff, and Peter Coveney**
**Universities of Oxford, Manchester, London & Rutherford Lab**

Leveraging work in the US and Europe together with UK specific workshops and discussions groups have lead to barriers for software development that fall into five themes

1. Cultural Issues
   - some people won't share with, or trust, others...
2. Applications and Algorithms
   - Need to bring application and algorithm development closer
   - Need new algorithms for new architectures
3. Software Challenges
   - Engineering, portability, programming models, .....
4. Sustainability
   - Need better models for sustainability not only for UK efforts but those we depend on!
5. Knowledge base
   - It would be good to know who is doing what and where
   - We need to train more people with this cross cutting set of skills.

**http://www.oerc.ox.ac.uk/research/hpc-na**

# BSC Vision Towards Exascale

**Jesus Labarta, Eduard Ayguade, Mateo Valero**
**BSC**

- Programming model: the key component
  - Decoupling: algorithms from systems, naming from containers, writing from execution
  - Global address space, task based: dependences and data access information
  - Heterogeneity support and hierarchically scalable: from node to cluster level, tunable granularity, asynchrony
  - Intelligence/responsibility to the runtime: dynamic, malleable; load balancing; architecture optimized; locality, latency, reuse and bandwidth aware
- Algorithmic work, only once restructuring, smooth path
- Tools: down to details, more intelligence, provide insight
- BSC contrib. : CellSs/SMPSs, Paraver/Dimemas, http://www.bsc.es

# Note: New Whitepapers for IESP Paris

- **An Exascale Approach to Software and Hardware Design**
  William Kramer (NCSA) and David Skinner (LBNL)
- **Consistent Application Performance at Exascale**
  William Kramer (NCSA) and David Skinner (LBNL)
- **Performance at Exascale**
  Bernd Mohr (JSC) and Matthias S. Mueller , Wolfgang E. Nagel (ZIH/TUD)
- **Resource Management**, Barney McCabe (ORNL) and Hugo Falter (ParTec)
- **Programmability Issues**, Vivek Sarkar (Rice U.), Jesus Labarta (UPC), Mitsuhisa Sato (U. of Tsukuba), Barbara Chapman (U. of Houston)
- **Models of Computation – Enabling Exascale**, Thomas Sterling (LSU)
- **Major Computer Science Challenges at Exascale**,
  Al Geist (ORNL) and Robert Lucas (ISI)
- **Towards Exascale File I/0**, Yutaka Ishikawa (U. of Tokyo)
- **Co-design of Architectures and Algorithms**,
  Al Geist (ORNL) and Sudip Dosanjh (SNL)
- **IESP Exascale Challenge: Resilience and Fault Tolerance**,
  Al Geist (ORNL) and Franck Cappello (INRIA)