OPTIMIZING GOOGLE'S WAREHOUSE SCALE COMPUTERS:

THE NUMA EXPERIENCE

Lingjia Tang, Jason Mars

UCSDCSE
Computer Science and Engineering

Xiao Zhang, Robert Hagmann, Robert Hundt, Eric Tune

Google

Warehouse Scale Computers

“Datacenters have become as vital to the functioning of society as power stations”
- *The Economist*

- Host large-scale Internet services (websearch, mail, etc)
- Expensive: hundreds of millions of dollars

Warehouse Scale Computers

"Datacenters have become as vital to the functioning of society as power stations"
- *The Economist*

- Host large-scale Internet services (websearch, mail, etc)
- Expensive: hundreds of millions of dollars
- Efficiency is critical

Inefficiencies

- Inefficiencies and missed optimization opportunities

Inefficiencies

- Inefficiencies and missed optimization opportunities
  - Lack of understanding of interaction between applications and micro-architectural features/properties

Inefficiencies

- Inefficiencies and missed optimization opportunities
  - Lack of understanding of interaction between applications and micro-architectural features/properties
  - Micro-architecture properties are abstracted away
    - a collection of thousands of cores, terabytes of main memory, petabytes of disk space, etc.
    - cannot adequately manage micro-architectural resources and features such as on-chip caches, non-uniform memory access, off-chip bandwidth, etc.

NUMA

NUMA

- NUMA is such a property
  - Old concept, yet limited understanding in new domain (new architectural implementations)
  - Software systems inadequate at effective management
  - Interaction between emerging applications in modern large scale WSCs unclear

NUMA

- NUMA is such a property
  - Old concept, yet limited understanding in new domain (new architectural implementations)
  - Software systems inadequate at effective management
  - Interaction between emerging applications in modern large scale WSCs unclear
- How do we understand the interaction?

Status-Quo

- Performance analysis in controlled environment
  - narrow focus; cannot replicate all aspects of the real production environment in a small-scale
  - miss the big picture
- Production study
  - Monitor datacenters with live services, interpret data

Challenges in Production Study

- Scale and complexity, intertwined performance factors
- Unknown factors, change spontaneously (load/user behavior, etc)
- Noisy performance data
- Inexplicable performance swing
  - 4x range of average request latency during a week's time for Gmail backend
- 1% performance improvement means millions

Challenges in Production Study

- Scale and complexity, intertwined performance factors
- Unknown factors, change spontaneously (load/user behavior, etc)
- Noisy performance data
- Inexplicable performance swing
  - 4x range of average request latency during a week's backend
- 1% performance improvement means millions

Difficult to reason about each individual microarchitectural factor' effect on applications

Methodology

- Controlled experiment vs. in-Production study

Methodology

- Controlled experiment vs. in-Production study
- need both
  - Production: identify evidence of a performance opportunity
  - Controlled: isolate and pinpoint the important factors related to the opportunity.

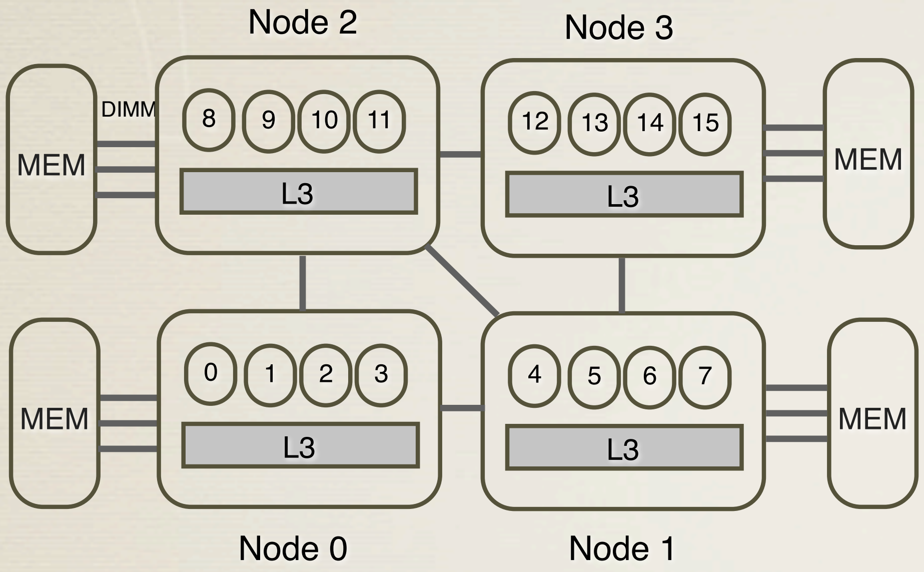
Methodology

- Controlled experiment vs. in-Production study

- need both
  - Production: identify evidence of a performance opportunity
  - Controlled: isolate and pinpoint the important factors related to the opportunity.

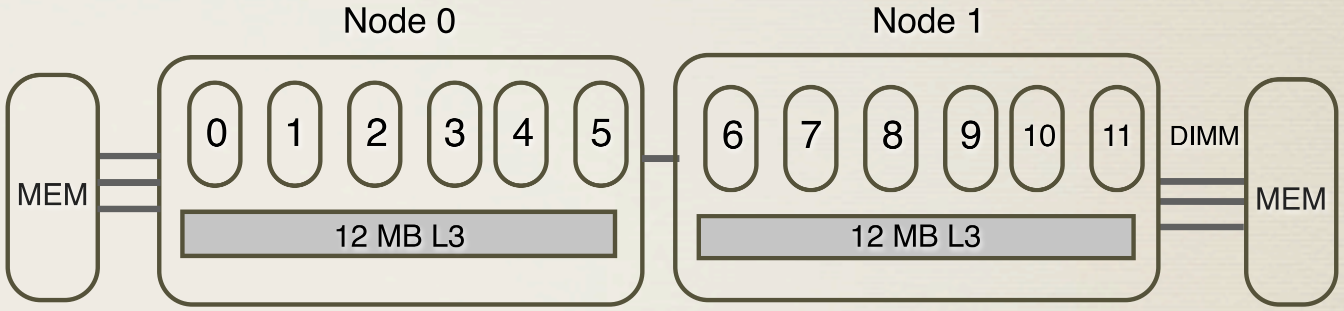
- NUMA
  - Performance impact of NUMA
  - Gmail backend and websearch frontend

NUMA (Non-Uniform Memory Access)



AMD Barcelona

- local memory
- 1-hop away
- 2-hop away



Intel Westmere

- local memory
- 1-hop away

Production Study

- What's the performance impact of NUMA in datacenters?
- What data to collect
  - **Metric**: to quantify the NUMA status
- How to collect them
  - **Profiling and monitoring:** lightweight, low overhead, for large-scale system
- How to interpret data
  - **Analysis**: Careful correlation and analysis of noisy data

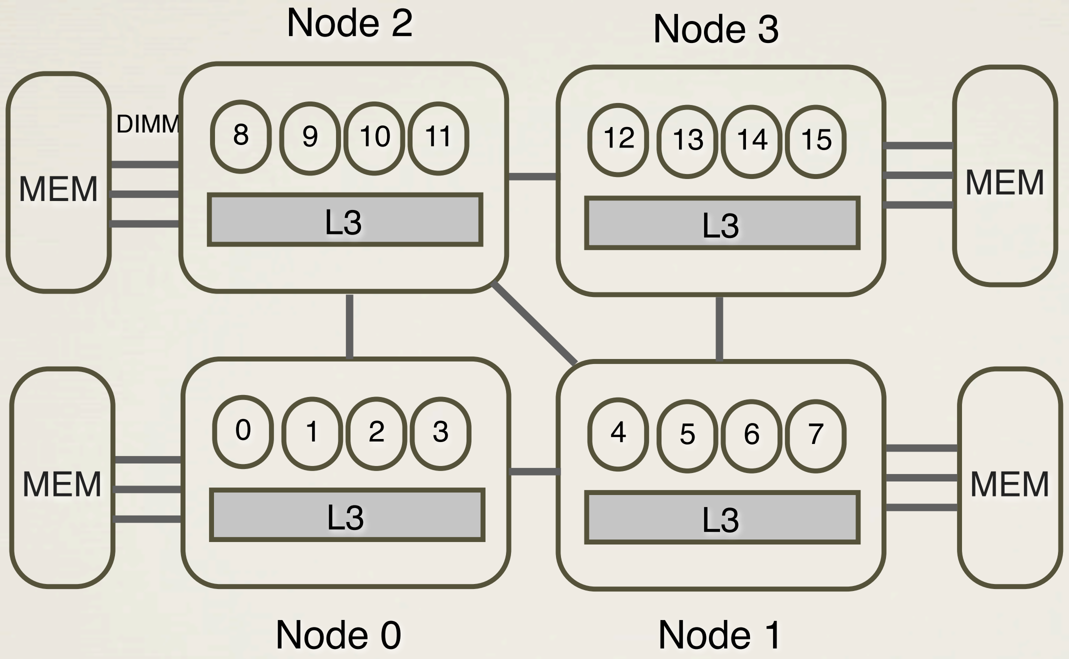
Metric: A job's NUMA Score

$$Score = \sum_{i=1}^{n} \sum_{j=1}^{n} C[i] \cdot M[j] \cdot \frac{D(i,i)}{D(i,j)}$$

- *C[i]:* normalized CPU usage for node *i*
- *M[j]:* normalized memory usage for node *j*
- *D(i,j):* distance between two nodes *i* and *j*

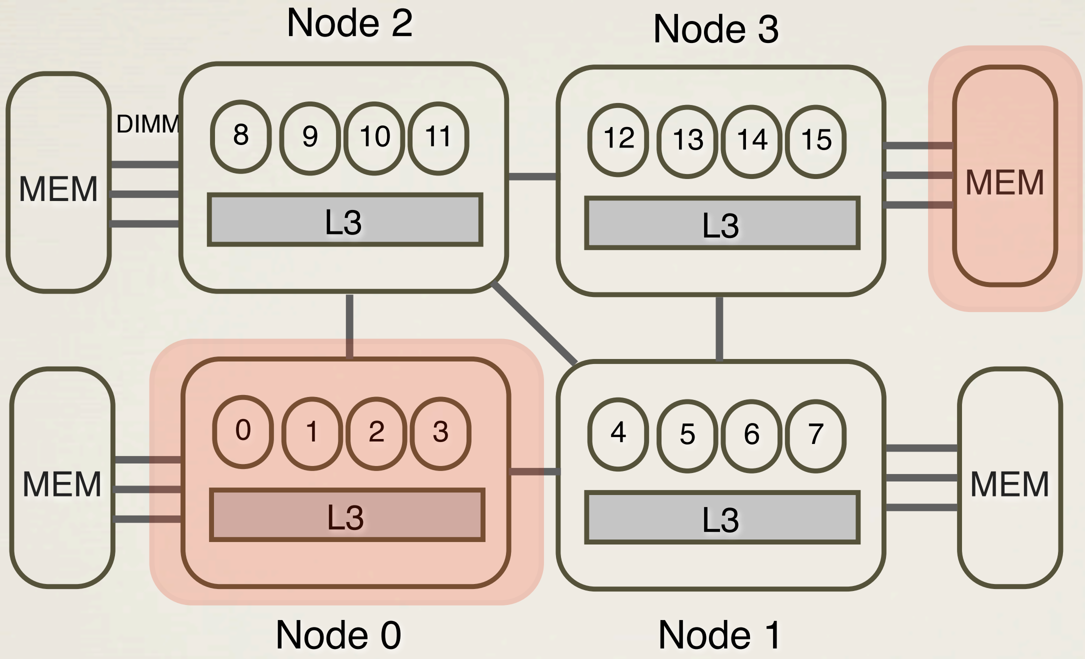
- between 0 and 1.
- allows low overhead profiling

NUMA Score: Example



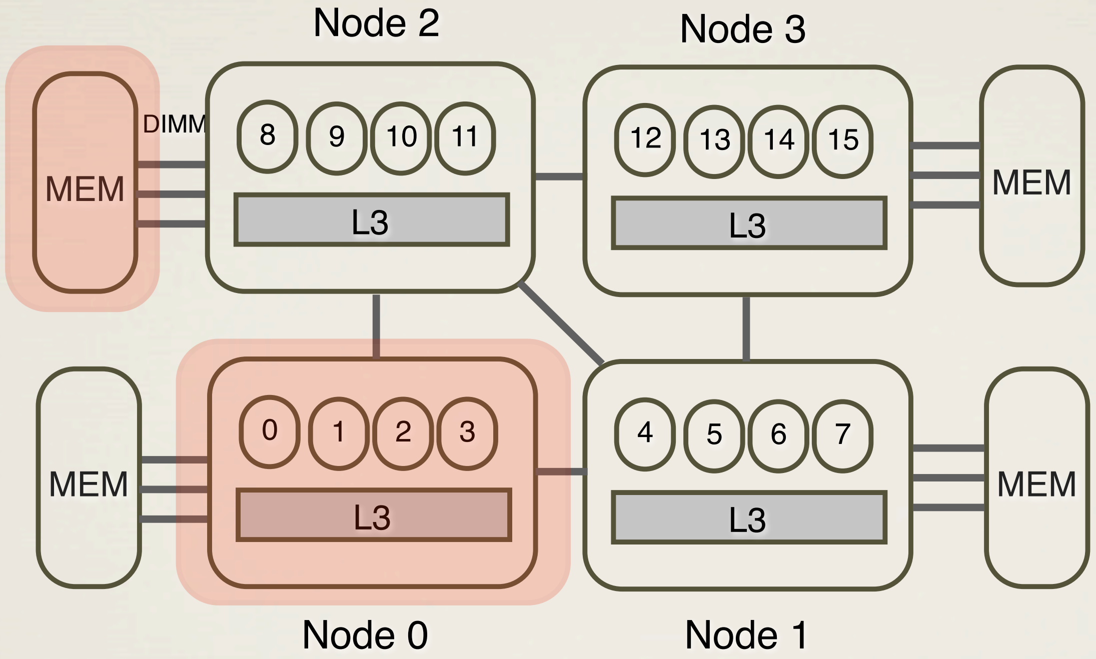
- 100% accesses between Node 0 and 3: 0.33
- 100% between Node 0 and 2: 0.66
- 100% local : 1

NUMA Score: Example



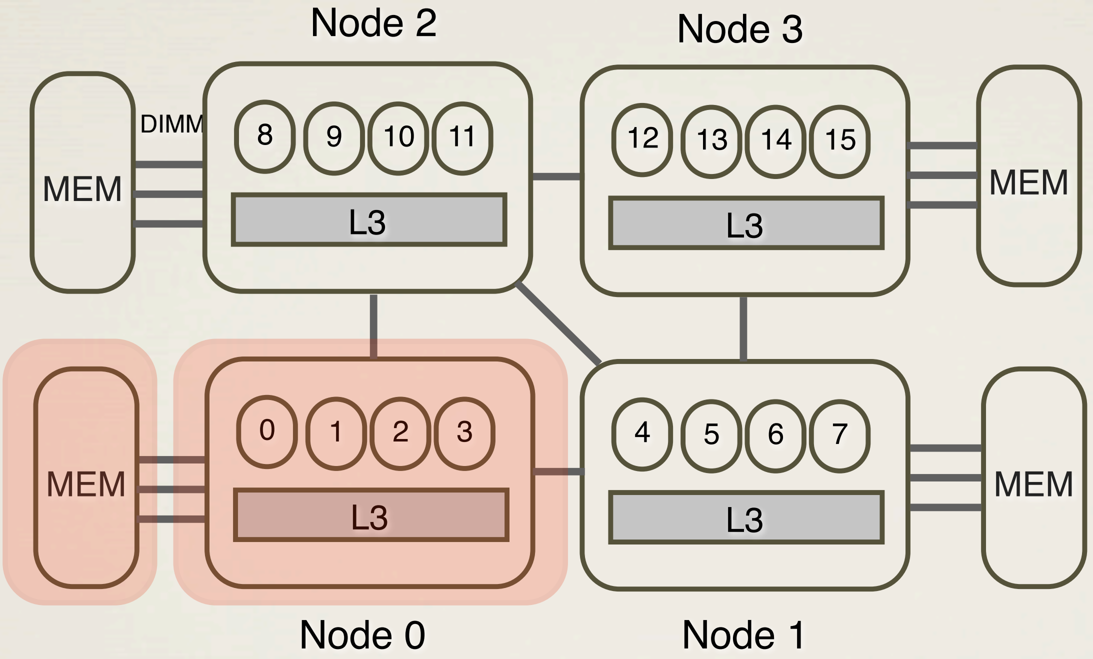
- 100% accesses between Node 0 and 3: 0.33
- 100% between Node 0 and 2: 0.66
- 100% local : 1

NUMA Score: Example



- 100% accesses between Node 0 and 3: 0.33
- 100% between Node 0 and 2: 0.66
- 100% local : 1

NUMA Score: Example



- 100% accesses between Node 0 and 3: 0.33
- 100% between Node 0 and 2: 0.66
- 100% local : 1

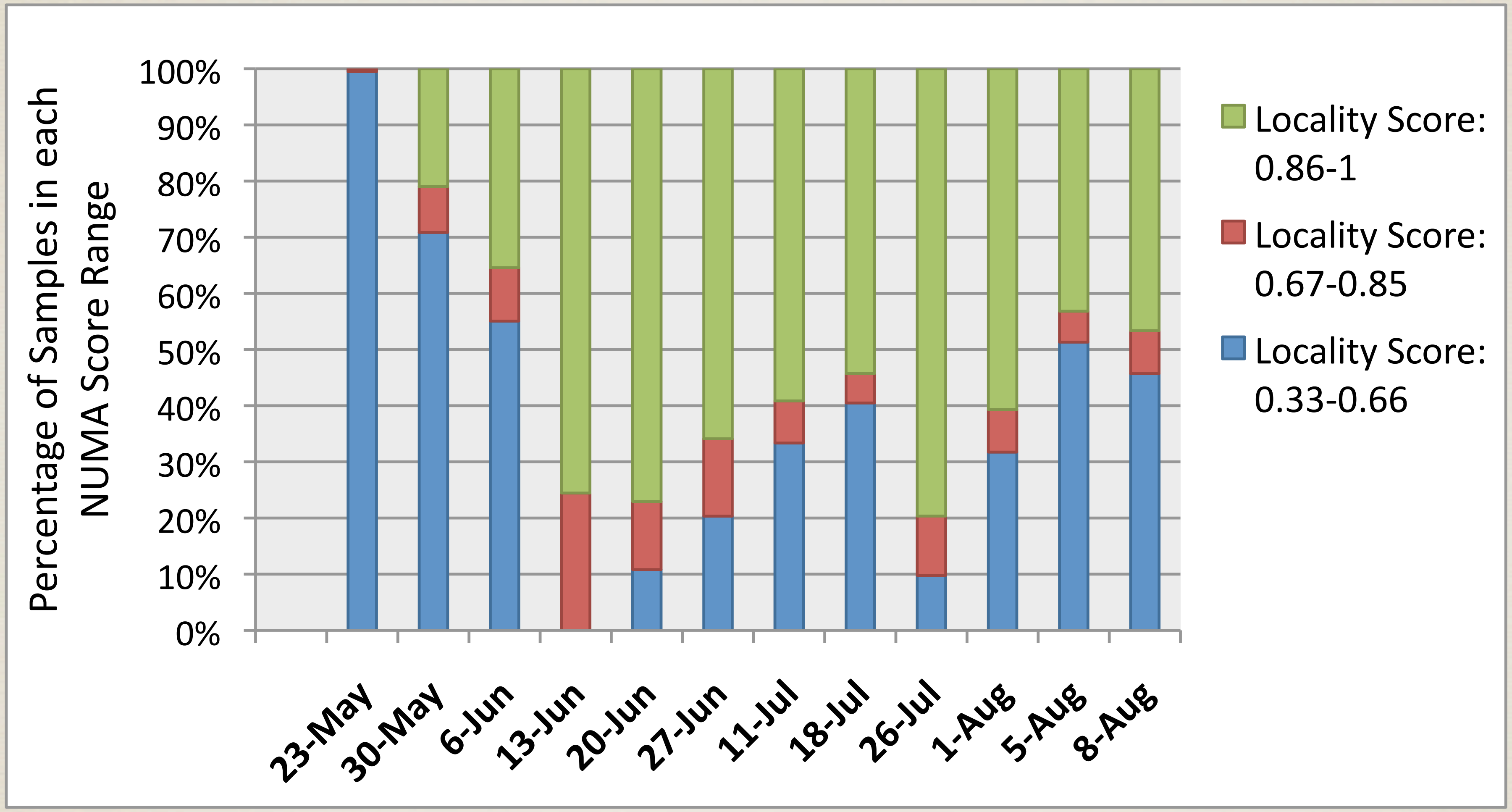
Profiling in Production

- Large-scale profiling/monitoring infrastructure in production
  - Example: Google Wide Profiling
- NUMA Score
- Performance metrics
  - CPI
  - Application-specific metrics

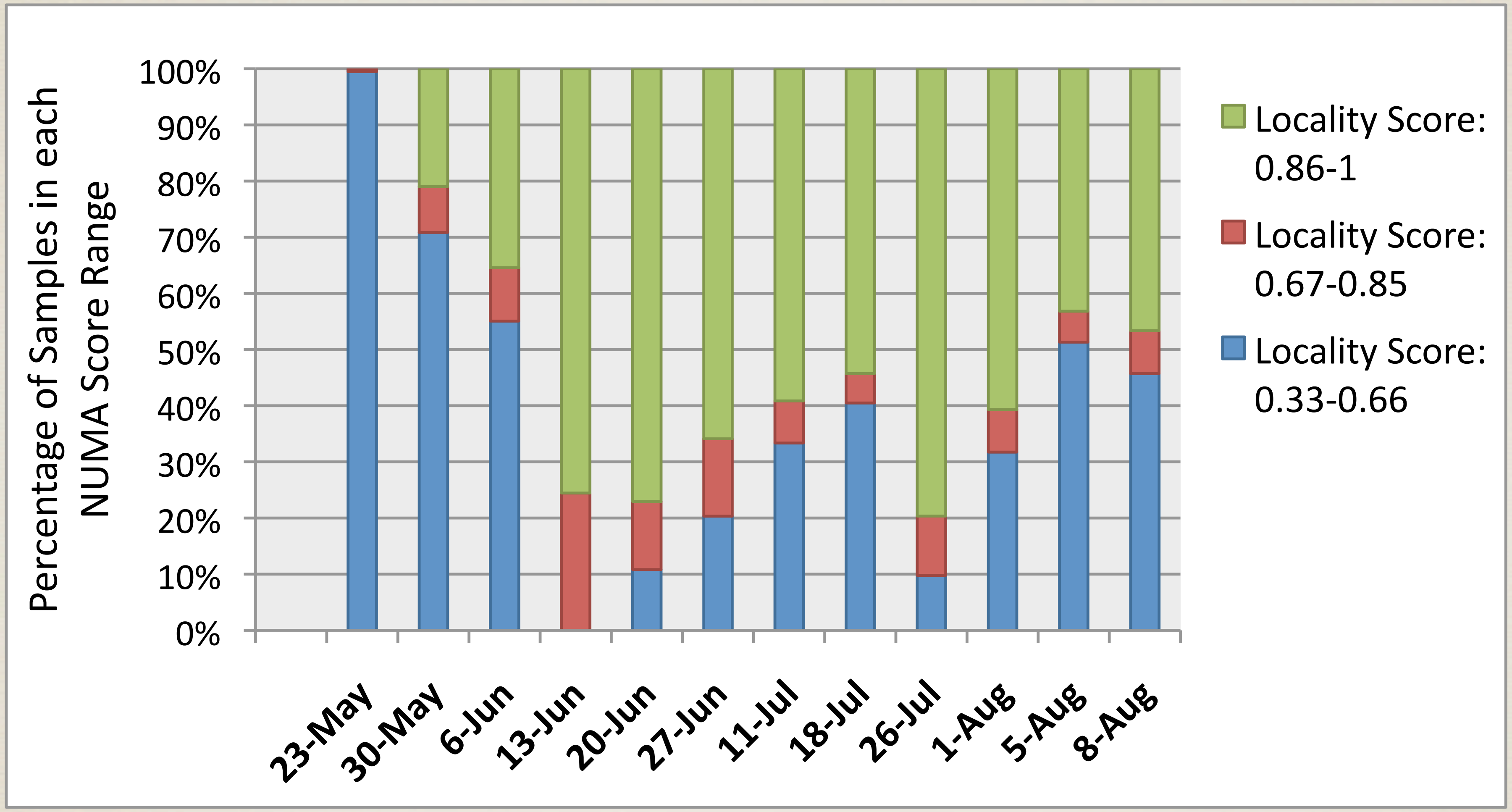
Gmail Backend

- Sticky service
- Running in co-located clusters
- Global datacenters
- Load balancer migrates user accounts
- Load fluctuates

NUMA Score Distribution

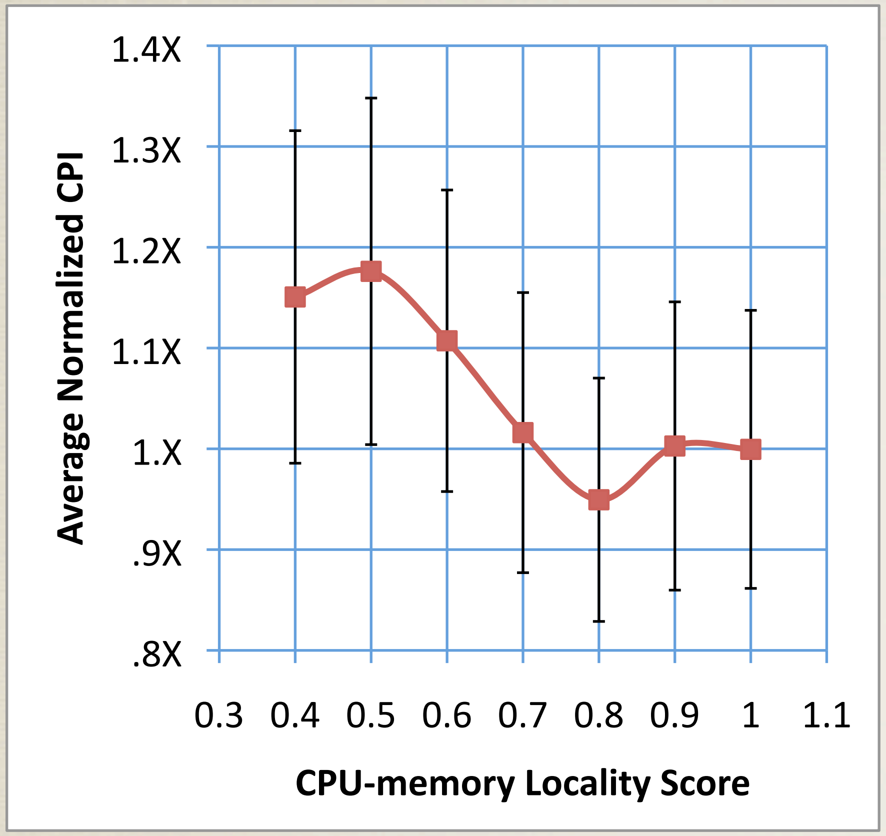


NUMA Score Distribution

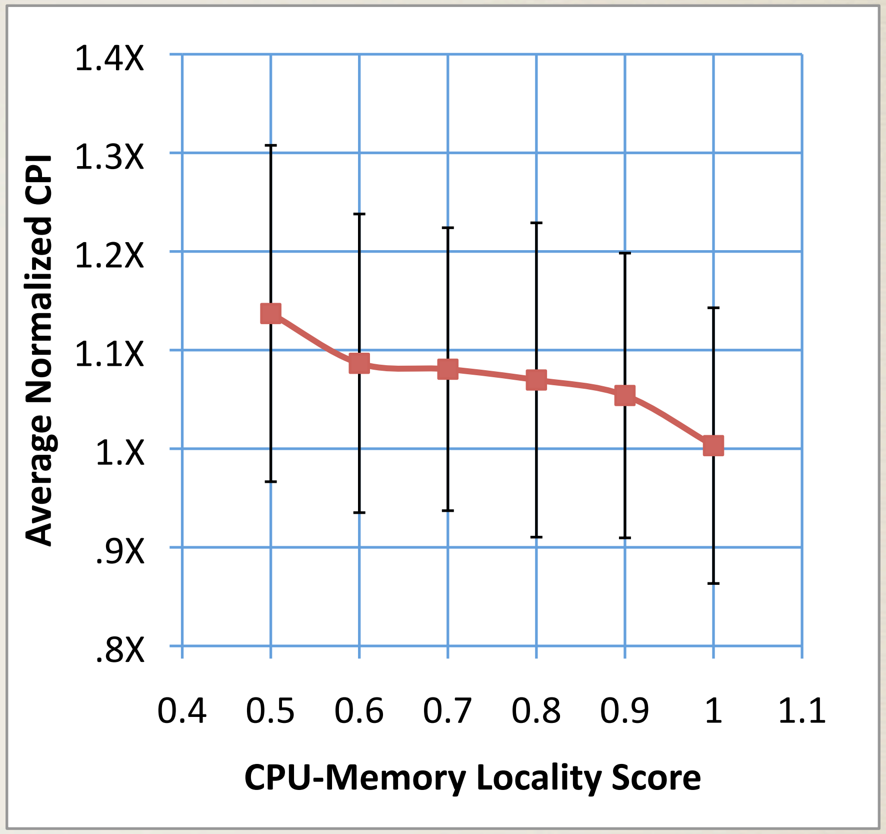


for a significant amount (often more than 50%) of jobs, all memory accesses are at least 1 hop away.

Gmail Backend



CPI vs. NUMA score 05/30.

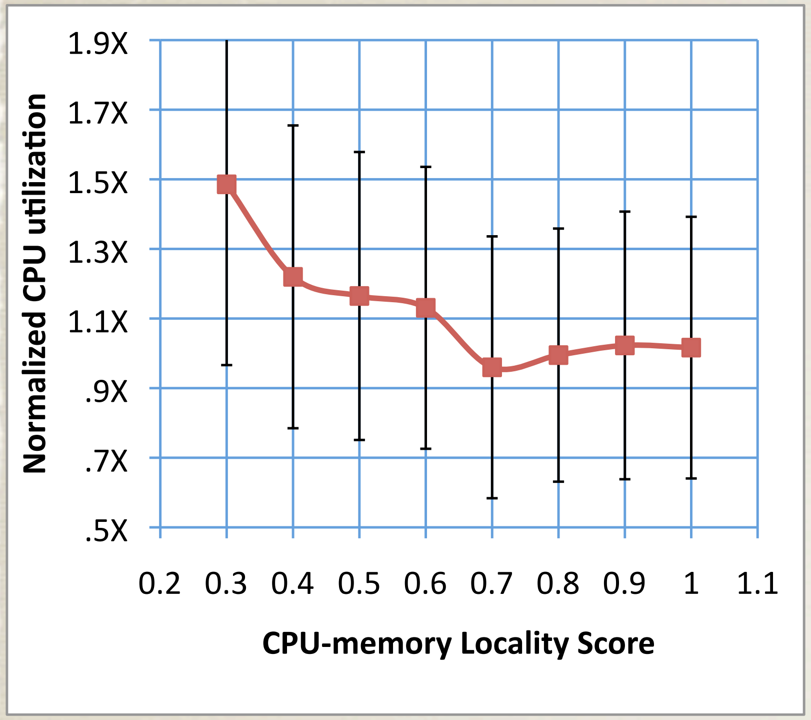


CPI vs. NUMA score on 06/20.

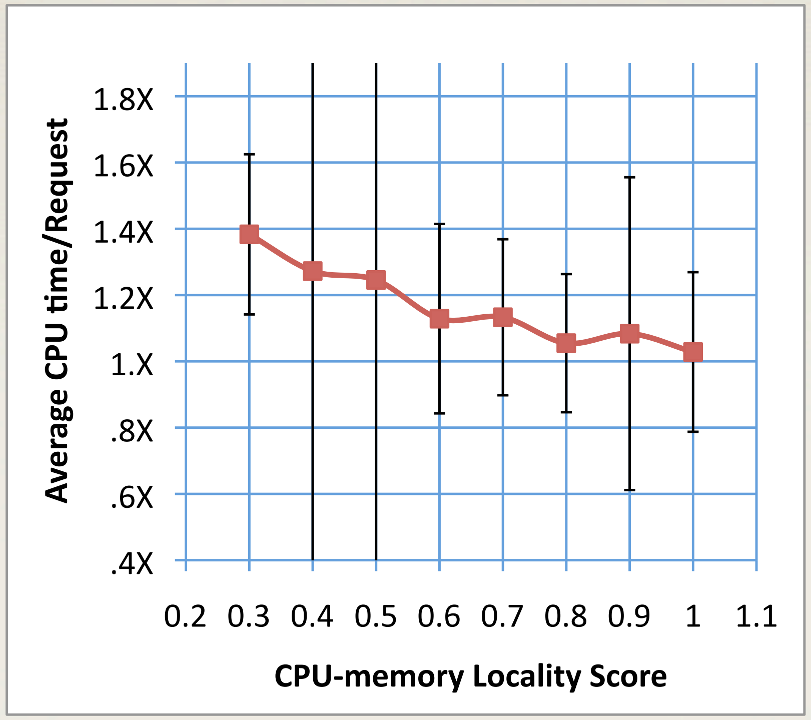
Better NUMA score correlates with lower CPI.

10-20% performance swing

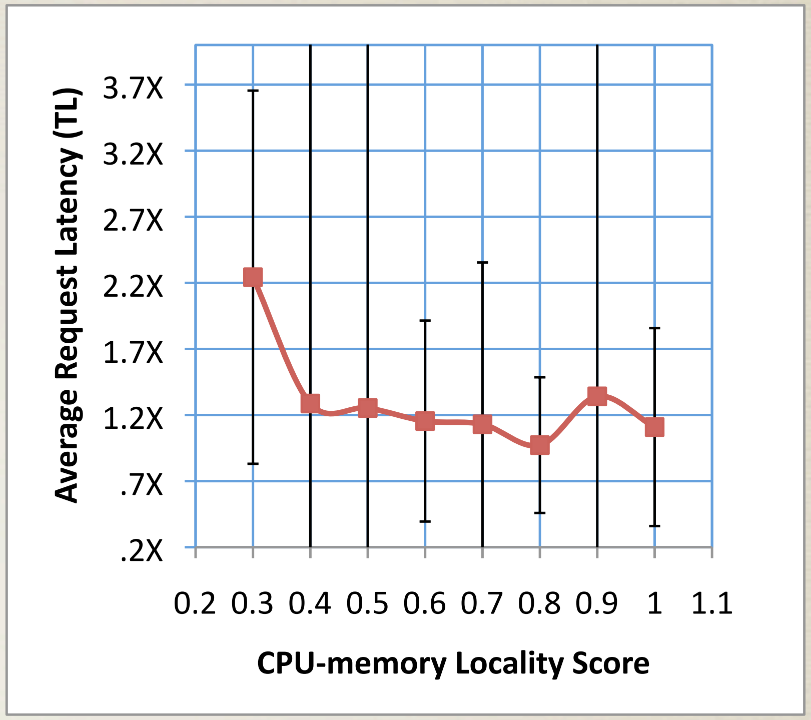
Gmail Backend



CPU utilization vs. NUMA



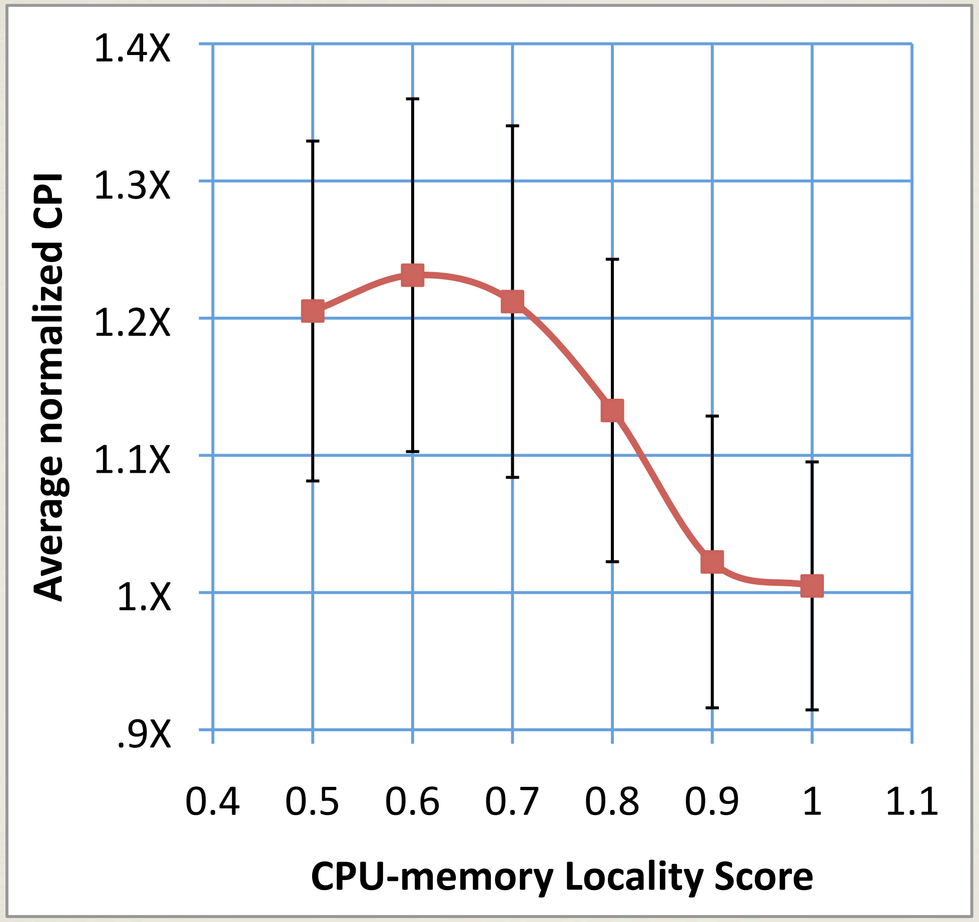
CPU time/request vs. NUMA



Request Latency (threadlist) vs. NUMA

- Better NUMA score correlates with lower CPU utilization.
- Noisy data for request latency and CPU/request

Websearch Frontend



CPI vs. NUMA score

Better NUMA score correlates with lower CPI.

~20% performance swing

Methodology

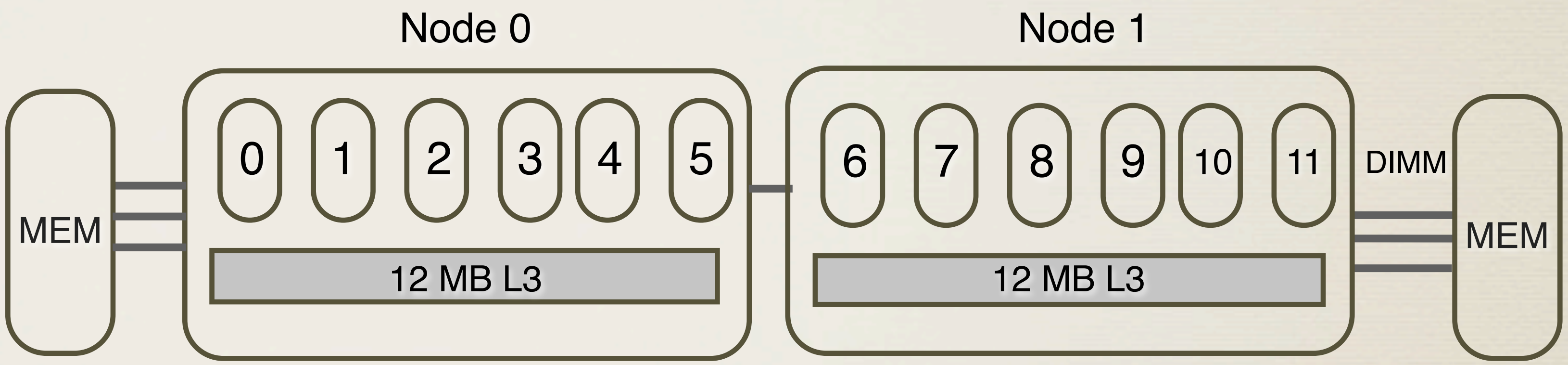
- 2-phase Methodology
  - Production study in the wild
  - Single-node load-test in the controlled environment

Load Test on Single Server

- Tradeoffs between **memory access locality** and the impact of **cache sharing/contention** on a CMP machine

Load Test on Single Server

- Tradeoffs between **memory access locality** and the impact of **cache sharing/contention** on a CMP machine



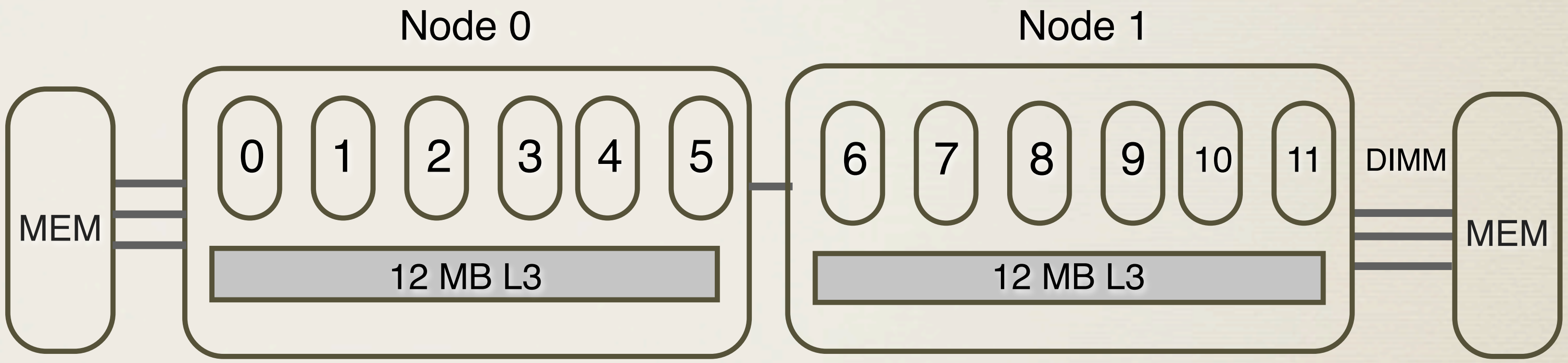
X solo:

1. 100% Local access, sharing 1 LLC
2. 50% Local access, sharing 2 LLCs
3. 0% Local access, sharing 1 LLC

	0	1	2	3	4	5	6	7	8	9	10	11
M-X	X	X	X	X	X	X						
M-X	X	X	X				X	X	X			
M-X							X	X	X	X	X	X

Load Test on Single Server

- Tradeoffs between **memory access locality** and the impact of **cache sharing/contention** on a CMP machine



X solo:

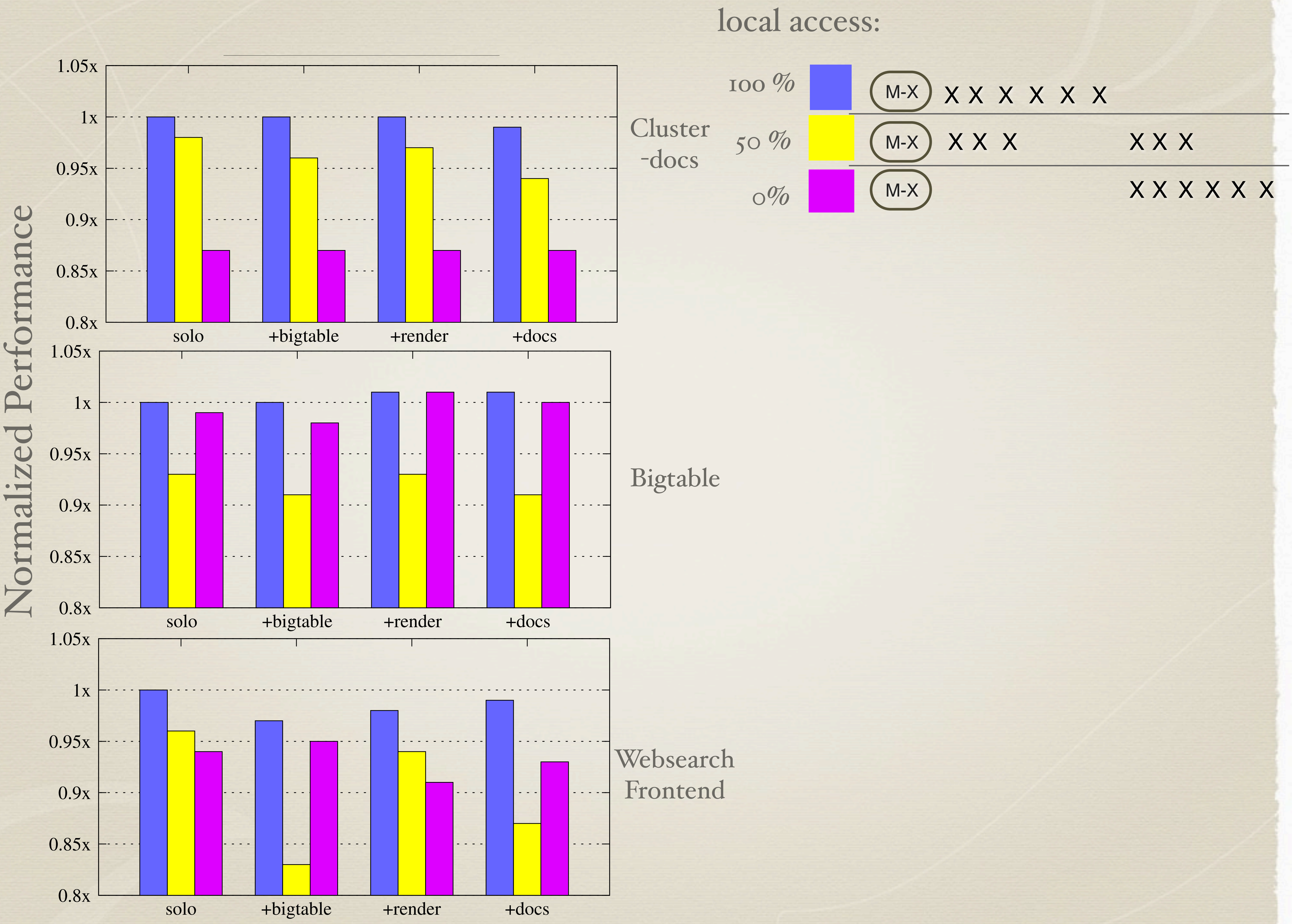
1. 100% Local access, sharing 1 LLC
2. 50% Local access, sharing 2 LLCs
3. 0% Local access, sharing 1 LLC

M-X	X	X	X	X	X	X							
M-X	X	X	X				X	X	X				
M-X							X	X	X	X	X	X	

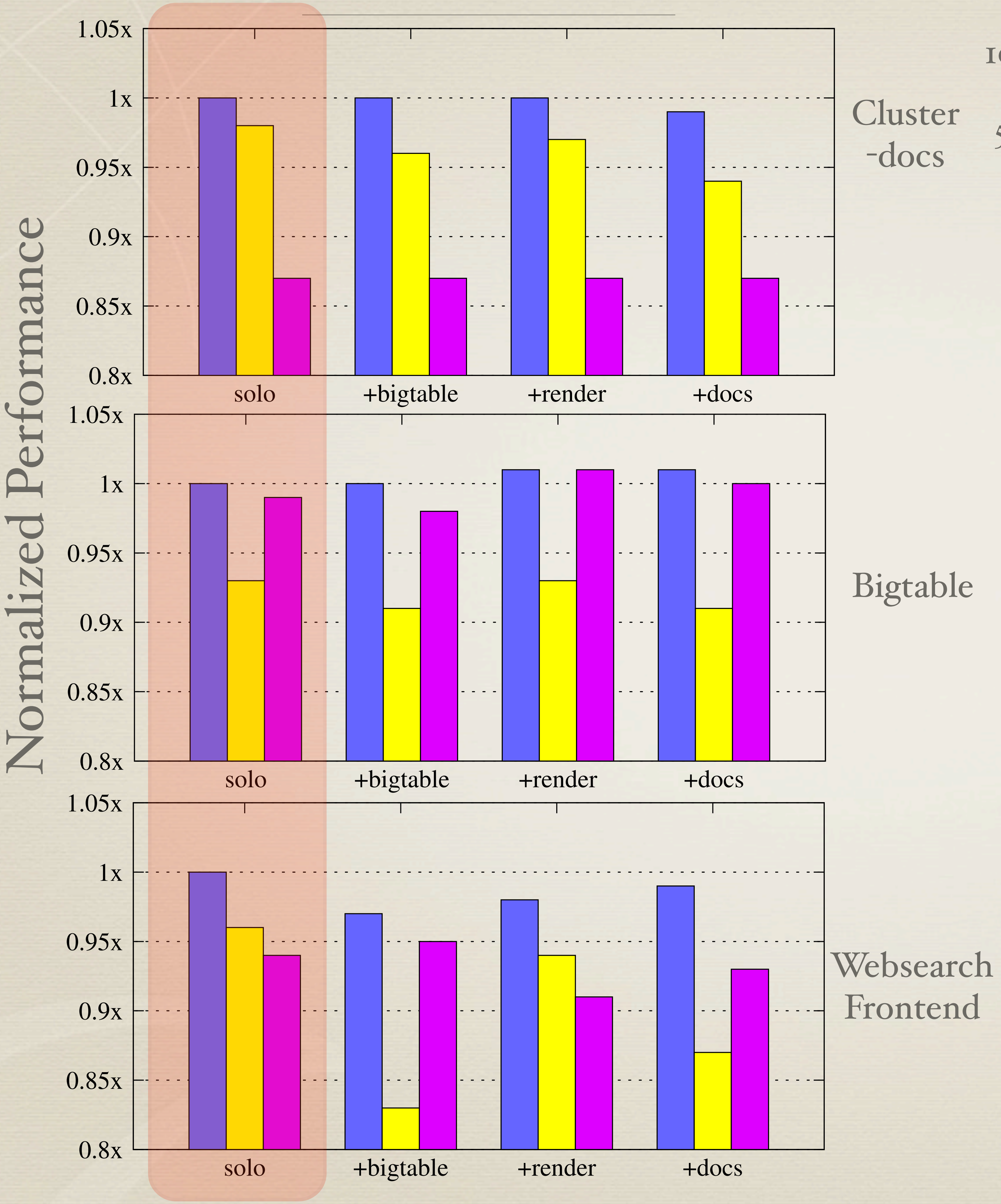
X coruns w/ Y:

4. 100 % Local access, sharing LLC w/ sibling
5. 50 % Local access, sharing LLC w/ Y
6. 0 % Local access, sharing LLC w/ sibling

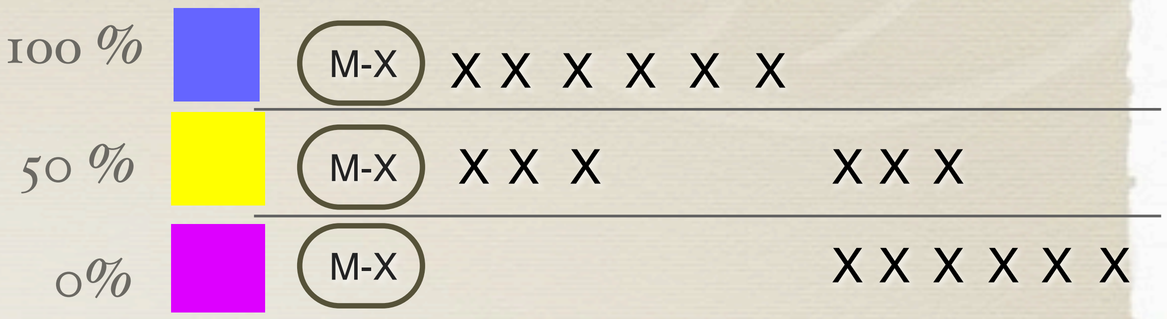
M-X	X	X	X	X	X	X	Y	Y	Y	Y	Y	Y	M-Y
M-X	X	X	X	Y	Y	Y	X	X	X	Y	Y	Y	M-Y
M-X	Y	Y	Y	Y	Y	Y	X	X	X	X	X	X	M-Y

local access:
100 %
M-X
X X X X X X
50 %
M-X
X X X
X X X
0%
M-X
X X X X X X
Normalized Performance
1.05x
1x
0.95x
0.9x
0.85x
0.8x
solo
+bigtable
+render
+docs
Cluster -docs
Bigtable
Websearch Frontend

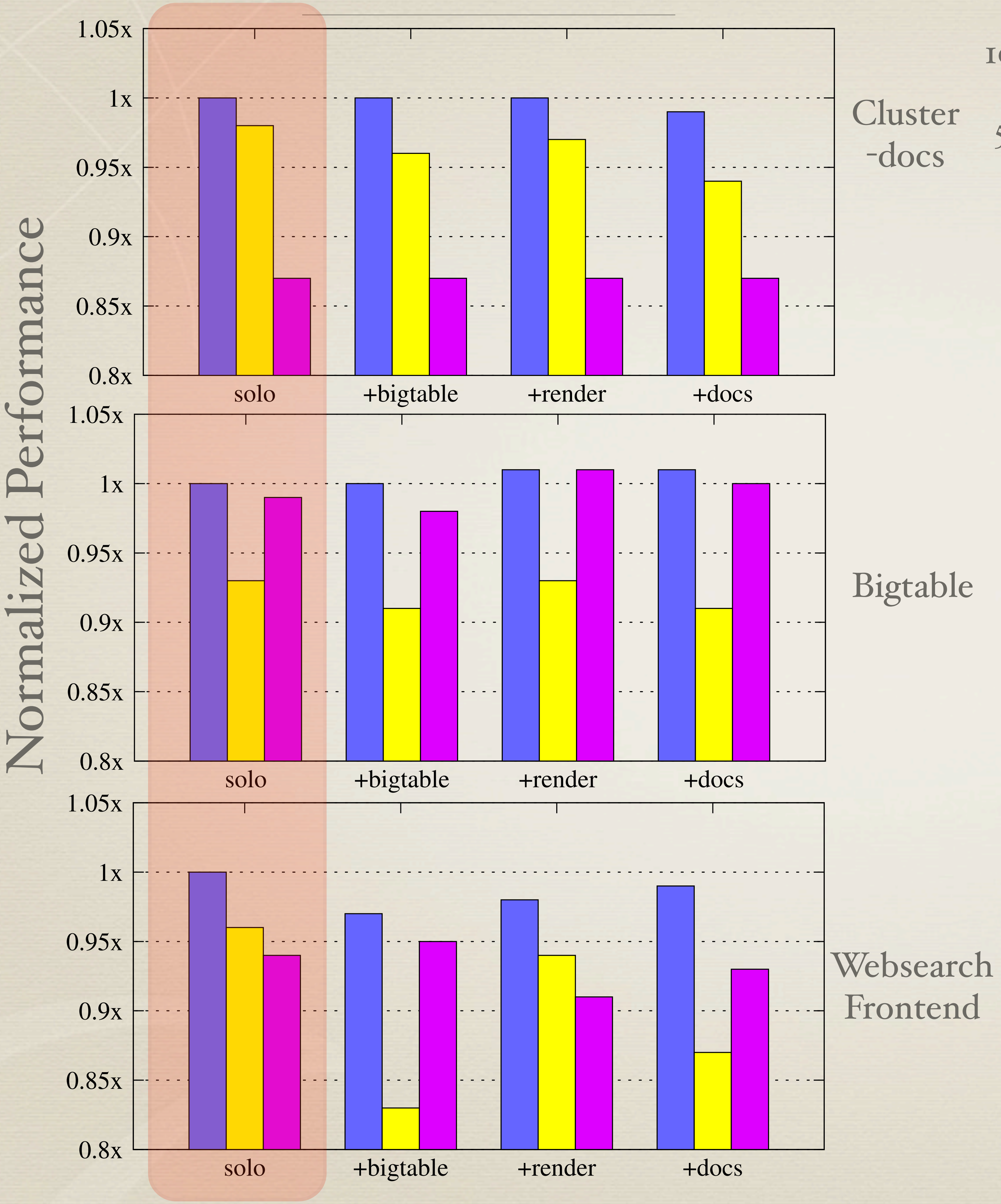
Solo



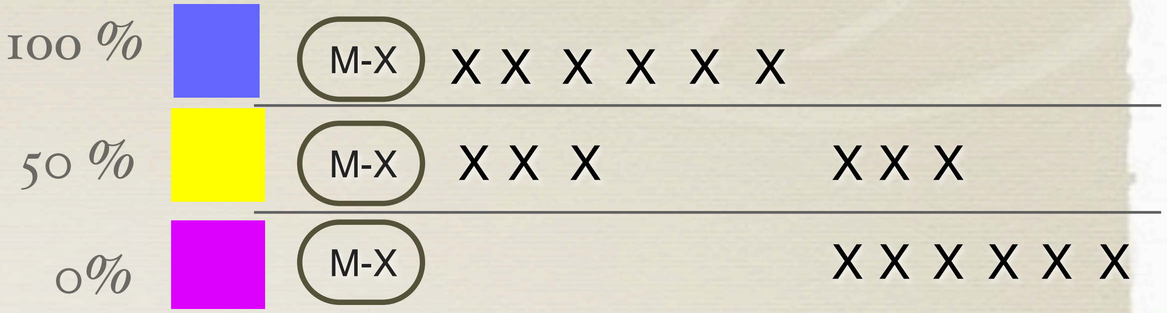
local access:



Solo

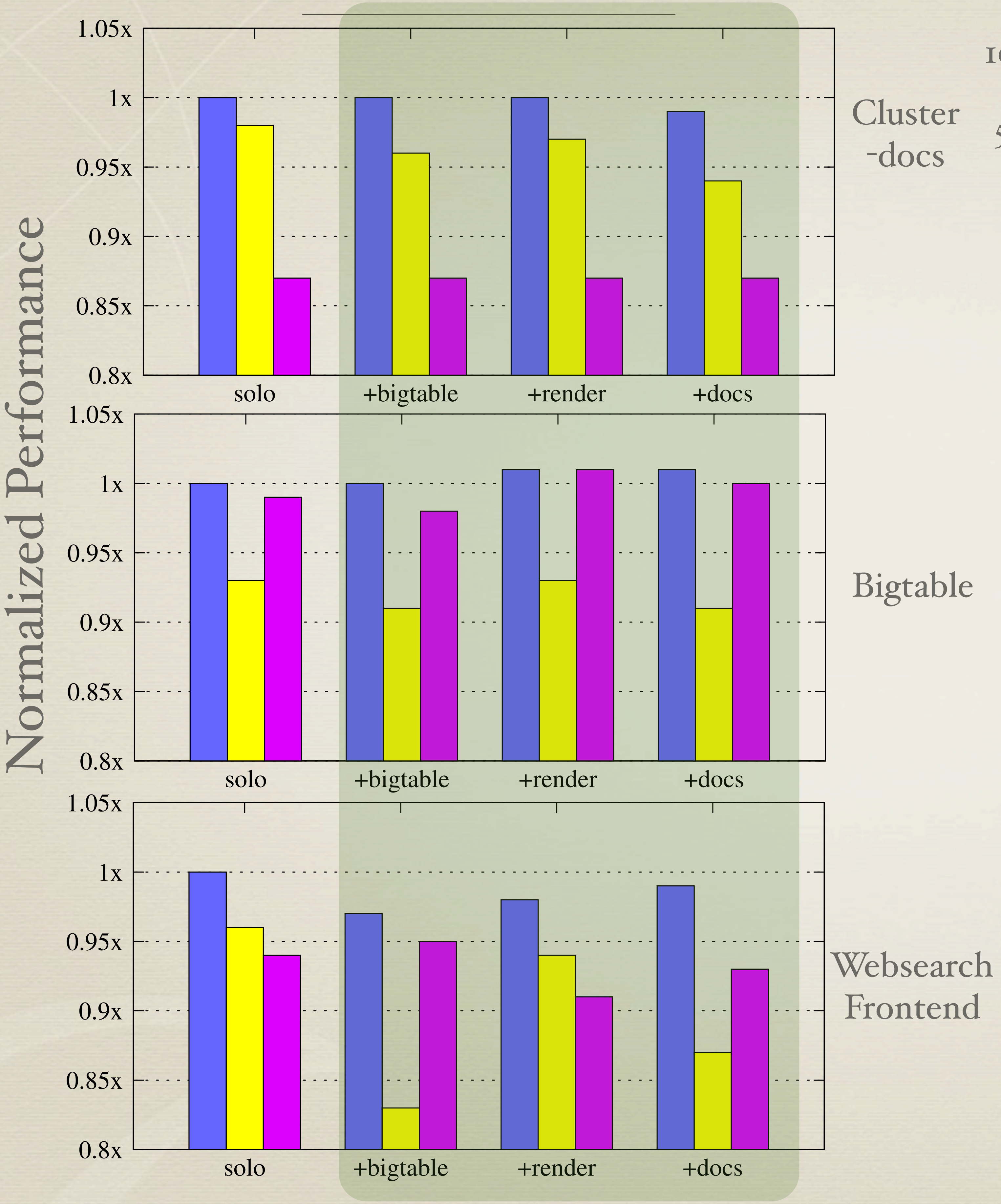


local access:

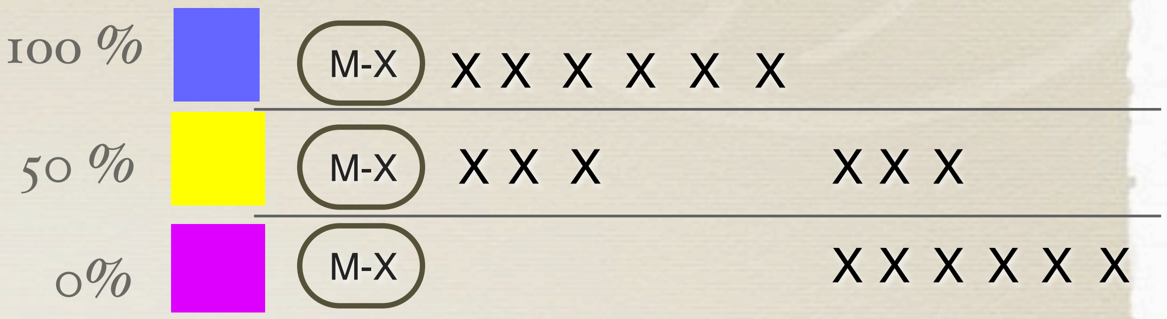


- Solo:
  - bigtable 0% local access outperform 50% local access

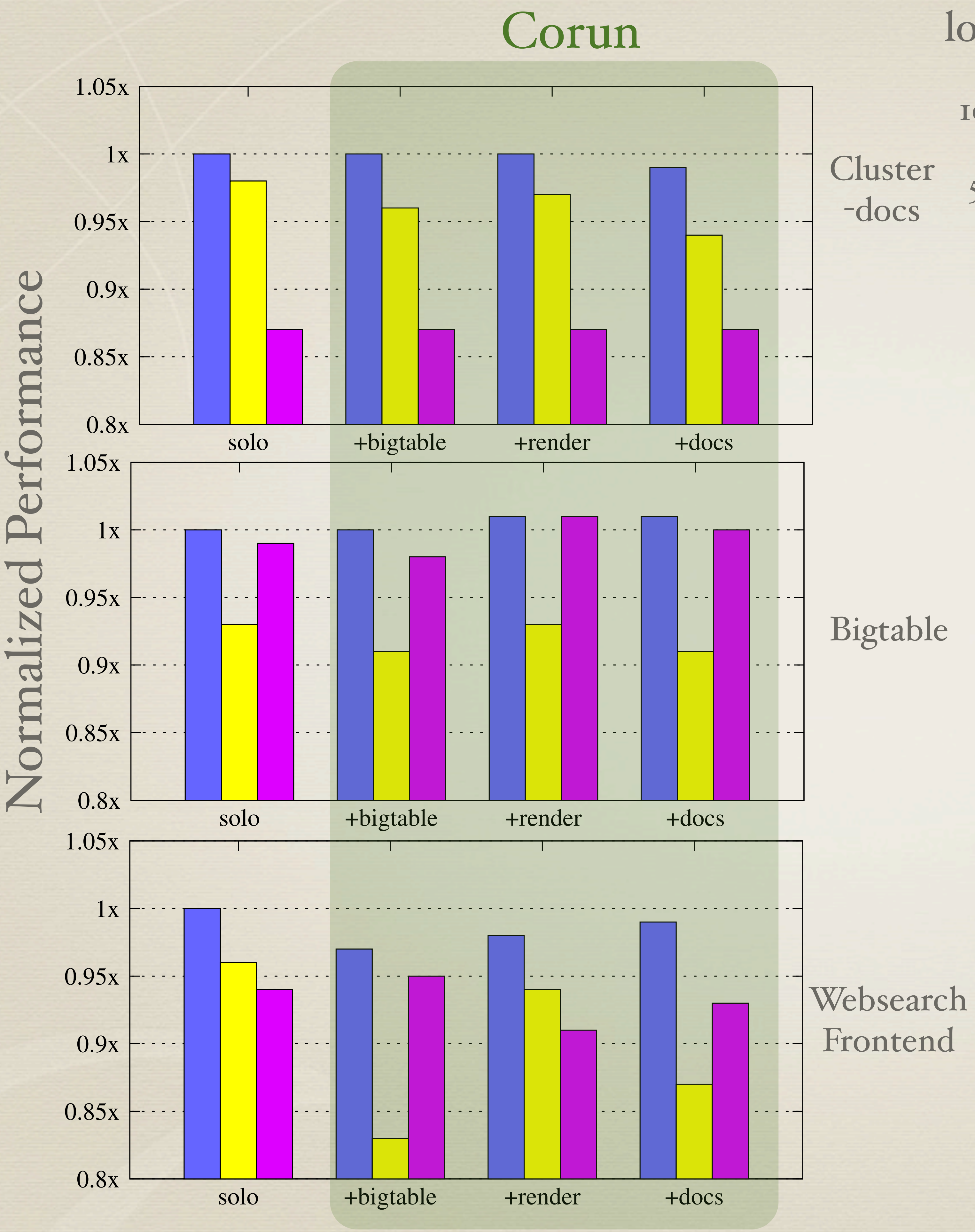
Corun



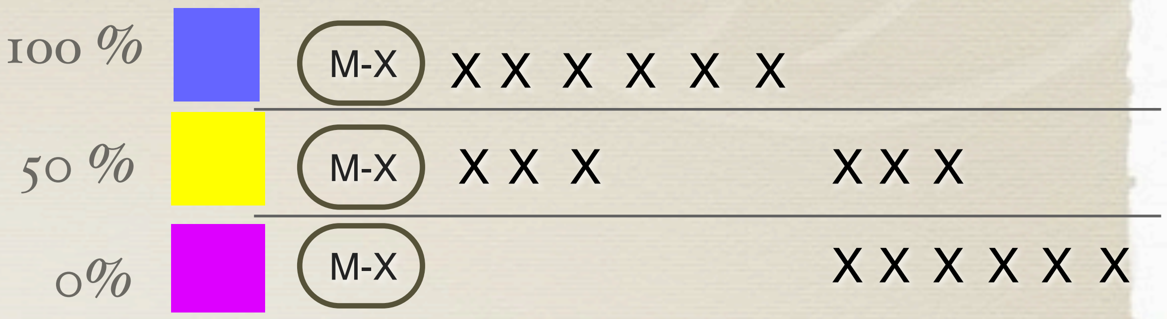
local access:



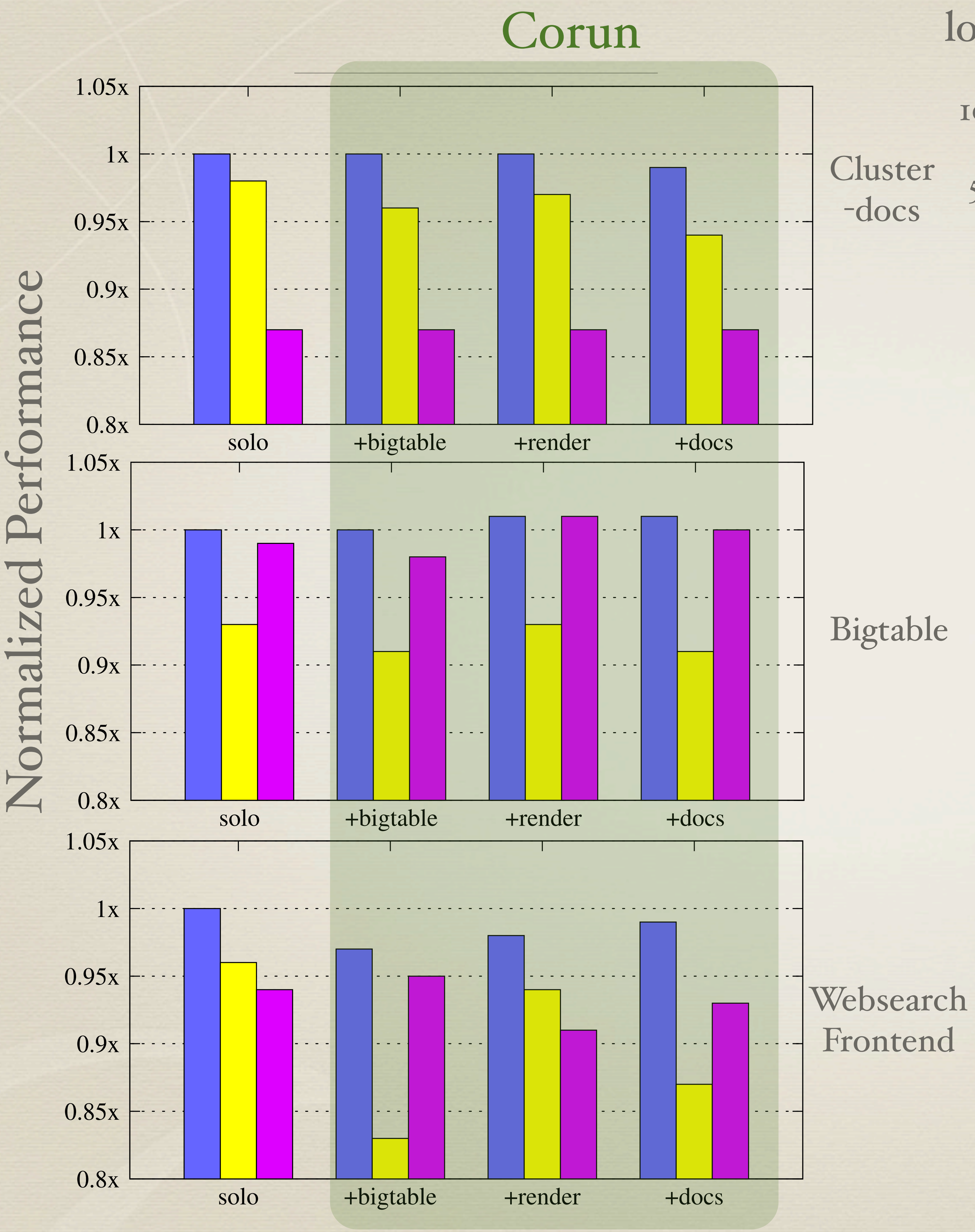
- Solo:
  - bigtable 0% local access outperform 50% local access

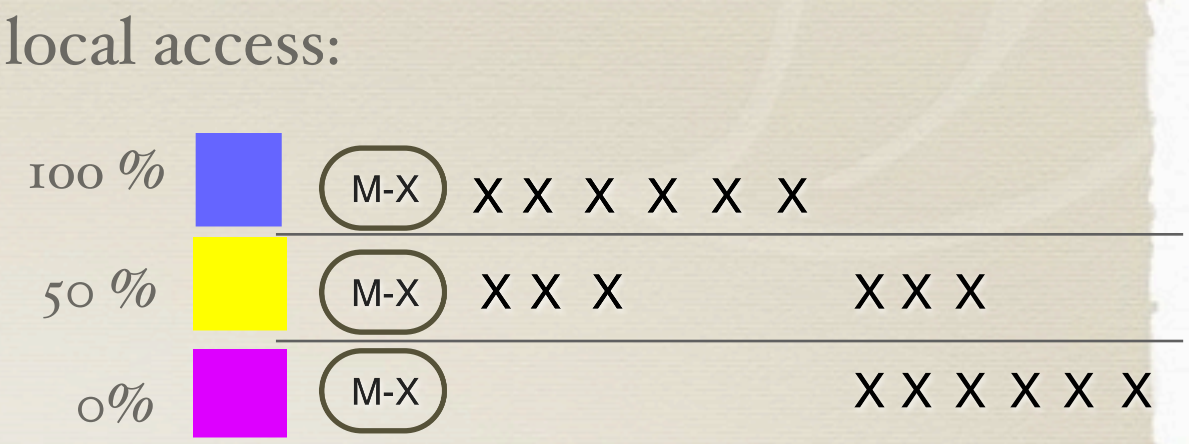


local access:



- Solo:
  - bigtable 0% local access outperform 50% local access
- Corun:
  - websearch: depends on the corunner, the performance ranking changes



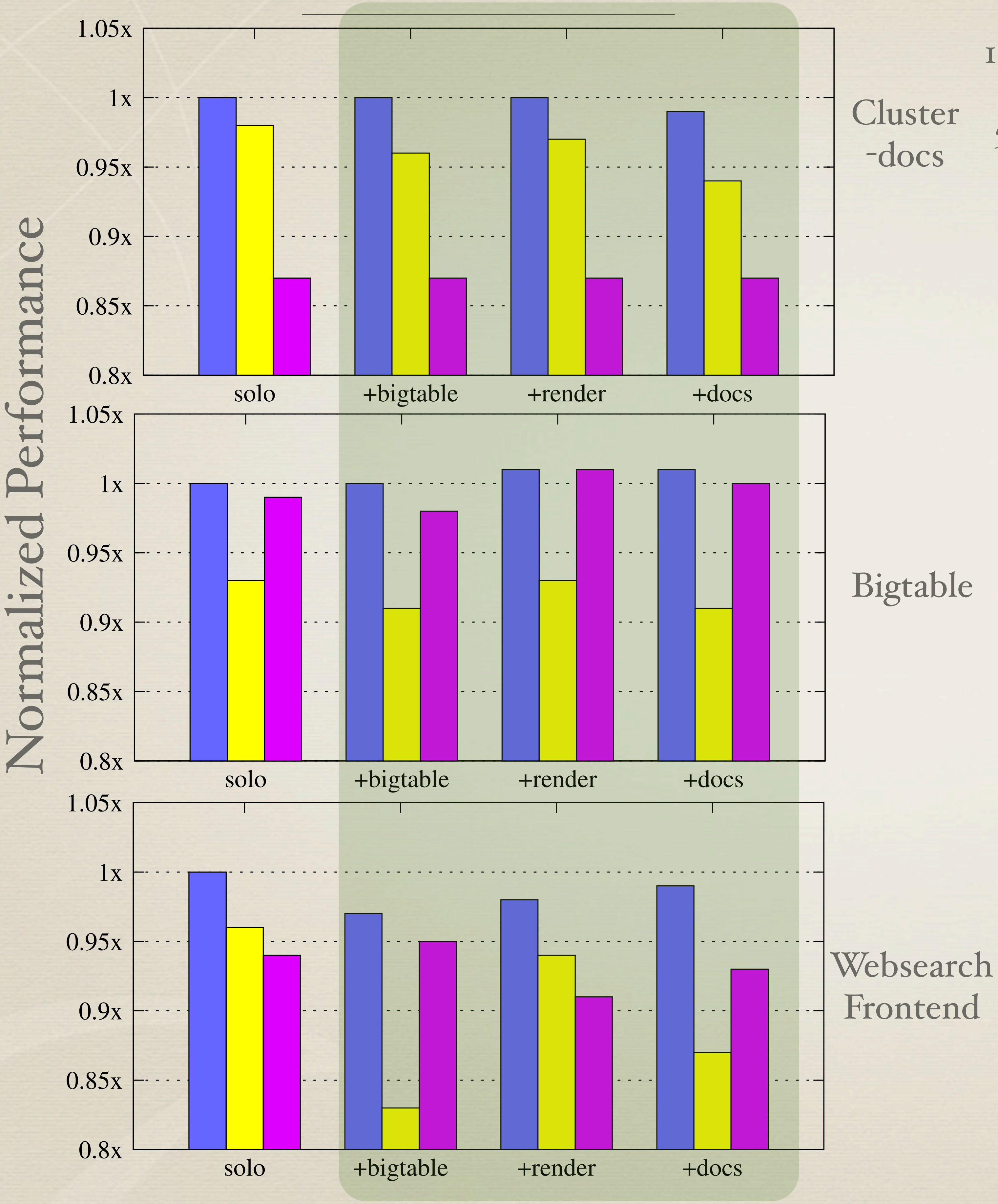


- Solo:
  - bi
    ac
    5

tradeoffs b/t NUMA and cache sharing/ contention

- Corun:
  - websearch: depends on the corunner, the performance ranking changes

Corun



local access:

100 % 🟦 M-X X X X X X X

50 % 🟨 M-X X X X X X X

0% 🟪 M-X X X X X X X

- Solo:
  - bi ac 5c

tradeoffs b/t NUMA and cache sharing/ contention

- Corun:
  - w o p c

varies for different applications and when the application's corunner changes.

Conclusion

- Combine production study and controlled study
- Production study
  - novel NUMA score
  - lightweight monitoring of large scale systems
  - careful correlation and analysis of noisy data.
  - conclusion: performance impact of NUMA is significant for large scale web-service applications
- Controlled study
  - Conclusion: some running scenarios with more remote memory accesses may outperform scenarios with more local accesses
  - This tradeoff b/t NUMA and cache sharing/contention varies for different applications and when the application's corunner changes.

when
how
who
why
which
where
what
how
when
which
why
what
how
which
who
when
why
when
where
which
what
how
who
why
how
when
how
where
who
which
how
when
which
how
when
what
why
how
how
which
when
which
how
which
who
why
when
what

- 1% performance improvement means millions
- Failure to tease out individual micro-architectural properties -> difficult to quantify the performance impact and potential optimization benefit
- Leave performance opportunity on the table