Argonne NATIONAL LABORATORY

Is Hybrid Programming a Bad Idea Whose Time Has Come?

Panel Moderator: Pavan Balaji, Argonne National Laboratory

Panelists:

Taisuke Boku, University of Tsukuba

Allen Malony, University of Oregon

Bronis de Supinski, Lawrence Livermore National Laboratory

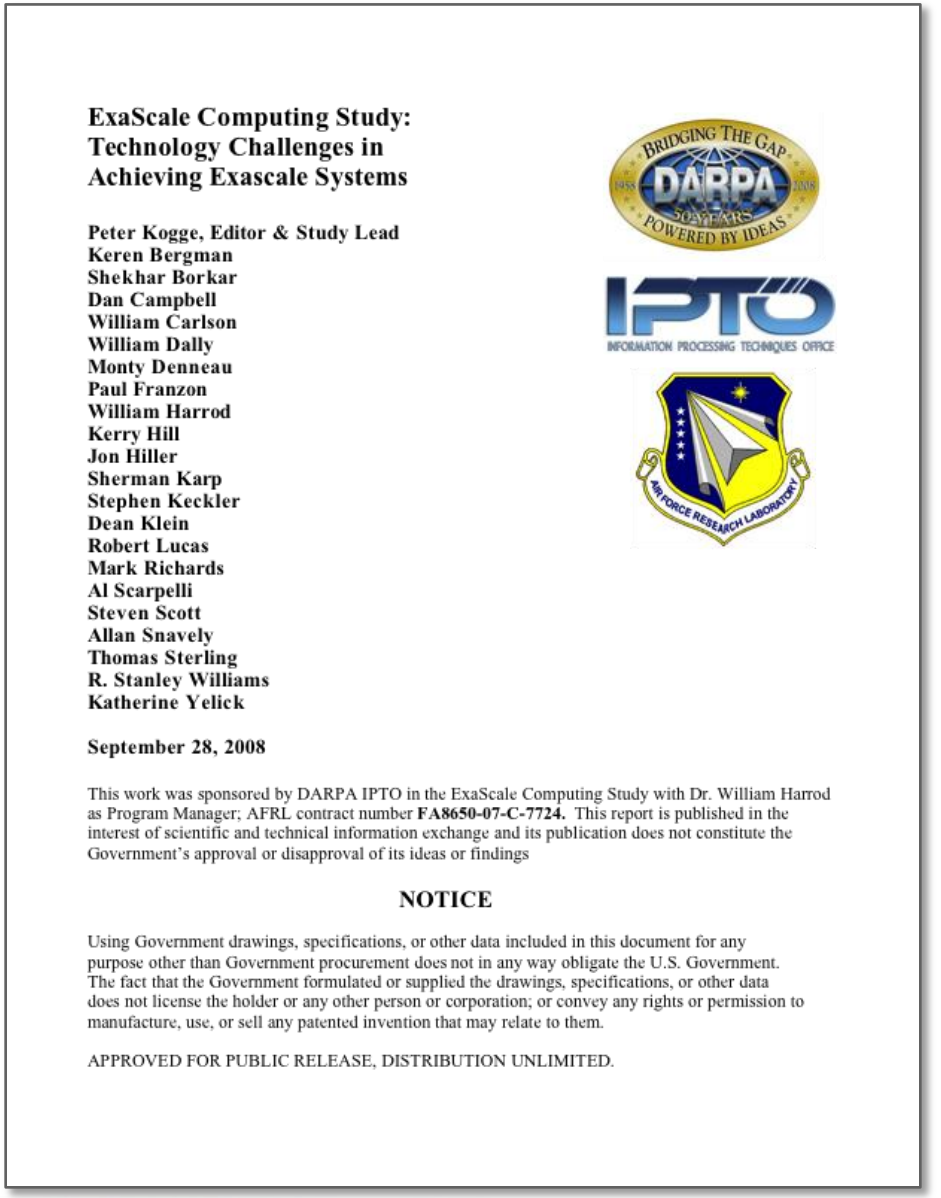
Vinod Tipparaju, Oak Ridge National Laboratory

Vijay Saraswat, IBM Research

U.S. DEPARTMENT OF ENERGY

Architectural Constraints at the Exascale

- Two primary constraints:
  - Upfront cost
  - Maintenance cost
- Power < 100MW (including cooling and storage)
  - 100-1000 times more power efficient that current petascale systems
- Cost and Size
  - 100-1000 racks



ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems

Peter Kogge, Editor & Study Lead
Keren Bergman
Shekhar Borkar
Dan Campbell
William Carlson
William Dally
Monty Denneau
Paul Franzon
William Harrod
Kerry Hill
Jon Hiller
Sherman Karp
Stephen Keckler
Dean Klein
Robert Lucas
Mark Richards
Al Scarpelli
Steven Scott
Allan Snavely
Thomas Sterling
R. Stanley Williams
Katherine Yelick

September 28, 2008

This work was sponsored by DARPA IPTO in the ExaScale Computing Study with Dr. William Harrod as Program Manager; AFRL contract number **FA8650-07-C-7724.** This report is published in the interest of scientific and technical information exchange and its publication does not constitute the Government's approval or disapproval of its ideas or findings

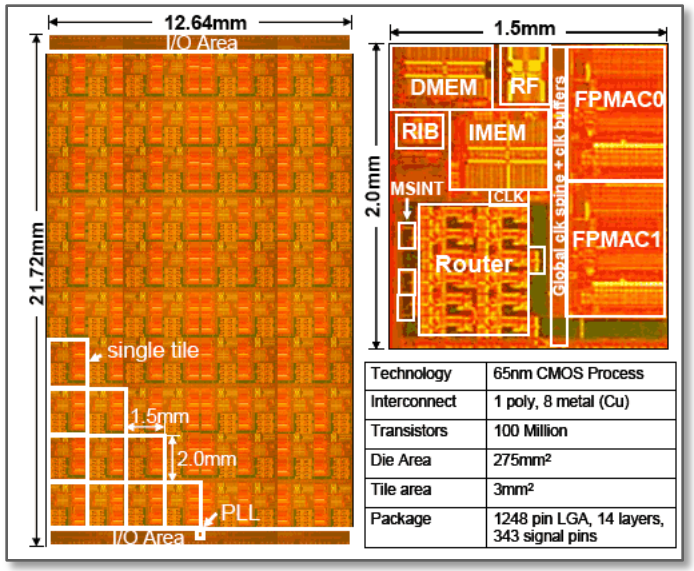
NOTICE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

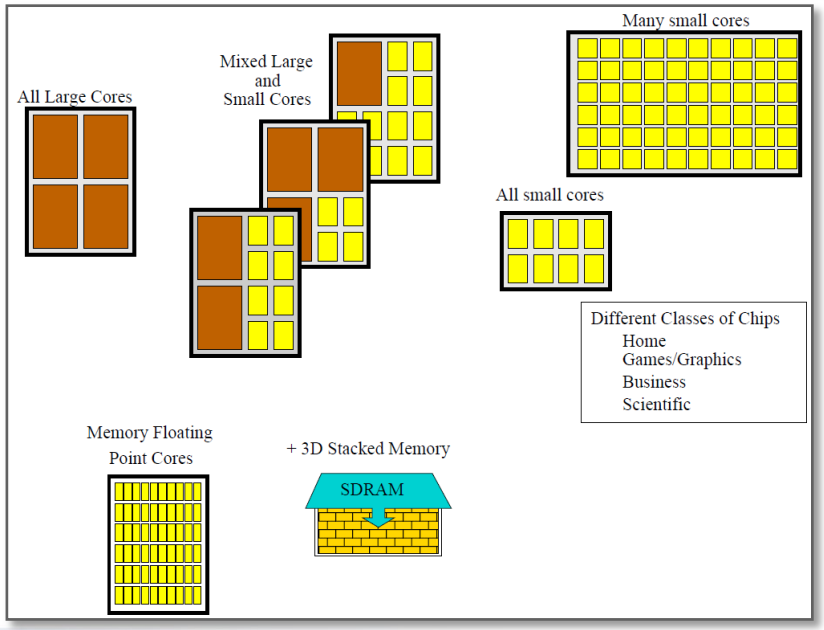
APPROVED FOR PUBLIC RELEASE, DISTRIBUTION UNLIMITED.

Two possible approaches to build Exaflop Systems

- Lots of light-weight cores
  - In-order execution (no speculation hardware)
  - Small caches
  - No hardware cache coherency
- Heterogeneous cores
  - General purpose processors + GPGPUs
  - More tightly integrated heterogeneous processors are coming



Technology	65nm CMOS Process
Interconnect	1 poly, 8 metal (Cu)
Transistors	100 Million
Die Area	275mm²
Tile area	3mm²
Package	1248 pin LGA, 14 layers, 343 signal pins



[Courtesy William Gropp, UIUC]

How do we program these systems?

- Single programming model everywhere
  - Expressing algorithms in a high-level model so that the same application can work on hybrid architectures unchanged
  - Can we make these models rich enough to work on current and future architectures?
- Hybrid programming models
  - Explicitly allow applications to deal with the architectural hybridness
  - Application programs to be modified for nested parallelism
  - Multi-programming model interaction is tough
    - Performance and debugging tools, incompatible runtime systems
  - Performance implications
    - Resource sharing interactions

Questions to Panelists

- While hybrid architectures are upon us, are we equipped to deal with them?
- Specific questions:
  - *Has the time for hybrid programming come?*
    - *Do we need to expose hybrid architectures to applications and let them deal with it?*
    - *Can we design high-level programming models, performance/debugging tools and runtime systems that can effectively abstract such architectural hybridness from the applications?*
  - *Is it a good idea?*
    - *"It is needed" does not necessarily mean "it is a good idea"*
    - *Is this the next step in our natural evolution or is it a terrible but necessary evil?*