

Some Thought on Parallel Programming and Multi-Core

Rich Brunner, AMD Fellow

April 2006

The Story So Far ...

- Brilliant researchers have advanced the cause of parallel programming by introducing:
 - New sophisticated programming paradigms/languages that naturally exploit parallelism of very expensive or specialized hardware.
 - Compilers and wizards that try to exploit parallelism of programs that were written with a serial mindset.
- Meanwhile, vast majority of applications programmers:
 - Have no idea about parallelism -- still fighting with Visual Basic.
 - Think C++ is hard enough, why worry about parallelism?
 - Don't know/care what MPI and OpenMP are; get confused by explicit parallel constructs anyway ...
 - Lack access to "parallel" hardware, even 2-processor development platforms.

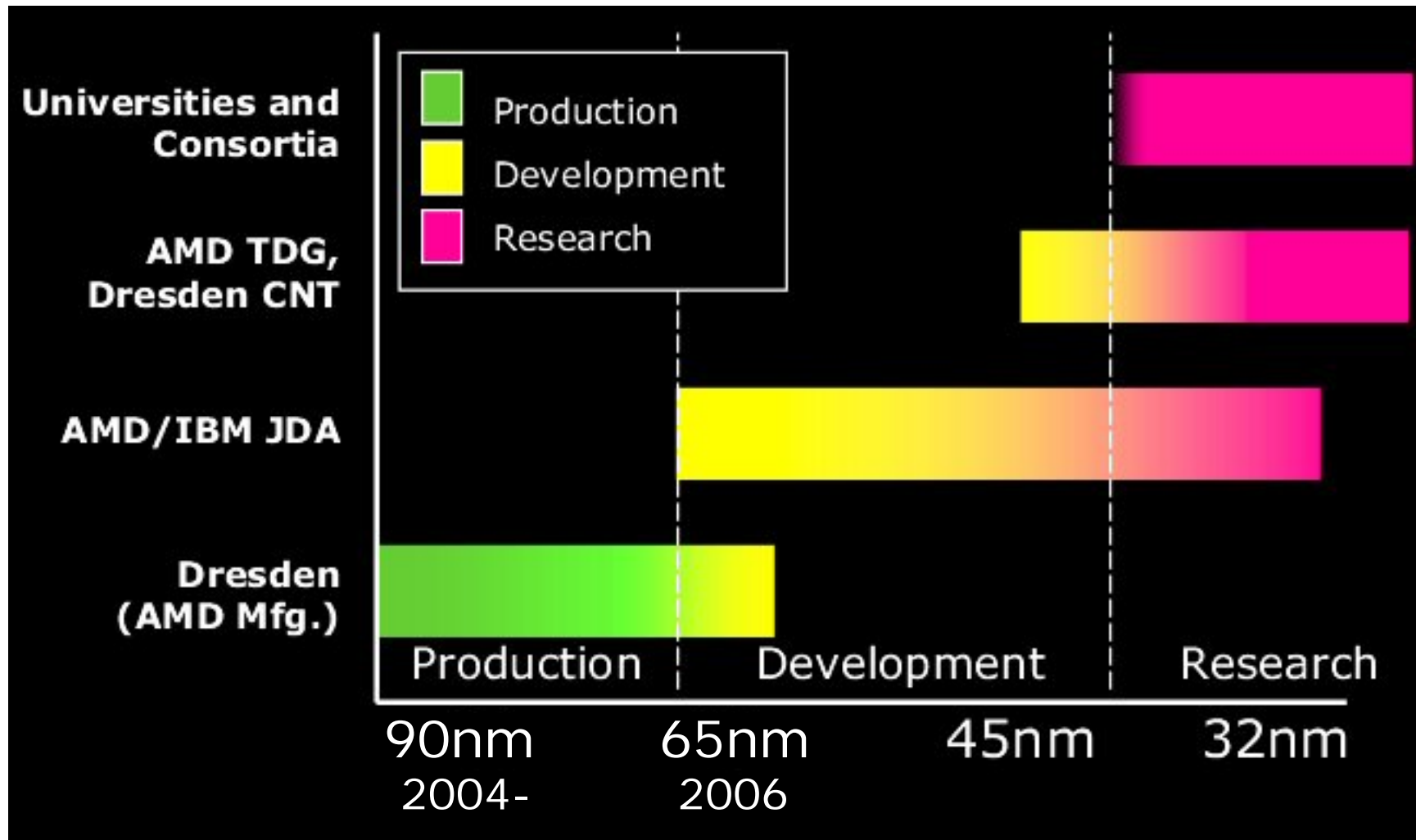
What is Changing?

- Commodity and Consumer Microprocessor vendors have “discovered” multi-threading.
 - High-end/Specialized Processor vendors have been doing this for many years.
 - But the commodity vendors believe they have solved the “Grand Challenge”:
 - *Staying profitable (in the long-term) selling parallel-processing hardware.*
 - Examples:
 - *IBM Cell Processors in PSP3*
 - *Multi-core processors from AMD, Intel, etc*
 - Why are we doing this?

Reason: Physics Is Not Always Kind

- Thermal density limits useful scaling of single-core frequency
 - Trying to work-around these limits breaks the platform infrastructure, increases cost and complexity, reduces yields and reliability.
- Meanwhile, we are able to add ever more transistors as we shrink process dimensions: 90nm, 65nm, 45nm, 32nm
 - But just shrinking from 130nm to 90nm, while reducing die area by 50%, doesn't help change power if you design transistors to MAX Frequency.
- Solution: re-balance the equation of performance, power consumption and die area:
 - Dropping the core frequency slightly can lead to a big power drop
 - *Transistors designed for < 100% of the MAX Frequency possible, are much more power efficient.*
 - Boost performance further (and use the die space) by adding multiple cores.
- E.g. at 90nm two extremely high performance processor cores fit easily on an affordable die size (100-200mm²)

AMD Technology Roadmap



Technologies Roadmap: Server and Workstation

2005

2006

2007

Processor



Dual Core
PowerNow!

Dual Core
Pacifica virtualization
Presidio security
Memory RAS

New core
Multi-core
Scale-up (32P+)
L3 Cache
Enhanced RAS
I/O Virtualization
HyperTransport™ 3.0

Chipset and Platform

PCI Express
Gigabit Ethernet
Serial ATA II
Software RAID 5
Hardware Firewall

PCI Express
Gigabit Ethernet
TCP Offload
Serial SCSI
Serial ATA II
Hardware RAID 5

HyperTransport 3.0
PCI Express 2
Gigabit Ethernet
TCP Offload
Serial SCSI
Serial ATA II
Hardware RAID 5
Fault Tolerant I/O

More Dual-core mid-2006, Quad core in 2007



2005 Analyst Day

Software Architecture Challenges

- Heterogeneous cores and Legacy OSes
- Non-uniform distributions of cores and Legacy OSes
- Describing Topology (and Legacy OSes)
- Exploiting new advances while not breaking Legacy OSes and Applications
- Debugging
- Performance Profiling
- Licensing of Applications
- Making multi-threading easier (or hiding it) from Application Programmers

Some Successes in Using Multi-Core

- There is enough OS multi-tasking to effectively utilize these cores right now even on desktop systems.
- Some Desktop Apps can do multi-threading.
- Virtualization: seems like a very good way to use multi-socket, multi-core platforms in the commercial space.
- Java programmers are more likely to thread; efficient JVMs can utilize the available hardware threads.
- Research in Hiding or simplifying synchronization from the application programmer: Transactional Memory, etc.
- Commodity x86 Multi-core platforms will soon be everywhere!
 - This can increase the support, \$\$, and interest in really trying to exploit small-scale parallelism.
 - This can also lead to the clueless hacking together and forcing some really ugly/broken solutions.

Trademark Attribution

AMD, the AMD Arrow logo, AMD Turion, and combinations thereof are trademarks of Advanced Micro Devices, Inc. in the United States and/or other jurisdictions. Other names used in this presentation are for identification purposes only and may be trademarks of their respective owners.

©2005 Advanced Micro Devices, Inc. All rights reserved.

