Memory Consistency Models: They are Broken and Why We Should Care

Sarita Adve

University of Illinois at Urbana-Champaign
sadve@Illinois.edu

Ken Kennedy Lecture

Work with numerous colleagues and students over 30 years

This work is currently supported in part by DARPA, NSF, and by the Applications Driving Architecture (ADA) Research center (JUMP center co-sponsored by SRC and DARPA)
My Story

Wisconsin

Rice

Illinois

Our community
<table>
<thead>
<tr>
<th>Location</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wisconsin</td>
<td>Question fundamentals</td>
</tr>
<tr>
<td>Rice</td>
<td>Believe in yourself</td>
</tr>
<tr>
<td>Illinois</td>
<td>Impact = Change minds. Takes time</td>
</tr>
<tr>
<td>Our community</td>
<td>Acknowledge your village. Pay it forward</td>
</tr>
</tbody>
</table>
• 1988 to 1989: What is a memory consistency model?
  – Simplest model: sequential consistency (SC) [Lamport79]
    • Memory operations execute one at a time in program order
    • Simple, but inefficient
  – Implementation/performance-centric view
    • Order in which memory operations execute
    • Different vendors w/ different models (orderings)
      – Alpha, Sun, x86, Itanium, IBM, AMD, HP, Cray, …
    • Complex, many ambiguities, …
  – A new memory model virtually everyday
• 1988 to 1989: What is a memory consistency model?

Memory model = What value can a read return?

Initially X=Y=Flag=0

<table>
<thead>
<tr>
<th>Thread 1</th>
<th>Thread 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>X = 26</td>
<td>if (Flag == 1) {</td>
</tr>
<tr>
<td>Y = 90</td>
<td>... = Y ← 90</td>
</tr>
<tr>
<td>...</td>
<td>... = X ← 26</td>
</tr>
<tr>
<td>Flag = 1</td>
<td>}</td>
</tr>
</tbody>
</table>

HW/SW Interface: affects performance, programmability, portability
• 1990-93: Software-centric view: Data-race-free (DRF) model
  – Sequential consistency for data-race-free programs [Adve, Hill ISCA90]
  – Distinguish data vs. synchronization (race)
    • Data can be optimized \(\Rightarrow\) \(\uparrow\) performance for DRF programs

Initially \(X=Y=\text{Flag}=0\)

\[
\begin{align*}
\text{Thread 1} & \\
X &= 26 & \text{Thread 2} & \\
Y &= 90 & \text{if } (\text{Flag} == 1) \{ & \\
... &= Y & \} \\
\text{Flag} &= 1 & \text{...} &= X & \\
\end{align*}
\]

Ack: Jim Goodman, Bart Miller, Rob Netzer, Kourosh Gharachorloo
Wisconsin ➔ Rice

“Two body problem” ➔
Two body opportunity

Dependence analysis, auto-vectorization, data parallel languages, parallel performance analysis tools, ...
1993-99: Performance benefits of relaxed models
   - New out-of-order processors emerging, new speculation techniques
   - No tools to understand performance implications
   - RSIM: Built first publicly available multiprocessor simulator with out-of-order processors [Pai et al. ASPLOS’96, ISCA’97, ...]

More confidence in DRF!
   - Called out compiler and PL community
   - Proceedings of IEEE paper caught attention of Bill Pugh
[with Bill Pugh, Jeremy Manson, Doug Lea, Hans Boehm, et al.]

- **2000-05: Java memory model** [Manson, Pugh, Adve POPL’05]
  - DRF model BUT racy programs need semantics
    - No out-of-thin-air values

Initially X=Y=0

Problem: Incredibly hard to formalize a spec that prohibits this result without prohibiting common optimizations

Java memory model = DRF + big mess
[With Hans Boehm et al.]

- **2005-08: C++ memory model** [Boehm, Adve PLDI’08]
  - DRF model BUT need high performance; mismatched hardware
  - Baseline DRF (DRF0) requires synchronization/atomics to be SC
  - Hardware vendors, software developers complained, but no option
  - Compromise: Relaxed atomics (only for experts)
    \[\Rightarrow \text{DRF + big mess}\]

Good news: After 20 years, convergence at last!

But: How to debug racy programs, how to avoid out of thin air values, no semantics for relaxed atomics, ...

CACM’10: Memory Models: A Case for Rethinking Parallel Languages and Hardware
C++17 "specification" for relaxed atomics

- Races that don't order other accesses
- Implementations should ensure no “out-of-thin-air” values are computed that circularly depend on their own computation

"C++ (relaxed) atomics were the worst idea ever. I just spent days (and days) trying to get something to work. ... My example only has 2 addresses and 4 accesses, it shouldn’t be this hard. Can you help?"

- Email from employee at major research lab
2008-14: Software-centric view for coherence: DeNovo protocol

- More performance-, energy-, and complexity-efficient than MESI
  - Began with DPJ’s disciplined parallelism
  - Identified fundamental, minimal coherence mechanisms
  - Loosened s/w constraints, but still minimal, efficient hardware

Ack: Marc Snir, UPCRC

Meanwhile: the end of Dennard and Moore’s laws

- Architecture enters golden age
  - Déjà vu for coherence and consistency

Next phase with Matt Sinclair and John Alsop, current group
The Golden Age of Specialization & Heterogeneity

Explosion of accelerators in SoCs

Source: Brooks, Wei group, http://vlsiarch.eecs.harvard.edu/accelerators/die-photo-analysis
Specialization Requires Better Memory Systems

Traditional heterogeneity

- Wasteful data movement
- No fine-grain synch
- No irregular access patterns

Coherent shared memory

- Implicit data reuse
- Fine-grain synchronization
- Irregular access

Existing solutions: complex & inflexible
CPU Coherence: MSI

- Single writer, multiple reader
  - On write miss, get ownership + invalidate all sharers
  - On read miss, add to sharer list

⇒ Directory to store sharer list
⇒ Many transient states
⇒ Excessive traffic, indirection

Complex + inefficient
• With data-race-free (DRF) memory model
  – No data races; synchs must be explicitly distinguished
  – At all synch points
    • Flush all dirty data: Unnecessary writethroughs
    • Invalidate all data: Can’t reuse data across synch points
  – Synchronization accesses must go to last level cache (LLC)

Simple, but inefficient at synchronization
• With data-race-free (DRF) memory model
  – No data races; synchs must be explicitly distinguished
  – At all synch points
    • Flush all dirty data: Unnecessary writethroughs
    • Invalidate all data: Can’t reuse data across synch points
  – Synchronization accesses must go to last level cache (LLC)
With data-race-free (DRF) memory model

- No data races; synchs must be explicitly distinguished
- At all synch points
  - Flush all dirty data: Unnecessary writethroughs
  - Invalidate all data: Can’t reuse data across synch points

- Synchronization accesses must go to last level cache (LLC)
- No overhead for locally scoped synchs

But higher programming complexity
Do GPU models (HRF) need to be more complex than CPU models (DRF)?

NO! Not if coherence is done right!

DeNovo+DRF: Efficient AND simpler memory model

[Sinclair et al. Micro’15]
A Classification of Coherence Protocols

- Read hit: Don’t return stale data
- Read miss: Find one up-to-date copy

<table>
<thead>
<tr>
<th>Track up-to-date copy</th>
<th>Invalidator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ownership</td>
<td>MESI</td>
</tr>
<tr>
<td>Writethrough</td>
<td>GPU</td>
</tr>
</tbody>
</table>

- Reader-initiated invalidations
  - No invalidation or ack traffic, directories, transient states

- Obtaining ownership for written data
  - Reuse owned data across synchs (not flushed at synch points)
  - Complexity, performance, energy
DeNovo shows 28% lower execution time than GPU with global synch
Global Synch – Energy

DeNovo shows 51% lower energy than GPU with global synch.
Do GPU models (HRF) need to be more complex than CPU models (DRF)?

**NO! Not if coherence is done right!**

DeNovo+DRF: Efficient AND simpler memory model
Heterogeneous Devices have Diverse Memory Demands

Typical **CPU** workloads: fine-grain synch, latency sensitive

Typical **GPU** workloads: spatial locality, throughput sensitive
Existing Solutions: Inflexible and Inefficient

Examples: ARM CHI, IBM CAPI, AMD APU
Spandex: Flexible Heterogeneous Coherence Interface

Adapts to exploit individual device’s workload attributes
Better performance, lower complexity
⇒ Fits like a glove for each device!

[Alsop et al. ISCA’18]
Key Components

- Flexible device request interface
- DeNovo-based LLC
- External request interface
- Device may need translation unit
Next steps: Dynamic coherence specialization

Exploit SW or HW hints about data access patterns

- Dynamic Spandex request selection
- Producer-consumer forwarding
- Extended granularity flexibility

⇒ Simple, Flexible, Efficient
Specialized coherence a la Spandex

Handle specialized memories in global address space
Scratchpad, FIFOs, …, compute-in-memory, HBM,
Stash: globally addressable scratchpads [ISCA’15]

Relaxed atomics
DRFrlx [Sinclair et al. ISCA’17]
SC-centric semantics for good code patterns
How to formalize other patterns?

Handle approximations & solution quality, security
Heterogeneous Parallel Virtual Machine (HPVM) [PPoPP’18]

– Virtual ISA, compiler IR (LLVM for heterogeneous systems)

Targets: CPUs, vector extensions, GPUs, FPGAs, domain specific accelerators

Model: Hierarchical dataflow graph with side effects

Runtime maps to accelerators

Another talk!
Looking Forward…

HPVM + DRF Consistency + ???

Software Innovations
- Synchronization locality
- Data locality, visibility
- Coarse-grain operations
- Producer/consumer relationships

Hardware Innovations
- hLRC adaptive laziness
- Coherent scratchpads (Stash, ISCA'15)
- Hardware queues
- Spandex dynamic caches
- HBM caches
- NVRAM
It takes a village to make a successful researcher

Paying it forward ...
Our Community: Paying it Forward

SIGARCH EC
Joel Emer
Babak Falsafi,
Natalie Enright Jerger,
Scott Mahlke,
Partha Ranganathan,
Karin Strauss,
David Wood

Natalie Enright Jerger,
Kim Hazelwood,
Margaret Martonosi,
Kathryn McKinley

Highlight: Diversity and Inclusion
A community effort to emulate
Our Community: Paying it Forward

Natalie Enright Jerger,
Kim Hazelwood,
Margaret Martonosi,
Kathryn McKinley

Highlight: Diversity and Inclusion

A community effort to emulate

Janie Irwin
Key Events Last Year in Architecture Community

Study shows poor gender ratios
- Keynotes, PC chairs, Awards
- All conferences must improve
- One stands out

Micro50: Legends of Micro panel
- All white, all male

Reading of Diversity Statement
- Call to action
- Clear public support for change

SIGARCH works for diversity
But study is wakeup call

SIGARCH Works to Improve Diversity
by Sameer in Oct 30, 2017 | Tags: SIGARCH, Diversity

Diversity in conference governance
- Institution, academic lineage, ...
Key Events Last Year in Architecture Community

SIGARCH CARES to Report on Discrimination and Harassment
By Sabine Aoue, SIGARCH Chair on Mar 1, 2016 | Tags: ACM SIGARCH, Discrimination, Harassment

SIGMICRO and SIGARCH Join Hands on CARES
By Saba Aoue, Michael Schindel, Margaret Mancini, Kathryn McKenzie on Mar 1, 2016 | Tags: Discrimination, Harassment

SIGARCH CARES: To help report harassment
Chairs: Martonosi, McKinley

SIGARCH works for diversity
But study is wake up call

SIGMICRO joins CARES

Welcome to the Women in Computer Architecture (WICARCH) community
By Natalie Enright Jerger on May 7, 2016 | Tags: Diversity

SIGARCH CARES to Report on Discrimination and Harassment
SIGMICRO and SIGARCH Join Hands on CARES

Get data
Raise awareness, fix problems
CARES, WiCarch, Bias busting workshop, Conference mentoring, ...

Study shows poor gender ratios
Keynotes, PC chairs, Awards
All conferences must improve
Micro stands out

SIGARCH CARES:
To help report harassment
Chairs: Martonosi, McKinley

SIGARCH Works to Improve Diversity
But study is wake up call

SIGMICRO joins CARES

WICArch is SIGARCH subcommittee
Web portal w/directory, profiles
Slack mentoring channel
Graduating women brochure
Strategizes diversity efforts
Chair: Enright Jerger

Get data
Raise awareness, fix problems
CARES, WiCarch, Bias busting workshop, Conference mentoring, ...

Welcome to the Women in Computer Architecture (WICARCH) community
By Natalie Enright Jerger on May 7, 2016 | Tags: Diversity

SIGMICRO and SIGARCH Join Hands on CARES
By Saba Aoue, Michael Schindel, Margaret Mancini, Kathryn McKenzie on Mar 1, 2016 | Tags: Discrimination, Harassment

SIGARCH CARES to Report on Discrimination and Harassment
By Sabine Aoue, SIGARCH Chair on Mar 1, 2016 | Tags: ACM SIGARCH, Discrimination, Harassment
Thank You!
Thank You!

Wisconsin: Question fundamentals
Rice: Believe in yourself
Illinois: Impact = Change minds. Takes time
Our community: Acknowledge your village. Pay it forward