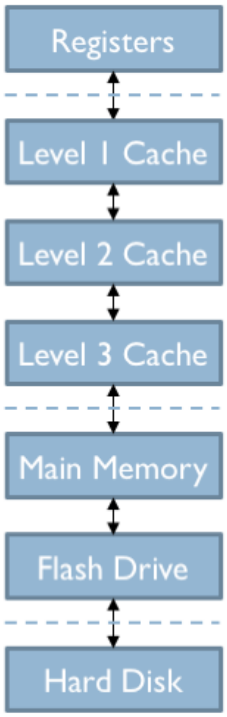
Locality Theory 101: Foundations of Fast Code

Chen Ding
Professor and Chair
Department of Computer Science
University of Rochester

Fastcode Online Seminar
15 December 2025

Memory Speed

- Program speed depends on speed of its data access
- Fundamentals of computer memory
  - Hierarchical
  - Shared
  - Dynamic: cache is computer's memory that forgets
- Locality theory
  - Analysis and optimization of the memory hierarchy



Chris Terman: Our "~~Computing~~ Memory Machine"

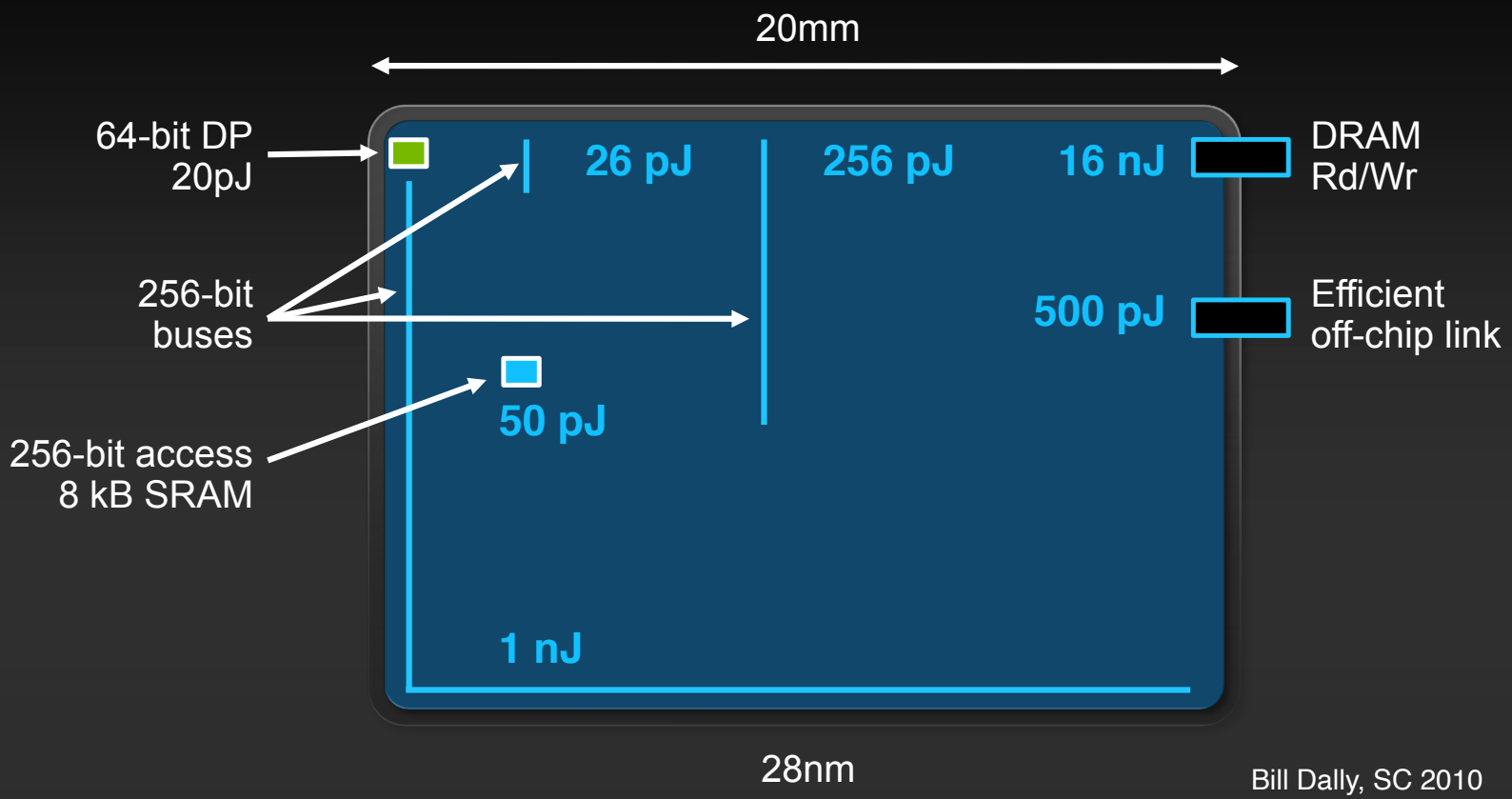
Access time	Capacity	Managed By
1 cycle	1 KB	Software/Compiler
2-4 cycles	32 KB	Hardware
10 cycles	256 KB	Hardware
40 cycles	10 MB	Hardware
200 cycles	10 GB	Software/OS
10-100us	100 GB	Software/OS
10ms	1 TB	Software/OS

Everything is a cache for something else

DOE's Office of Science builds large science facilities such as the Frontier supercomputer at Oak Ridge National Laboratory.

The High Cost of Data Movement

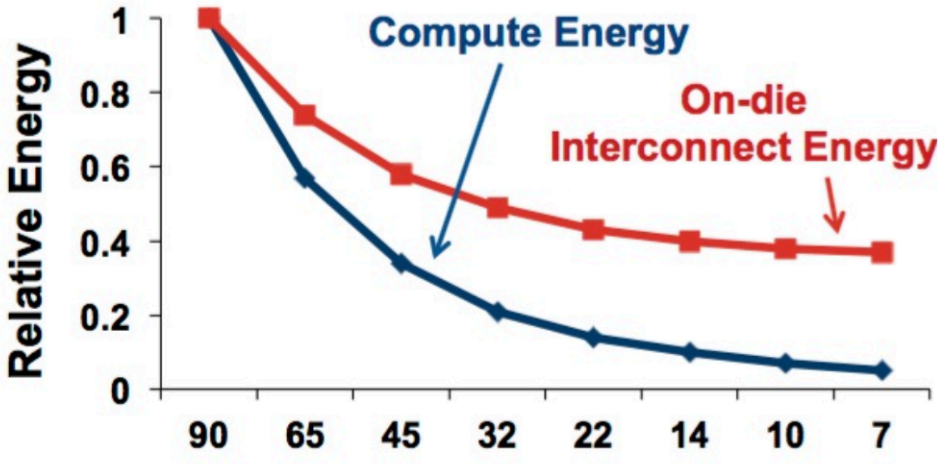
Fetching operands costs more than computing on them

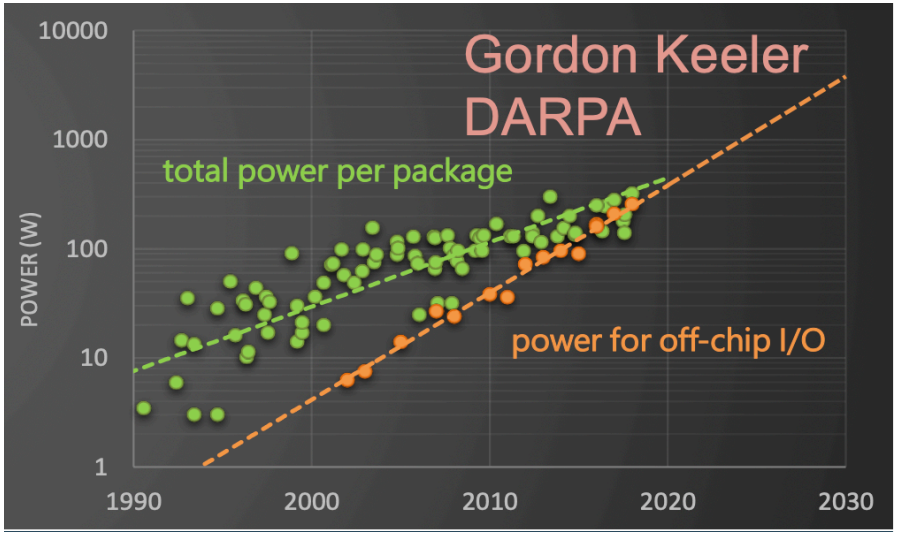


Bill Dally, SC 2010

John Shalf
Department Head for Computer Science
Lawrence Berkeley National Laboratory

Memsys 2025 Keynote: Codesign for Energy Efficient Computing
Adaptable Memory Systems for the Future of AI and HPC



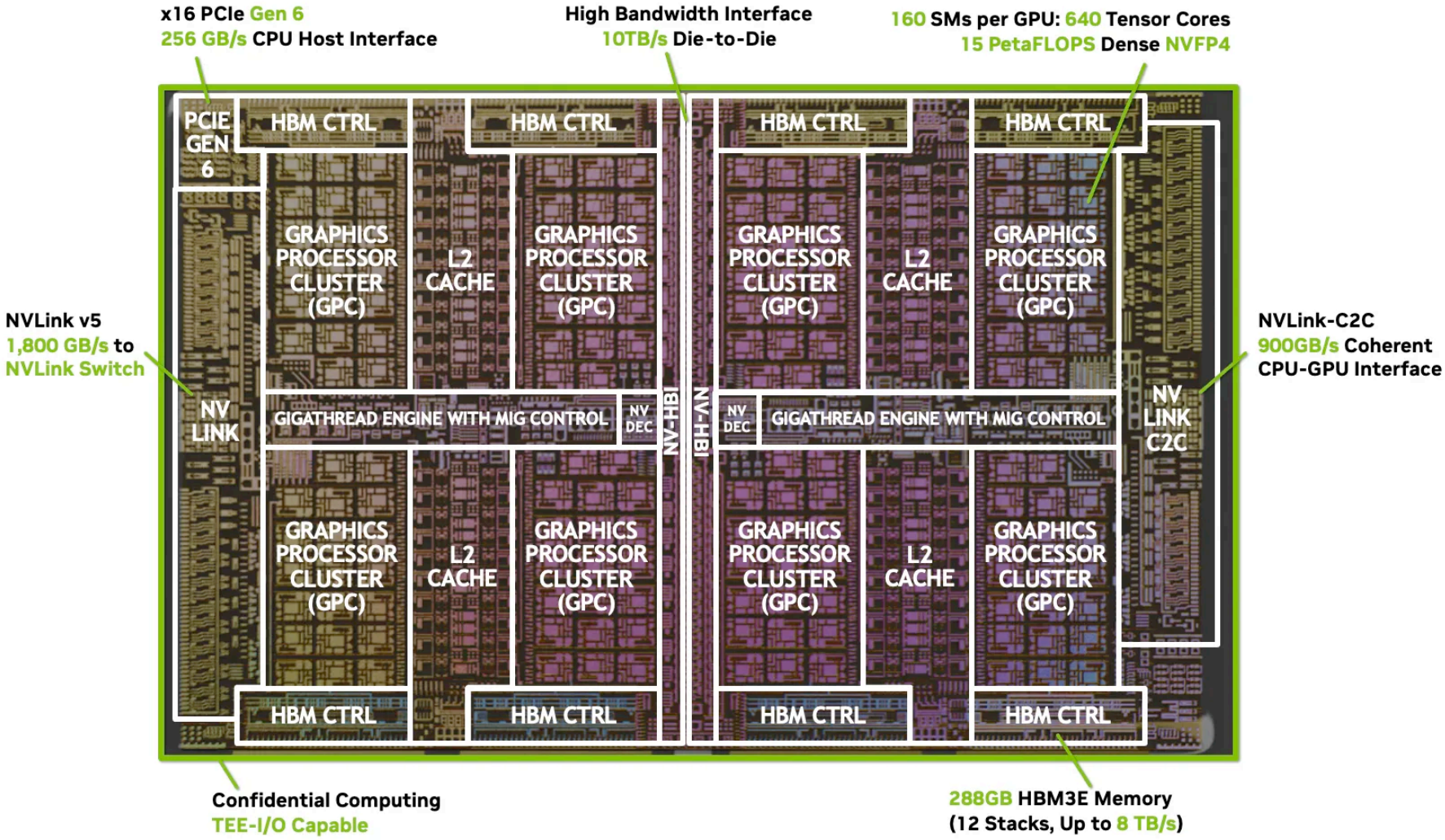


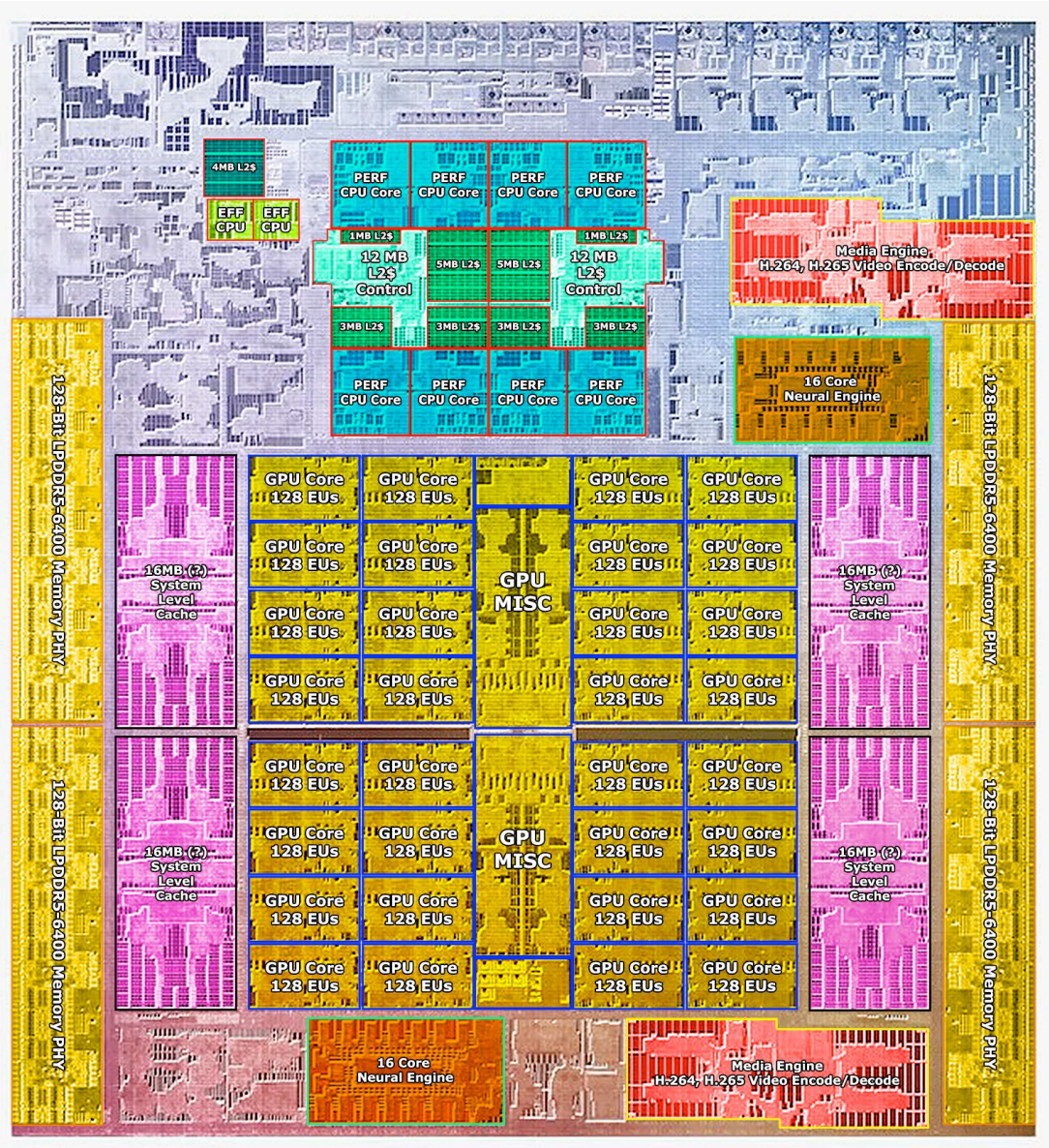
Locality

- Locality means proximity in hardware
  - Closest access, shortest data movement
- What is locality in software?
- Why do I care?

> **The ultimate goal of all computer science is the program. The performance of programs was once the noblest function of computer science, and computer science was indispensable to great programs.**
>
> **-Noble and Biddle, Notes on Postmodern Programming**

NVIDIA Blackwell Ultra GPU





Decoded Apple M1 Max SOC

Released October 15, 2025

Exploring multi-level cache prefetching for fabric attached memory

Chandrahas Tirumalasetty, Narasimha Reddy

MEMSYS 2025

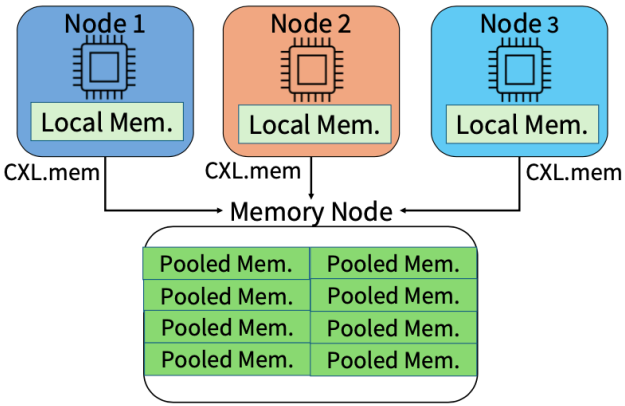


Figure 1: Logical view of multiple nodes pooling capacities from memory node using CXL.mem protocol

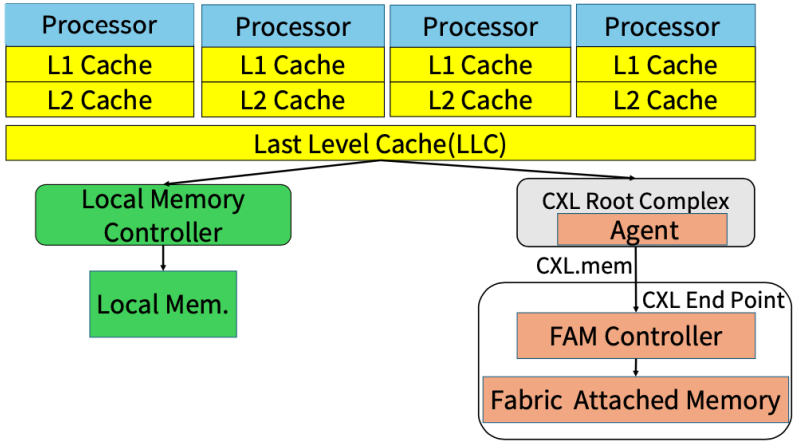


Figure 2: CXL & Fabric Attached Memory(FAM) Architecture

From Proximity to Locality

- Hardware proximity in data access
  - Fast, high bandwidth, energy and power efficient
- What is locality in software?
  - The miss ratio
    - Precise but machine dependent
  - Temporal/spatial locality
    - Qualitative, not quantified
- Introducing locality theory
  - Locality measures, in particular, data reuse and working set
  - Formal analysis and optimization

Data Access Complexity: Monotonicity and Proportionality

Chen Ding Yifan Zhu

University of Rochester

MEMSYS 2025

Time Complexity vs Data Access Complexity

- **Time Complexity**: Measures "processor work" (operations)
- **Data Access Complexity**: Measures "memory work" (data transfers)
- Locality = Lower data access complexity

Definition (Data Reuse)

Two consecutive accesses to the same data item

Reuse Measures

- **Reuse Interval (RI)**: Time between consecutive accesses
- **Reuse Distance (RD)**: Number of distinct items accessed between reuses

Example

For sequence "abcca":

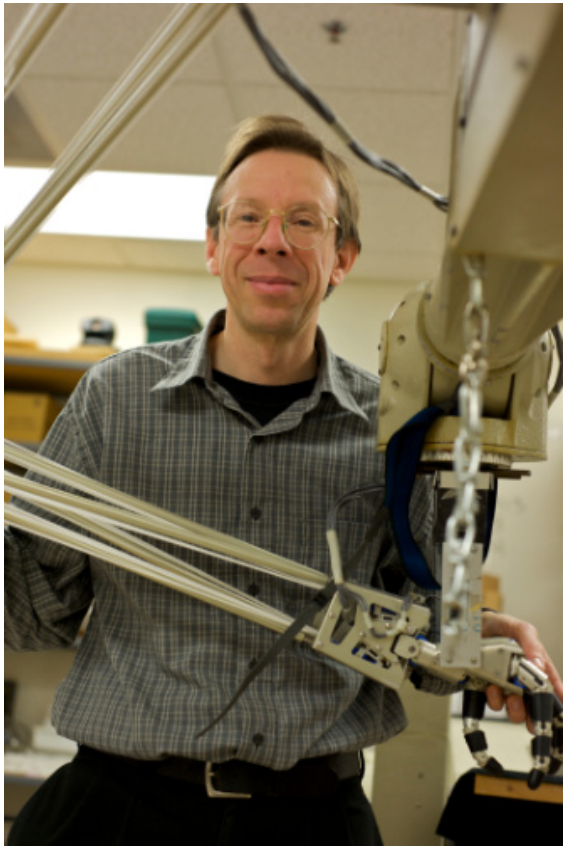
- Reuse of 'a': RI $= 4$, RD $= 3$
- Reuse of 'c': RI $= 1$, RD $= 1$

Focus

Data movement = **demand caching** (no prefetching).

Science is the art of measurement.

Randal C. Nelson (1958—2020)



- Dewey: The scientific attitude is experimental as well as intrinsically communicative.

The Reuse Distance

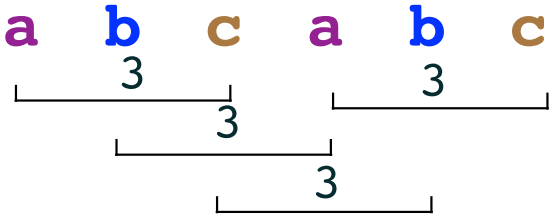
- Reuse distance
  - amount of data between use and reuse
  - same as LRU stack distance [Mattson et al. IBM 1970]
- Shorter reuse distance -> more data reuses -> better locality

∞	∞	∞	3	3	3
a	**b**	**c**	**a**	**b**	**c**

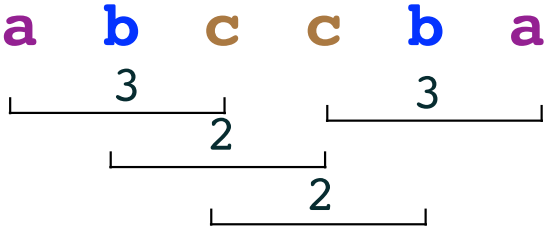
∞	∞	∞	1	2	3
a	**b**	**c**	**c**	**b**	**a**

The Footprint

- Working set theory [Denning CACM 1968, Denning and Schwartz CACM 1972]
  - Average working-set size (WSS) in an infinite stationary process
  - Computed iteratively (Denning Recursion [Yuan et al. TACO 2019])
- Working set in a program execution [Xiang et al. PPOPP 2013, PACT 2013]
  - Footprint, fp(x): average WSS for all windows of length x (x >= 0)
  - Parameter x is a timescale



fp(3) = 3



fp(3) = 10/4 = 2.5

A Relational Theory of Locality

- Common problems of poor locality
  - Too many cache misses
  - A lack of data reuse
  - Too large working sets
  - Are these three monsters or a single monster with three heads?

CERBERUS

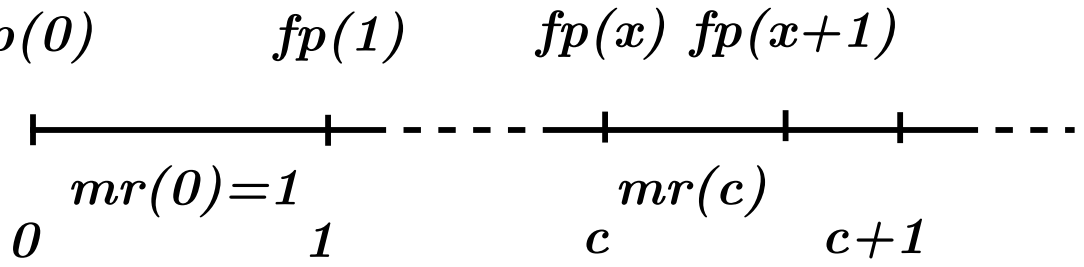
- Relational theory
  - How to convert between locality measures
  - From observation to explanation
    - Correlation, if not causation
- Science quantifies relationship between different measures

$E=mc^2$

Higher Order Theory of Locality [Xiang et al. ASPLOS13]

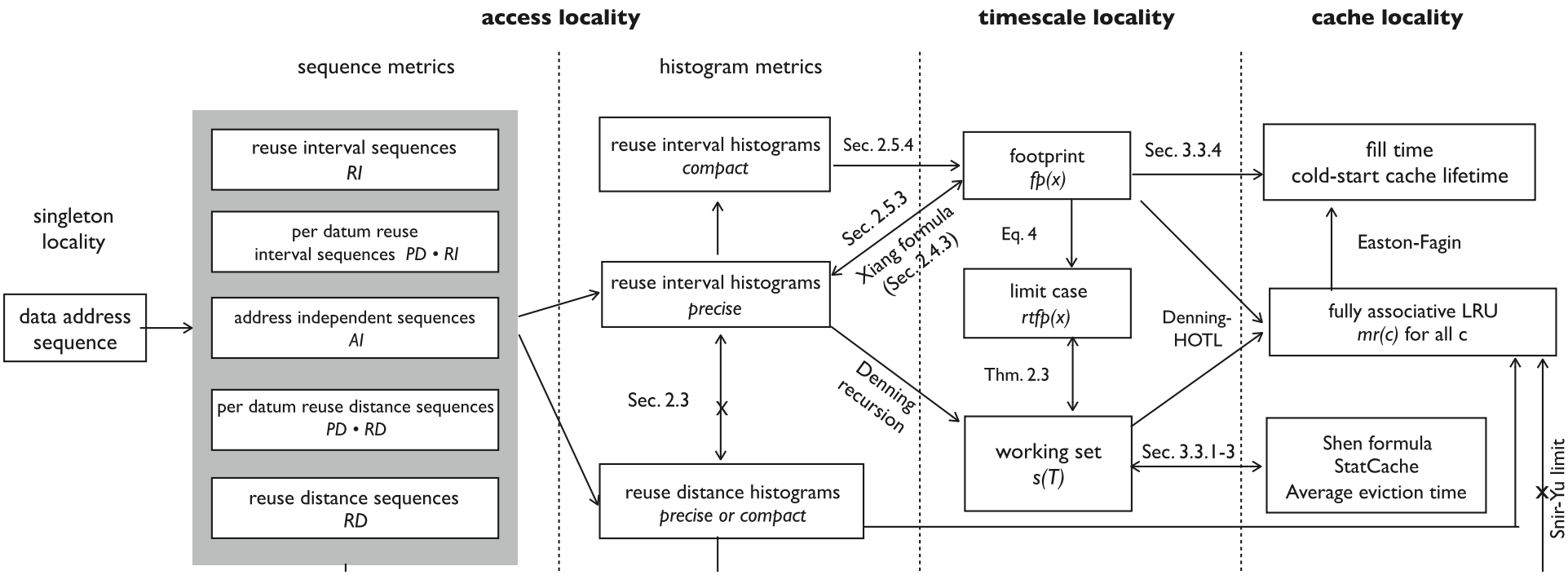
- HOTL conversion from footprint to miss ratio

$$mr_{HOTL}(c) = \Delta fp(x)\Big|_{fp(x)=c}, \text{ where } \Delta fp(x) = fp(x+1) - fp(x)$$



$$mr(0) = \Delta fp(x)\Big|_{fp(x)=0} \qquad mr(c) = \Delta fp(x)\Big|_{fp(x)=c}$$

Relational Theory of Locality [TACO 2019]



Locality "Metrics"

Four properties of a metric $d : X \times X \to [0, \infty)$ are:

1. **Non-negativity**

   $d(x, y) \geq 0$

2. **Identity of indiscernibles**

   $d(x, y) = 0 \iff x = y$

3. **Symmetry**

   $d(x, y) = d(y, x)$

4. **Triangle inequality**

   $d(x, z) \leq d(x, y) + d(y, z)$

- Do they qualify as metrics?
  - RD as a distance between the use and the reuse?
  - RI as a distance?
  - WSS of a window as a distance between two end points?
- Should RD of "aa" be 0 or 1?
- RI/RD/WSS of "a" and ""?

A Note on Cache Management

LRU Cache

- Fixed size c
- Evicts Least Recently Used
- Miss when RD $> c$
- Most common in hardware

Working-Set Cache

- Variable size
- Data stays for time x
- Miss ratio: $mr(c) = P(ri > x)$
- Theoretical model

Lease Cache Hardware and Software (Cache Programming using Leases)

Joint work with Professor Dorin Patru at RIT
[ASPLOS'19, MEMSYS'21, TACO'22 (miss ratio convexity), TACO'23, MEMSYS'24, MEMSYS'25]

NSF Grants 1909099 and 2114285

Chen Ding, University of Rochester

Advance in Locality Theory over Time

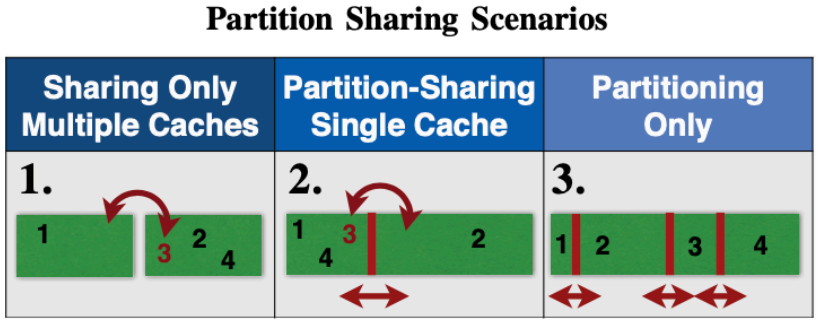
- Reuse distance histogram
  - Numerous applications (see Zhong et al. TOPLAS 2009)
  - Four decades of algorithmic improvements since 1970
- Footprint function
  - Linear complexity through Xiang formula
  - HOTL conversion is a mathematical operation
- New since 2010s
  - Analysis and optimization entirely as math operations on functions
  - Footprint locality is defined on all non-negative integers
    - Monotone and (largely) concave [Xiang et al. PACT 2013]

Locality Optimization

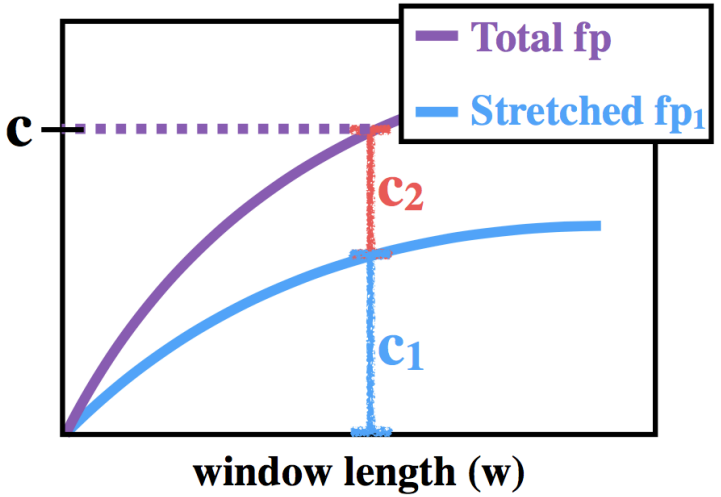
- Going beyond intuition
  - Complex dynamic systems defy simple intuition
  - Rules of thumbs are imprecise and brittle
    - e.g. sqrt(2) rule of cache scaling
    - Imprecise: only partially quantitative
    - Brittle: unclear when it stops being applicable
- From ad-hoc to mathematical analysis
  - Shared cache performance
    - The miss ratio is not composable, but the footprint is [Ding et al. JCST 2014]
  - From "blackbelt" programming and heuristics to smart math


AAAS
Science
4 SEPTEMBER 2025
A "last gift" from people with HIV to the hunt for a cure p. 968
Enhancing sensitivity of gravitational-wave detectors with machine learning p. 1012
Estrogen effects on the kidneys decrease preeclampsia risk p. 1016
RULES OF
THUMB
The importance of a hand that holds in the evolution of rodents p. 1049

Optimal Partition-Sharing [Brock et al. ICPP 2015]



$$fp(w, ar_1, ar_2) = fp_1\left(w * \frac{ar_1}{ar_1 + ar_2}\right) + fp_2\left(w * \frac{ar_2}{ar_1 + ar_2}\right)$$

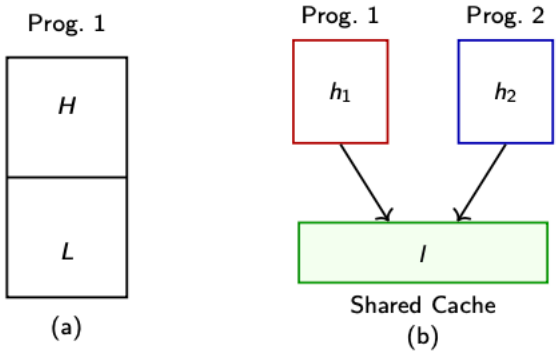


Exclusive Cache Hierarchy

Split LRU Stack Abstraction

Models exclusive cache hierarchy with two partitions:

- Upper partition H: Private cache (size h)
- Lower partition L: Shared victim cache (size l)
- Data evicted from H becomes victims stored in L



Victim Cache Requirement (VCR)

VCR Equation: For a single program using an exclusive hierarchy, the miss ratio must equal that of a combined cache:

$$vmr(h, l) = mr(h + l) \quad \text{for all } h, l \geq 0$$

Intuition: VCR ensures consistency with a single-level cache.

Footprint in Victim Cache [Ye et al. TACO 2017]

Victim Footprint (VFP) Definition

$$VFP(h, x) = fp(x_h + x) - h, \quad \text{where } fp(x_h) = h$$

- fp: footprint in single-level cache
- h: upper level cache size

Theorem (VFP Theorem 3.1)

The VFP defined above is the **only** *solution that satisfies the Victim Cache Requirement.*

Theoretical Significance

- **Uniqueness**: VFP is the only solution satisfying VCR
- **Composability**: VFPs of individual programs can be combined to model shared victim cache

Summary

- Essence of computer memory
  - Hierarchical, dynamically managed, and shared
- Relational theory of locality
  - Metrics: RI, RD, footprint
  - Conversions: from RI and RD to miss ratio
    - HOTL: from footprint to miss ratio
- Theory-based optimization
  - Cache partition-sharing (function scaling and composition)
  - Victim footprint (equation solving, function translation)
  - Polynomial data access complexity (on-going work)