

New perspective on parallel programming

Marco Danelutto
Dept. Computer Science
University of Pisa



*Master Degree (Laurea Magistrale) in
Computer Science and Networking
Academic Year 2009-2010*





- **A View of the Parallel Computing Landscape**
 - *bY Krste AsAnoViC, rAstisLAV boDiK, JAmes DemmeL, tonY KeAVenY, Kurt KeutZer, John KubiAtoWiCZ, neLson morGAn, DAViD PAtterson, KoushiK sen, John WAWrZYneK, DAViD WesseL, AnD KATHERine YeLiCK*
 - *from the same authors of*
 - ***The Landscape of Parallel Computing Research: A View from Berkeley (2006)***



The rationale

- **Industry Needs help from the research community to succeed in its recent dramatic shift to parallel computing. Failure could jeopardize both the IT industry and the portions of the economy that depend on rapidly improving information technology.**



What happened

After crashing into the power wall, architects were forced to find a new paradigm to sustain ever-increasing performance. The industry decided the only viable option was to replace the single power-inefficient processor with many efficient processors on the same chip. The whole microprocessor industry thus declared that its future was in parallel computing, with increasing numbers of processors, or cores, each technology generation every two years. This style of chip was labeled a multicore microprocessor. Hence, **the leap to multicore is not based on a breakthrough in programming or architecture and is actually a retreat from the more difficult task of building power-efficient, high-clock-rate, single-core chips**



Challenges (1)

- **One especially vexing challenge for the parallel software span is that sequential programming accommodates the wide range of skills of today's programmers. Our experience teaching parallelism suggests that not every programmer is able to understand the nitty gritty of concurrent software and parallel hardware; difficult steps include locks, barriers, deadlocks, load balancing, scheduling, and memory consistency. **How can researchers develop technology so all programmers benefit from the parallel revolution?****



Challenges (2)

- **A second challenge is that two critical pieces of system software—compilers and operating systems—have grown large and unwieldy and hence resistant to change. One estimate is that it takes a decade for a new compiler optimization to become part of production compilers. **How can researchers innovate rapidly if compilers and operating systems evolve so glacially?****



Challenges (3)

- A final challenge is **how to measure improvement in parallel programming languages**. The history of these languages largely reflects researchers deciding what they think would be better and then building it for others to try. As humans write programs, we wonder whether human psychology and human-subject experiments shouldn't be allowed to play a larger role in this revolution.



The principles (sw)

- (a) Architecting parallel software with design patterns, not just parallel programming languages**
- (b) Split productivity and efficiency layers, not just a single general-purpose layer**
- (c) Generating code with search-based autotuners, not compilers**
- (d) Synthesis with sketching**



The principles (sw, 2)

- (e) Verification and testing, not one or the other**
- (f) Parallelism for energy efficiency**
- (g) Efficiency**
- (h) Energy amortization**
- (i) Energy savings**
- (j) Space-time partitioning for deconstructed operating systems**



The principles (hw)

- **Supporting OS partitioning**
- **Optional explicit control of the memory hierarchy**
- **Accurate, complete counters of performance and energy**
- **Intuitive performance model**



Reasons for optimism

- **No killer microprocessor**
- **New measures of success**
- **All the wood behind one arrow**
- **Manycore synergy with cloud computing**
- **Vitality of open source software**
- **Single-chip multiprocessors enable innovation**
- **FPGA prototypes shorten the hardware/software cycle**



Berkeley, what else?

- **A strong viewpoint from a strong community**
- **This is 90% in line with SPM perspective**
- **And mainly**
 - *there is no other serious proposal tackling the same problem(s) with the same strength ...*