Get the Parallelism out of my Cloud

Karu Sankaralingam and Remzi H. Arpaci-Dusseau

University of Wisconsin-Madison
http://www.cs.wisc.edu/vertical
The Good Old Times

horse pulls the cart
or maybe not so good times....
New World?

Multi-core drives bloated Software?
Do (real) developers and users care?
What does it provide to end-users?
Who is the end-user?
Can we exploit some synergistic trends...  
Cloud + Multicore = ?
Cloud + Devices
Cloud + Devices

Users (billions)
Myth vs. Reality

i. Hardware drives Software

ii. Multicore everywhere

iii. Everyone must be a parallel programmer
Hardware Drives Software

- Primary driver for software:
  - Efficiency and thus cycles were precious
  - Hardware folks have been hard at work
  - Counting cycles is no longer the developer’s primary worry
Hardware Drives Software

- Today’s development is at a high-level of abstraction (figuratively and literally at the cloud level):
  - Google’s 1st hit for lamp?
  - Django, Ruby, AJAX
  - Primary driver:
    - Laziness 😊 - write as little code as possible
    - Don’t repeat yourself (DRY)
    - Duplication is Evil

*LAMP* (software bundle) - [Wikipedia, the free encyclopedia](https://en.wikipedia.org/wiki/LAMP_(software_bundle))

*Lamp* - [Wikipedia, the free encyclopedia](https://en.wikipedia.org/wiki/Lamp)
Does Hardware drive Software?

- Software trends driven by:
  - Development productivity
  - What end-users want
- Maybe software is driving hardware?
  - Intel SCC, Larrabee
- Software trends have become decoupled?
Multicore Everywhere
Multicore Everywhere

<table>
<thead>
<tr>
<th>Number of cores</th>
<th>Today</th>
<th>2014</th>
<th>2018</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile-devices</td>
<td>1</td>
<td>3-4</td>
<td>9-10</td>
</tr>
</tbody>
</table>
Multicore Everywhere

Cloud-driven free performance

Energy vs. Performance

Traditional desktop
Parallelism is a great challenge, but:
- disruptive (change undergrad curriculum?)
- if necessary for multi-core, we probably needed it solved yesterday
- Very hard

If possible, lets not solve it, because:
- disruptive; if necessary, for multi-core we needed it solved yesterday; very hard
- Lets leave it as an open problem 😊
- May not be necessary
Cloud + Devices ➞ Parallelism Unnecessary (client-side)
Cloud + Devices

- Abundant concurrency at server
- Programmer in cloud
- Simple clients

Gurus (1271)

Average Joe programmer

Users (billions)

100 million
We probably agree that...

Efficiency is a primary goal...

If efficiency is a primary goal...

The fact that so much code is written at higher and higher levels of software may have system-wide implications...

... motivates a vertically integrated approach?
Data-Centric Concurrency

- Simple hardware, because of s/w model
- Simple s/w, because done at high-level
- Specialization for efficiency
Some Implications

- Computation
- Specialization
- Concurrency
- Coherence
- Scheduling
- Using the Cloud
- Beyond C & Java?

Efficiency is a primary goal
Hardware: Exploit Software Trends

- Use processors for computation
  - 128 ALUs consume less area than Atom’s L1 data-cache
  - Specialization is possible
- Map software abstraction to hardware
  - Staged databases
  - PLUGs map data-structures to hardware
Software: exploit app. trend

- Concurrency management

- Global resource management

- Storage
  - On devices
  - On cloud
Education

- Continue with C, Java, assembly?
- Or parallelism?
  - Some (small) number of programmers must wrestle with parallelism
  - Should it be pervasive?
- Or Cloud: Joe programmer is at this level
  - What higher level languages?
  - What abstractions, mechanisms, skills?
Conclusion

- Today’s processor is very capable...
  - This talk created and given on a $220 netbook

- S/W & programmer driven by productivity
  - Developing at cloud-level already

- Hardware must follow the S/W trend

- Should we rethink roll of parallelism?
Questions*

* Remzi is to blame for the things you disagree with...
Must Solve this Equation

Cloud computing

+ Multicore

= DCC
Ecosystem of Devices + Cloud
Users like it (want it)
Simple hardware and simple software
Must Solve this Equation

Cloud computing + Multicore = ?
One solution...

Does it exploit cloud + multicore synergy?
Do (real) developers care?
What does it provide to end-users?
Trends

- Driven by energy efficiency
- Hardware guys got lazy

Multicore
Trends

- Easy to deploy software
- Easy to deploy hardware
- Simple devices to access the cloud