Is the Multi-Core Roadmap going to Live Up to its Promises?

**Moderator:** Per Stenstrom, Chalmers

**Panelists:**
- Robert Cypher, Sun Microsystems
- Michel Dubois, Univ. of Southern California
- Tim Mattson, Intel
- Kunle Olukotun, Stanford University
- Greg Pfister, IBM
- Dean Tullsen, University of California, San Diego
Why multi-cores are here

Multiple upcoming walls, simultaneously

• Clock frequency scaling
  – Power consumption
  – Memory latency

• Instruction-level parallelism drained
  => Multiprocessors (multi-cores)
Old concept in a new setting

30 years of research: What did we learn?

– Applications/Algorithms:
  • Successful in some domains; less so in others

– Software:
  • Progress in some areas (OS) less so in others (compilers + languages)

– Architecture
  • Lots of progress – is it enough?
Questions

Q1: Is today a better setting for multiprocessors, or not? If yes, in what sense? If not, why?

Q2: Do we need more research? Then, what are the challenges? If not, why?
Multi-cores: A new software crisis

• Most legacy code is single threaded
• Compilers are of little help
• Parallel programming difficult
  – Educational gap
  – Semantic gap
  – Maybe, a mental gap

Q3: Can sequential programming prevail; if not, how do we provide a smooth path?
Progress was needed, yesterday!

Mismatch between all levels

- Applications
- Languages
- Software
- Architecture

Q4: Would more collaboration across disciplines help progress? Then how? If not, why?