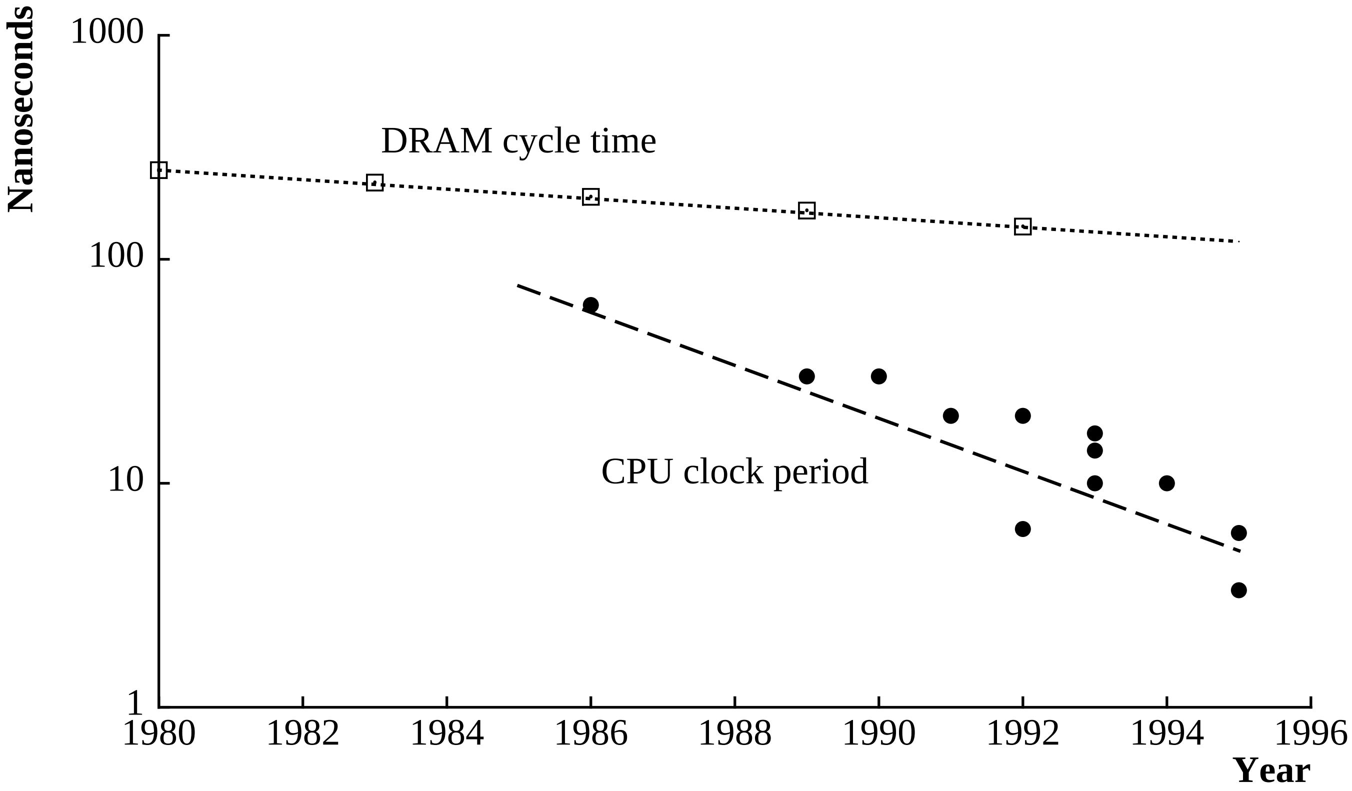
Accounting for Memory Bank Contention and Delay in High-Bandwidth Multiprocessors

Marco Zagha (Carnegie Mellon)

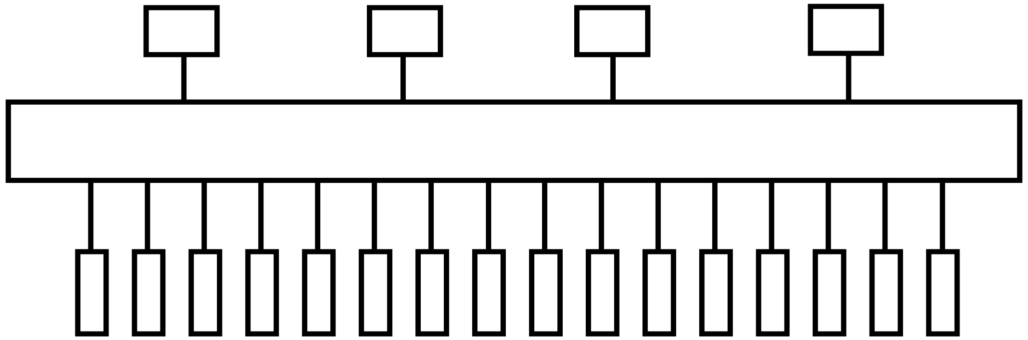
Joint work with:

Guy Blelloch (Carnegie Mellon)
Phil Gibbons (Bell Labs)
Yossi Matias (Bell Labs)

Memory speeds are improving very slowly

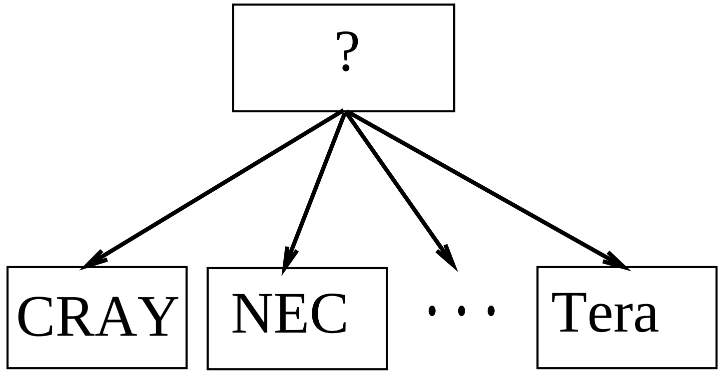


Memory bank expansion provides high bandwidth



Machine	Procs (p)	Banks (B)	Expansion ($x = B/p$)	Delay d
NEC SX-3	4	1024	256	8
Cray C90	16	1024	64	6
Cray J90	16	512	32	14
Convex C4	4	128	32	8
Meiko CS-2 node	2	16	8	
Tera MTA	256	512×64	128	
SGI Power Challenge	18	8×8	3.5	

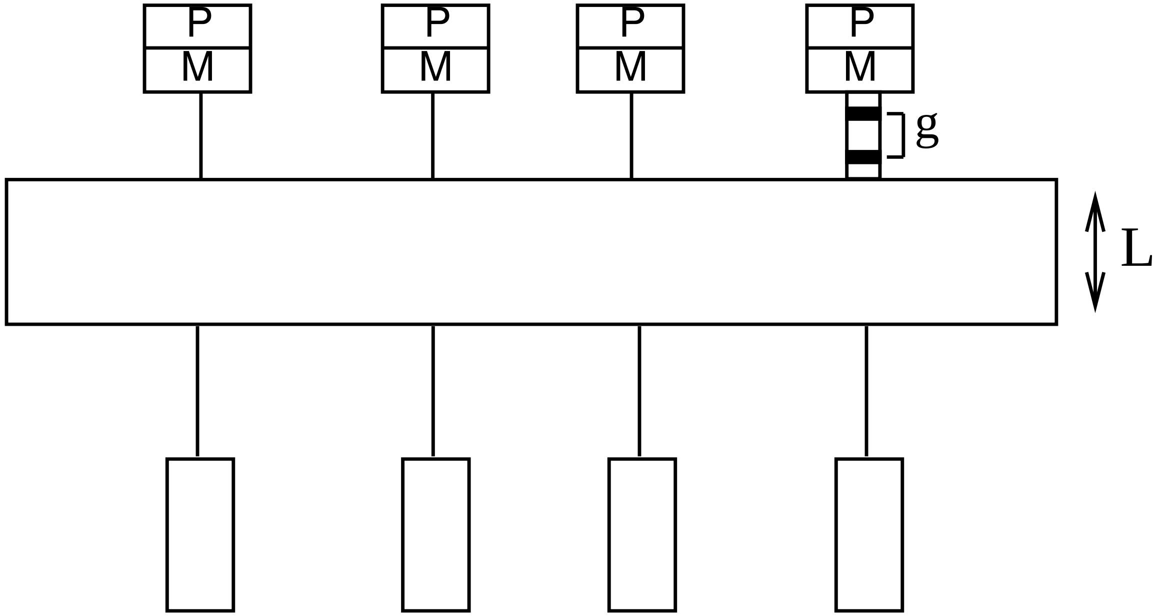
What is a good performance model?



Candidates:

- PRAM
- LOGP [Culler *et al.*]
- Bulk Synchronous Parallel (BSP) [Valiant]

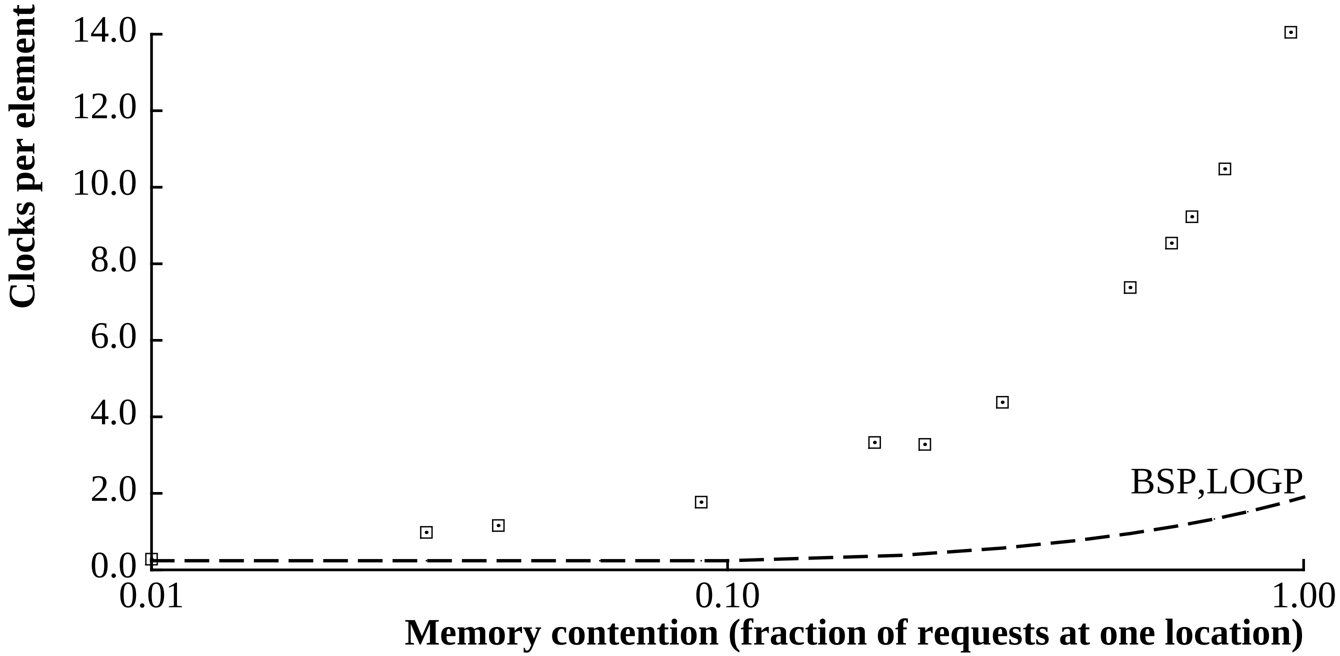
One candidate: the BSP



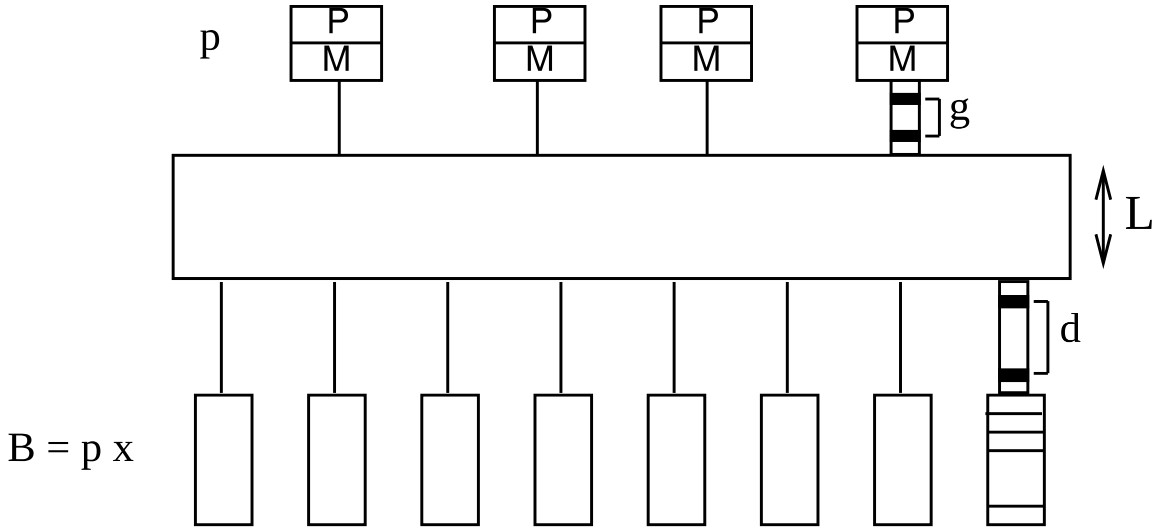
p number of processors
g cycles between memory requests at a processor (gap)
L the periodicity factor (latency + synchronization)

Are existing models accurate enough?

A CRCW PRAM algorithm for connected components running on an 8-processor Cray J90:



The (d,x)-BSP ($d$elu$x$e BSP)



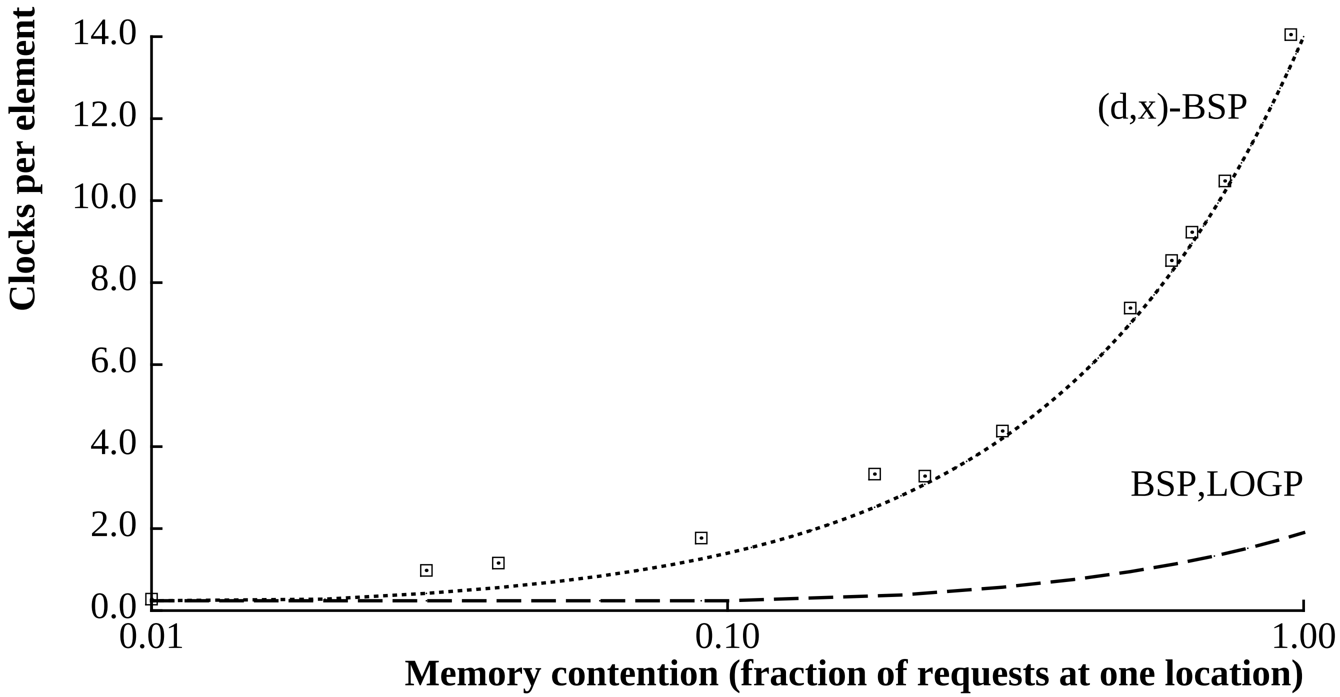
d cycles between memory accesses at a bank (delay)
x memory bank expansion factor (B/p)

(d,x)-balanced if $x = d/g$

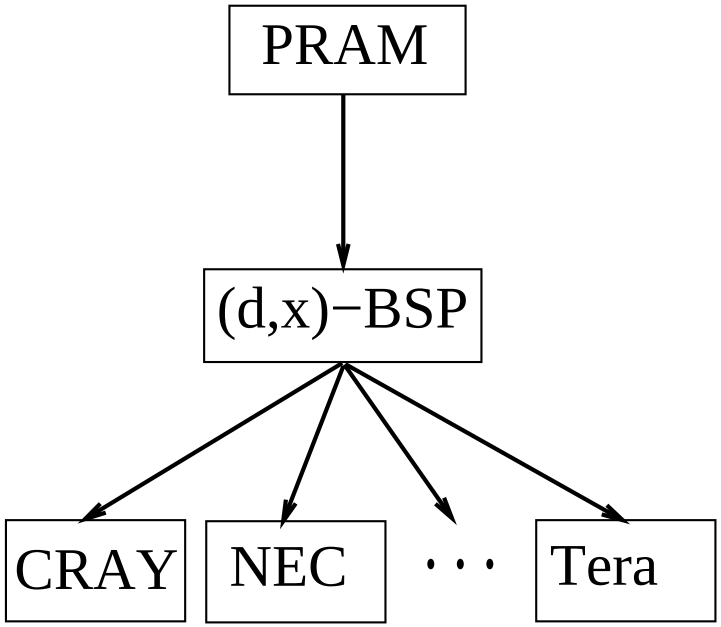
T = max(W, $g \cdot S$, $d \cdot R$, L)

Predictions with the deluxe BSP

Connected components on the Cray J90:



The deluxe BSP as a bridging model



1. How accurate is the deluxe BSP?
2. What is a good programming model?
3. How can that model be mapped onto the deluxe BSP?

Experimental Platform

Features of Cray vector multiprocessors:

- fine-grained random access
- hardware support for latency hiding (gather/scatter)
- high-bandwidth interconnection network

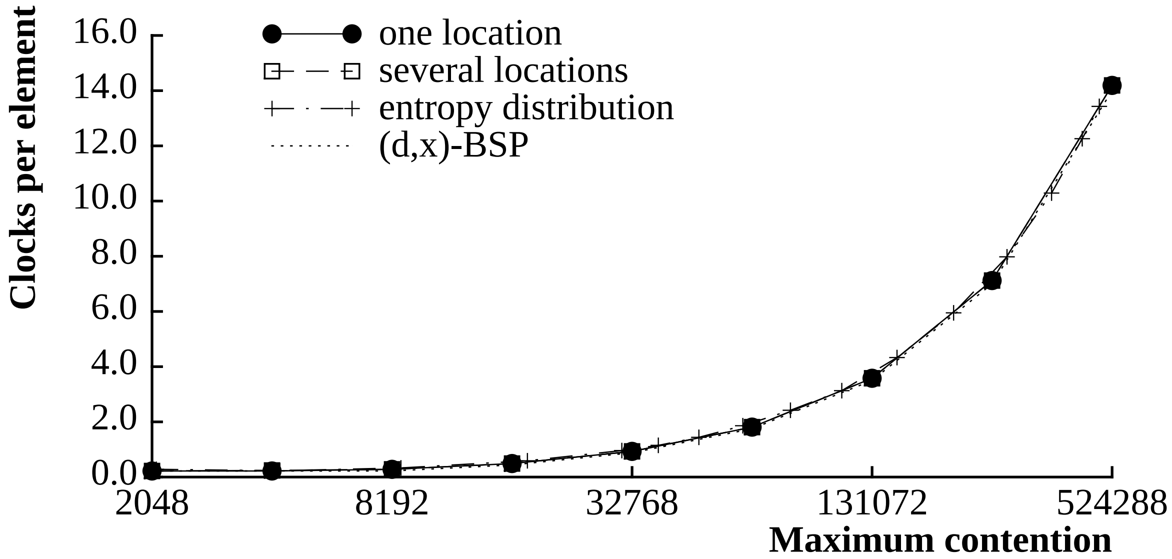
Parameters:

- Cray C90: $g = 1.2$, $d = 6.0$ (SRAM)
- Cray J90: $g = 1.8$, $d = 14.0$ (DRAM)

Assumptions:

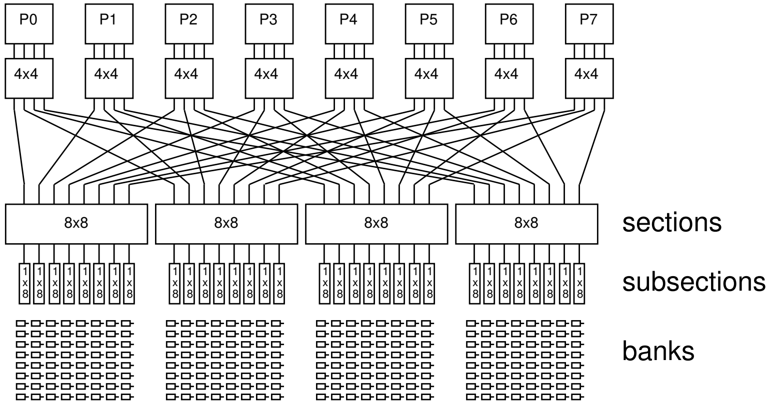
- latency hidden (high slackness)
- random injection from each processor

Is the cost model accurate?



What about network contention?

The Cray Y-MP network (from [Smith and Taylor 91]):



Experiments:

Conclusion: network effects are small for random mappings

What is a good high-level model?

Several flavors of PRAM contention rules:
concurrent, exclusive, or queued access

Motivation for the QRQW PRAM [Gibbons, Matias, Ramachandran 94]:

- CRCW is too powerful
- EREW is too strict
- QRQW matches most machines

Simulating the QRQW PRAM on the deluxe BSP

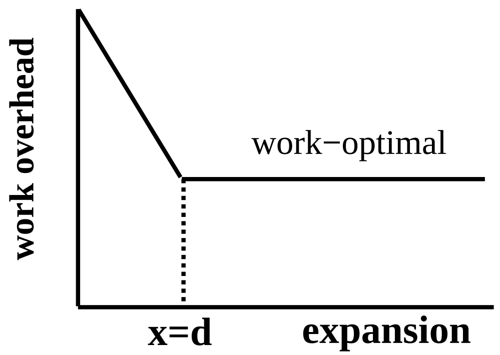
Previous work:

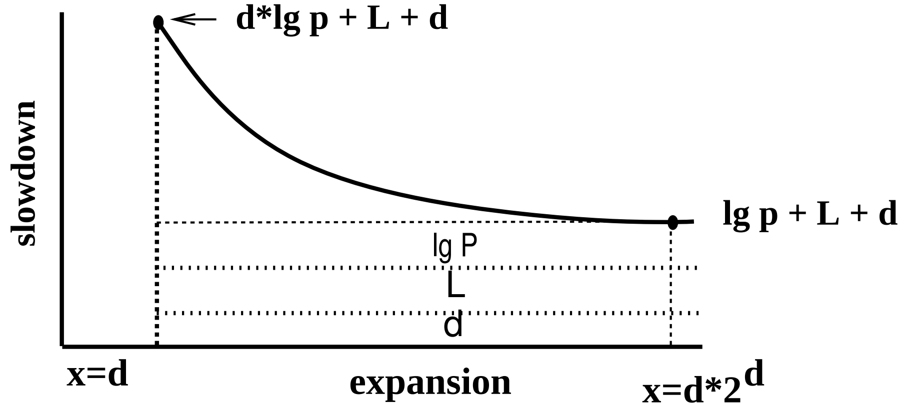
- EREW PRAM simulation on BSP [Valiant]
- QRQW PRAM simulation on BSP [Gibbons *et al.*]

How does the delay change the results?

- Is the emulation work-preserving?
- How much slackness is required?
- Can the memory bank delay be absorbed with more banks?
- Are the constants reasonable?

Theoretical results





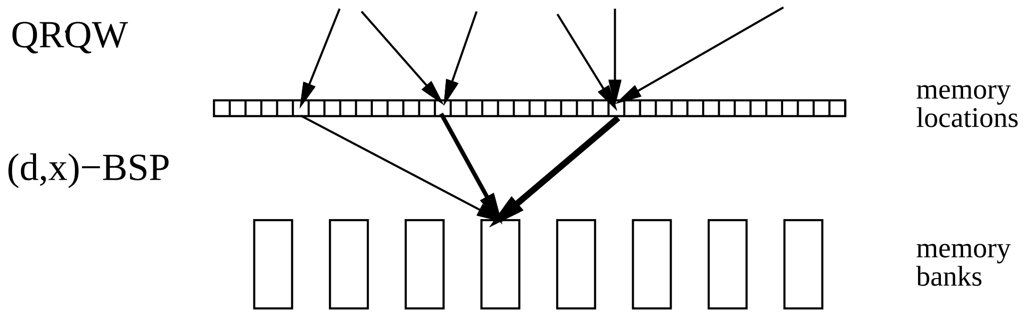
$x < d$:

- inherent work overhead
- emulation that minimizes overhead

$x \geq d$:

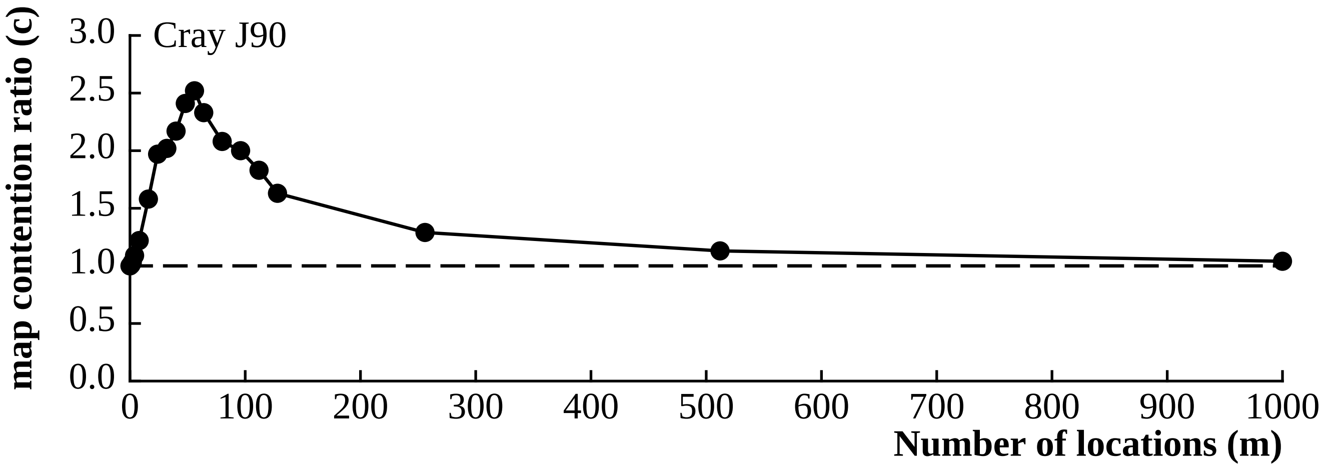
- *work-preserving* emulation using random hashing
- required slackness of $\delta \cdot \lg p + L + d$
- $\delta = 1$ for $x \geq d \cdot 2^d$

Analysis of random mappings



Analysis: tail of weighted sums of Bernoulli trials. . .

How big are the emulation constants?



Other experiments

- Algorithms:
  - Sparse matrix multiplication
  - Binary search
  - Random permutation
- Cost of computing hash functions
- Effect of increasing expansion with random mappings

Future Work

- experiment with other high-bandwidth machines
- implement more QRQW algorithms
- practical CRCW simulation on the deluxe BSP
- analyze bank caches: how much cache at the bank?

Conclusions

- necessary to account for delay when contention is high
- deluxe BSP serves as a bridging model:
  - models performance of high-bandwidth machines
  - efficient QRQW emulations in theory and in practice