Principles/theory matter and can matter more: Big lead of PRAM algorithms on prototype-HW

Uzi Vishkin

There is nothing more practical than a good theory-- James C. Maxwell

A. JAMES CLARK SCHOOL *of* ENGINEERING

Mainstream many-cores are in trouble

Future of Computing Performance - Game Over or Next level, National Academy of Engineering 2011

- *"Only heroic programmers can exploit the vast parallelism in today's machines"*
  - Parallel HW is way too difficult to program for faster completion time of a single general-purpose task
  - Yawn factor. Apps innovation over last decade in performance general-purpose desktops low. Compare to internet & mobile
  - SW spiral (business model!) is broken
- Need whole **new parallel computing stack** [➔'generalist researchers']

Will advocate a theory-driven parallel computing stack

Quiz who was the generalist behind the 1st general-purpose serial computer?

Where are all the theorists?

- Many parallel computing theorists in 1980s & early 1990s
- Why recent rise in other communities, but not theory?
- My conjecture:
  - Unstable/Unappetizing programming models
    - Theorists seek robust contributions. Important for business reasons, as well. Don't blame the messenger.
    - Recall also the "yawn factor"
  - Collective memory: Trauma of 1990s belittling/bashing theory.

➔ Theory-driven parallel computing stack

Good news We can build on

$$\frac{\text{Bandwidth2012}}{\text{Bandwidth1980}} \sim= 300$$

$$\frac{\text{Latency1980}}{\text{Latency2012}}$$

Source [HP-12]

Building Confidence
Theory Hypotheses and their Validation

Be guided by theory/principles:

1981 – Introduced Work-Depth methodology [& aspiration that WD will become synonym with *thinking in parallel*] & **2 hypotheses**:

1. **WD methodology will be sufficient for PRAM**

   [Argued: PRAM is too difficult. Common wisdom: too easy]

1980s-90s – PRAM won battle of ideas on parallel algorithms theory. WD used as description framework + validation of WD

2. **PRAM/WD algorithms will be sufficient for** (the right) **stack**

[Skill – generalist]

1998 - XMT@UMD: Negotiated WD/PRAM with HW constraints

2012 – Full prototype-stack (incl. HW) validation of WD/PRAM

Also: validation of generalist skill

Can do orders-of-magnitude better on speedups and ease-of-programming

Best speedups on non-trivial stress tests

	XMT	GPU/CPU	XMT source
BiConn	33X	4X	TarjanV85
TriConn	129X	?	RamachandranV88, MillerR
MaxFlow	108X	2.5X	ShiloachV82 + GoldbergTarjan88

"Normal" algorithm to implementation effort. No algorithmic creativity.

1. XMT enables PhD-level programming at High School, May 2012
2. *"XMT is an essential component of our Parallel Computing courses because it is the one place where we are able to strip away industrial accidents from the student's mind, in terms of programming necessity, and actually build creative algorithms to solve problems"*—national award winning HS teacher
3. 2012 Parallel **Algorithms** course: prog assignment parallel BiConn

Not just talking – See CACM, Jan 2011

Algorithms&Software

WorkDepth

Creativity ends here

PRAM validation completed with Bi&Tri-connectivity & Max-flow results – most advanced algs

Programming & **workflow**

No 'parallel programming' course beyond freshmen

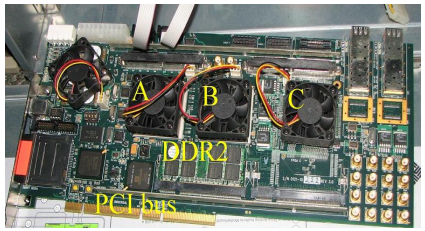
Stable **compiler**

Direct HW 'orchestration'

PRAM-On-Chip HW Prototypes

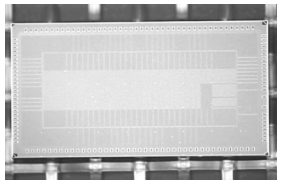
64-core, 75MHz FPGA of XMT

(Explicit Multi-Threaded) architecture

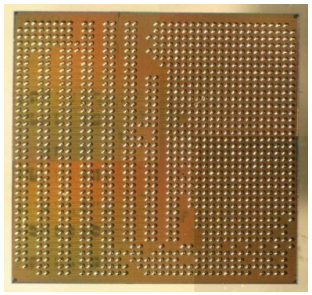


SPAA98..CF08

128-core intercon. network



IBM 90nm: 9mmX5mm, 400 MHz [HotI07]



FPGA design➔ASIC

IBM 90nm: 10mmX10mm

Architecture scales to 1000+ cores on-chip

Research challenge in principles of parallel computing

- Establish that principles/theory actually matter and can once and for all remedy the ills of the field

XMT validated the hypothesis on WD/PRAM algorithms.

- Now, advance XMT into a serious new stack alternative:
  - Applications Establish: XMT can provide order-of-magnitude improvements on interesting scientific/market applications
  - Wall-clock runtime Upgrade current prototype to yield wall-clock speedups ➔ appealing to widely try by others including application people

Need NSF program Enable new stack

Must enable:

- Coherent theme. Prefer: theory- or architecture-driven proposals led by generalists (plus domain experts), rather than 'check-list collection' of such experts
- Development budget, so that research can follow

- Opportunity to test recent concepts; e.g., min primitive
- Separate in budget/features from other programs

Last 3 arguments in favor of a theory-driven stack

Architects:

- Really good at optimizing HW, given code examples. But, what if 1st HW draft is not good enough? Jury out for … 5th decade☺
- Primary future beneficiary from good enough 1st draft - problems they are best at

Biggest (?) problem Diminished competition among HW vendors

- NSF should not limit itself to industry welfare. How about the opposite: compensate for insufficient competition?