Optimizing Datacenter Power with Memory System Levers for Guaranteed Quality-of-Service

UTAH ARCH *

intel §

Kshitij Sudan*

Sadagopan Srinivasan§

Rajeev Balasubramonian*

Ravi Iyer§

PACT-2012, 9/20/2012

Executive Summary

- Goal: Co-schedule ***N*** applications on ***S*** servers
  - **Minimize** ***S*** while observing QoS guarantees
- Approach: Limit hardware resources for applications ***just enough*** to meet their QoS
- Mechanism: Use ***memory system levers*** to limit resources
  - *Simple and effective because memory is a bottleneck*

UTAH ARCH

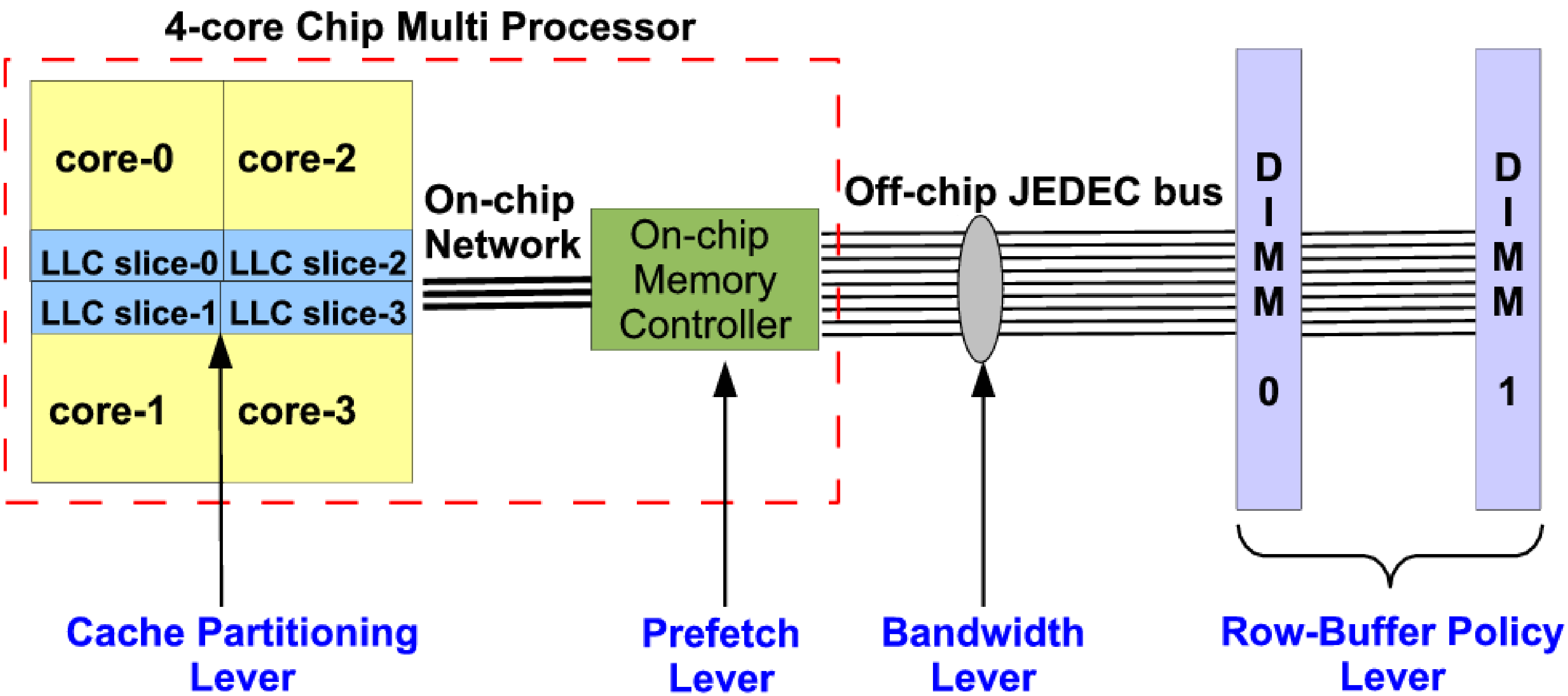
Motivation

- Co-locating applications at datacenters reduces costs
- While co-locating, application QoS guarantees need to be maintained
- Datacenter operators co-locate conservatively
  - Leads to server under-utilization
- Determining efficient co-location on-the-fly is a complex resource allocation problem

UTAH ARCH

intel

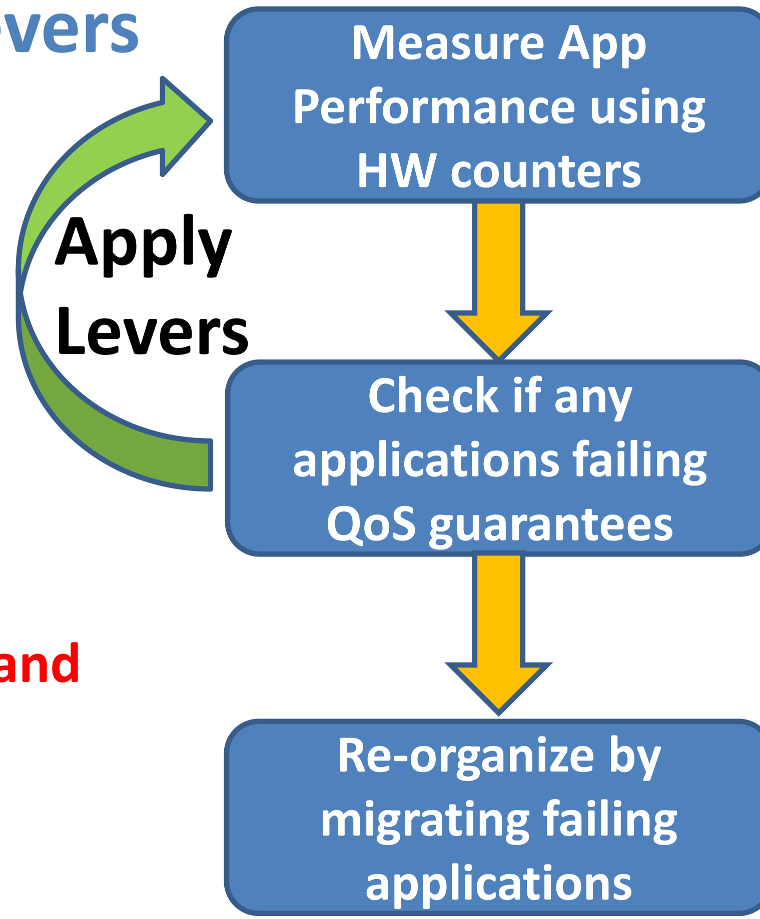
Memory System Levers



Each ***lever*** can have multiple ***levels*** (low, medium, high)

Problem of “Plenty”

- **Many levers, many levels of levers**
  - When to apply which lever?
  - At what strength?
- ***Measure-check-reorganize***
  - ***Measure*** application perf.
  - ***Check*** if all applications meeting their QoS
    - If not, **systematically apply levers and levels**
  - ***Re-organize*** (migrate) if all lever applications fail



Sufferer Applications

- ***Check*** phase identifies applications failing QoS
  - These are the ***sufferer*** applications
- Apply levers to ***limit*** resources for ***non-suffering*** applications
  - Limiting resources is simpler to implement
- Start applying levers to non-suffering applications with largest headroom between current performance and QoS

UTAH ARCH

Methodology

- Trace-based, 4-core, OoO CMP platform
  - x86, 128 KB private L1 cache, 2 MB shared L2
  - 4 GB DDR2 DRAM main memory
- Four enterprise applications
  - SAP, TPC, SJAS, SJBB

Application	LLC MPKI	DRAM access/Kilo-inst.
TPC	8.23	13.25
SJAS	4.77	9.49
SJBB	3.33	4.79
SAP	1.91	3.71

Methodology

- 12 workload mixes
  - created by combining application traces of varying memory-intensity and QoS aggressiveness

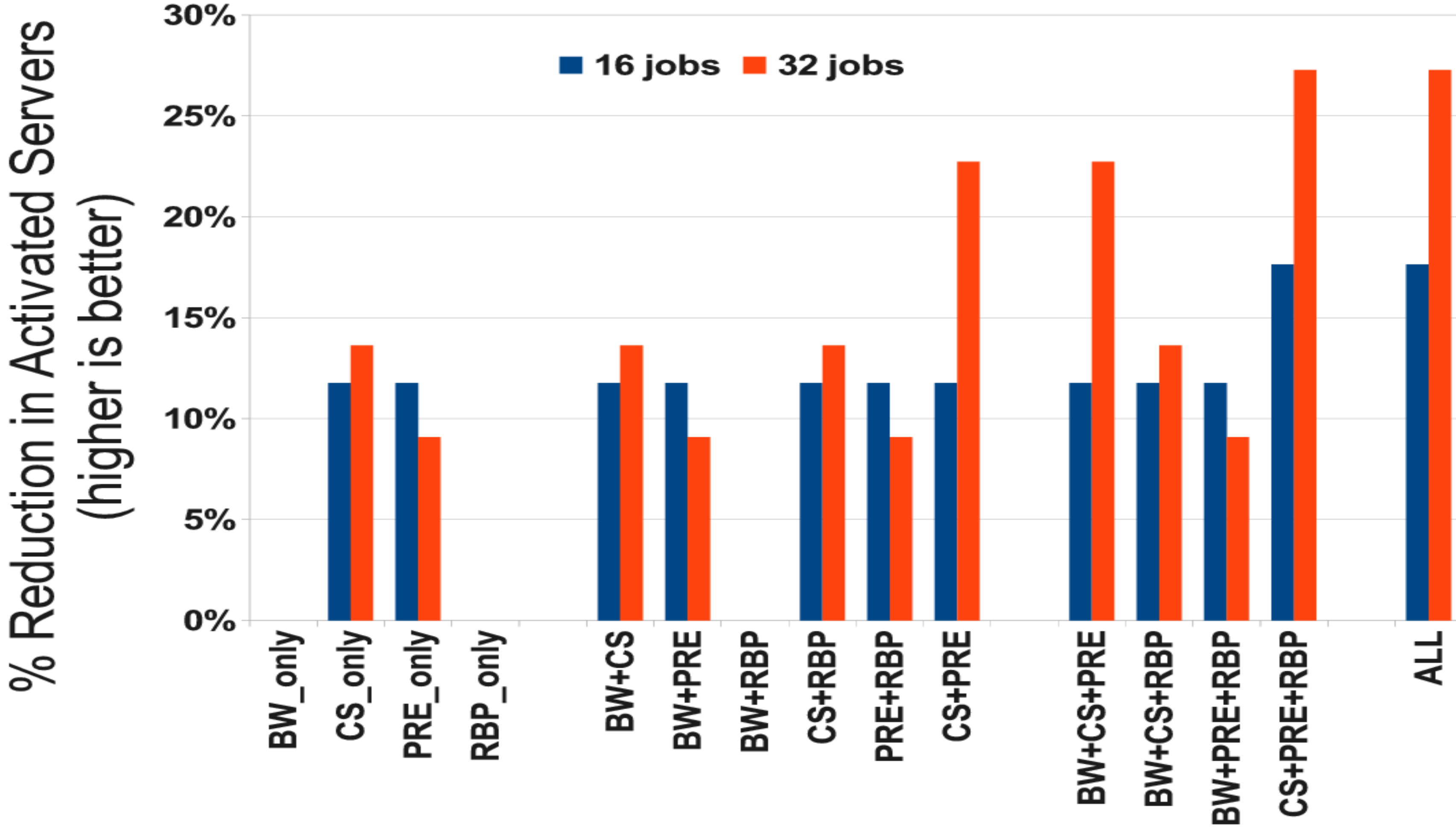
Memory Intensive (MI)	TPC
Balanced (B)	TPC, SAP, SJAS, SJBB
Non-Mem. Intensive (NMI)	SAP

QoS Level	Fraction Of Baseline IPC
Relaxed	[0.66, 0.77]
Medium	(0.77, 0.88]
Aggressive	(0.88, 1.0]
Mix	[0.66, 1.0]

Lever Effectiveness

- Application characteristics and available resources dictate lever effectiveness
- For enterprise applications, effectiveness typically follows the following trend:
    1. Shared cache capacity (**CS**)
    2. Pre-fetching memory requests (**PRE**)
    3. DRAM row-buffer policy (**RBP**)
    4. DRAM bandwidth (**BW**)

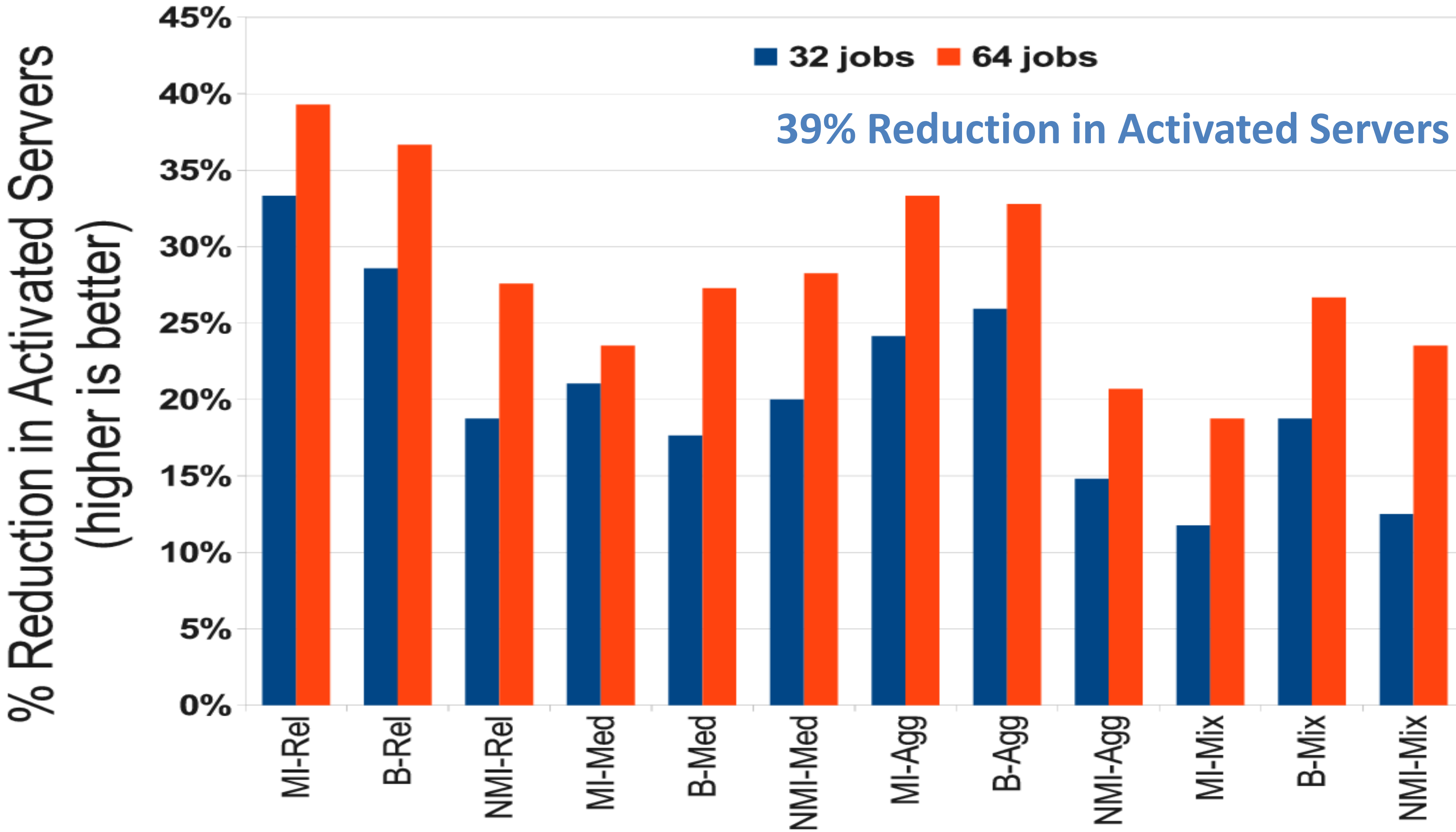
Cumulative Effect of Lever Application



Cumulative Effectiveness

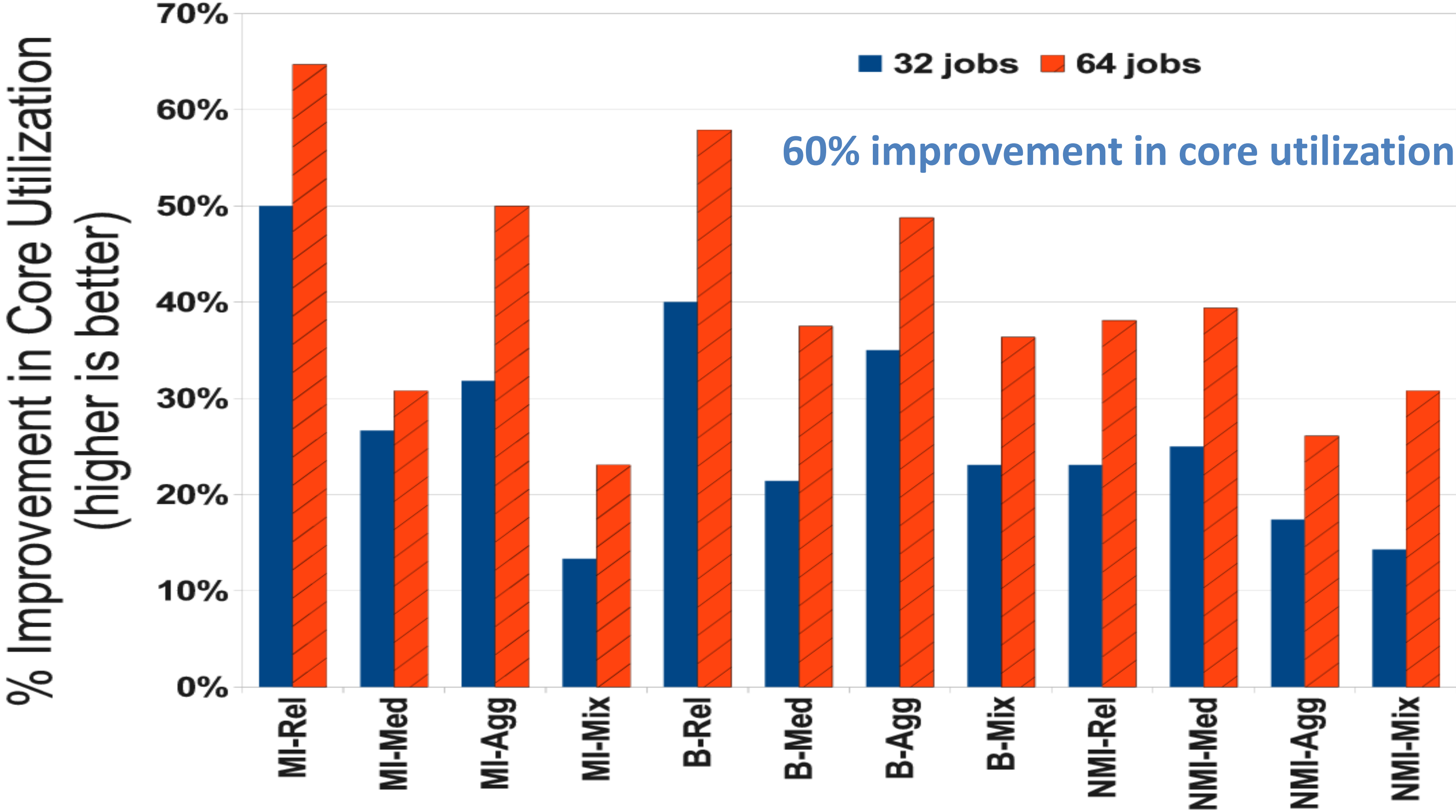
- Memory lever application has a ***cumulative effect*** in limiting resources
  - Makes lever ***and*** lever level application simple
- Example lever application
  1. Apply CS-low, -medium, -high
  2. Apply PRE-low-, -medium, -high
  3. Apply RBP-closed
  4. Apply BW-low, -medium, -high

Reduction in Activated Servers

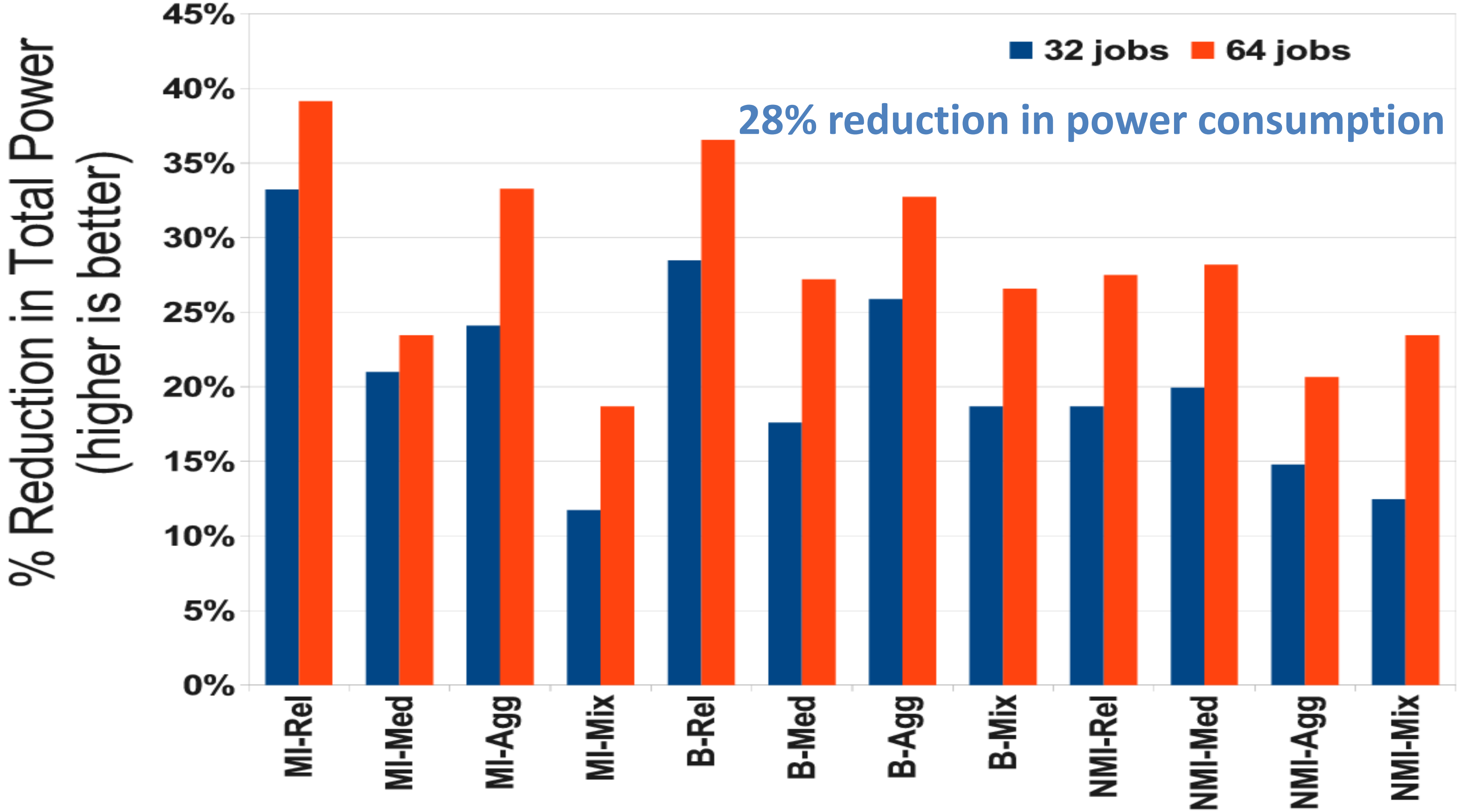


UTAH ARCH

Improvement in Core Utilization



Power Savings



UTAH ARCH

Conclusions

- **Memory system ideal for limiting resources**
  - Simple to implement, cumulative effect of levers
  - For enterprise applications, lever effectiveness follows the order **CS** → **PRE** → **RBP** → **BW**
  - **Number of activated servers** reduced by **39%** ↓
  - **Core utilization** increases by **60%** ↑
  - **Power consumption** reduces by **28%** ↓

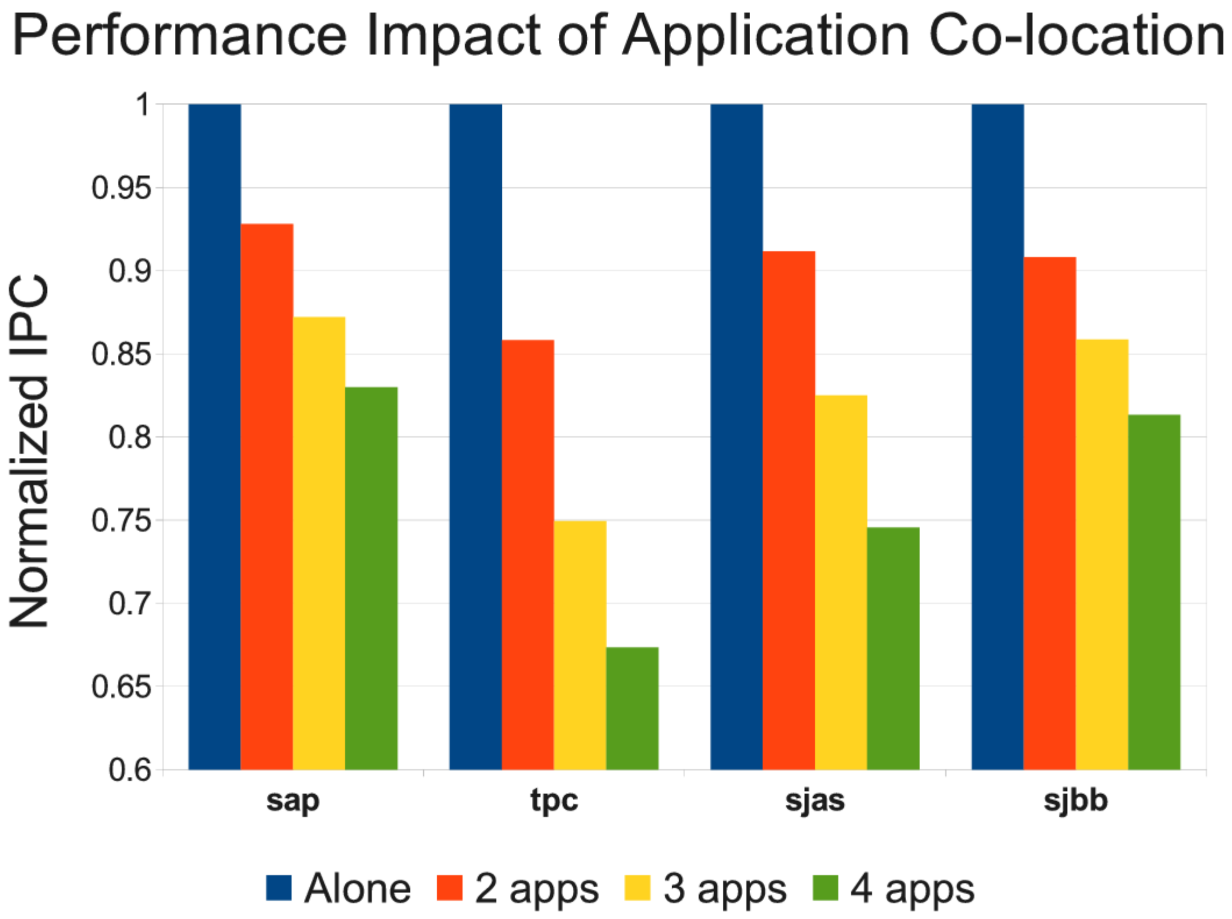
Questions?

Thank You

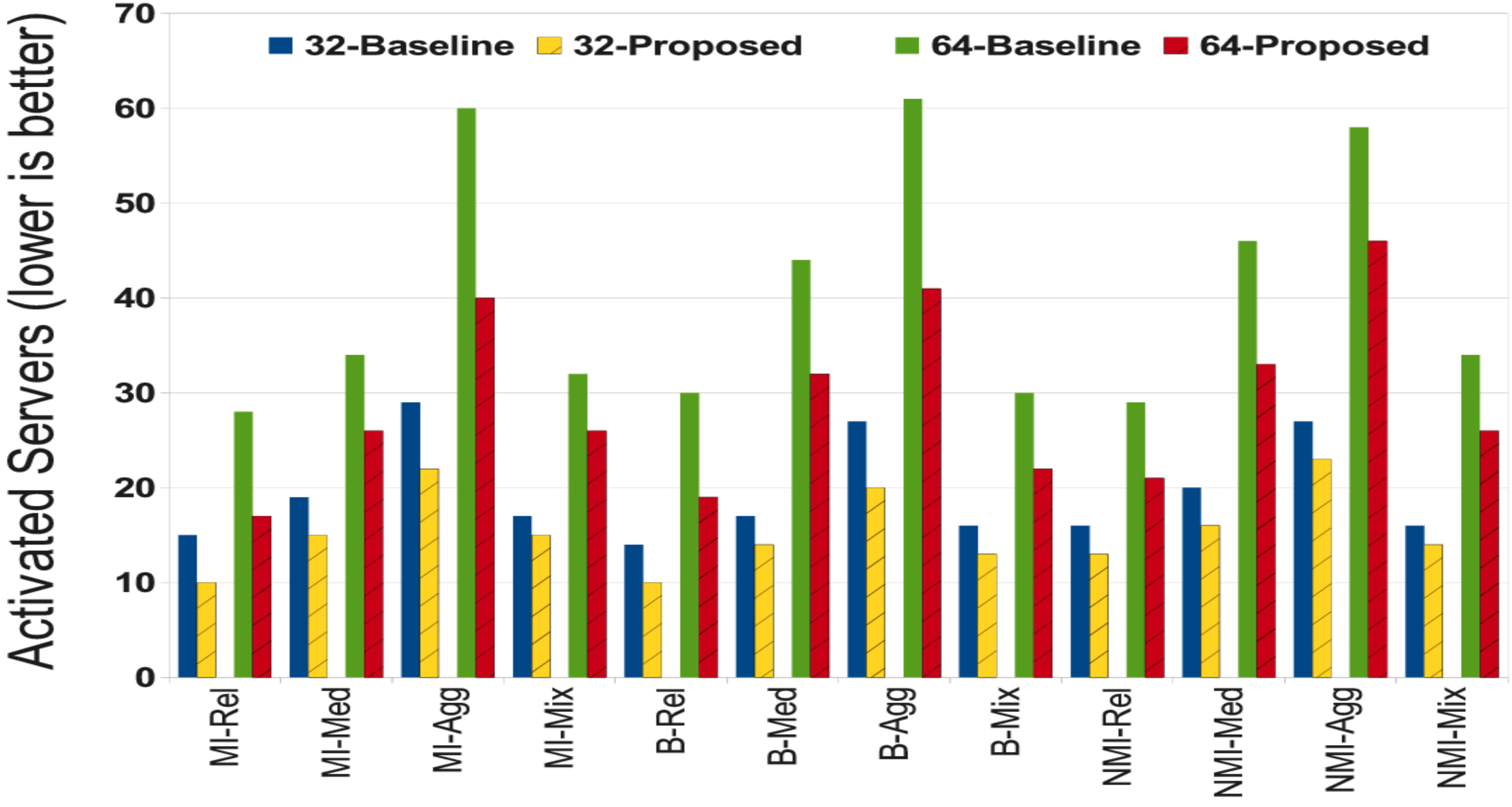
BACKUP SLIDES

Other Considerations

- QoS Aggressiveness
  - Indicates application resource requirements
  - Customers paying more demand more resources
  - If all apps have aggressive QoS guarantees, harder to find a dense co-location schedule
- Job list length
  - More applications implies more opportunity to find a co-location schedule which preserves QoS

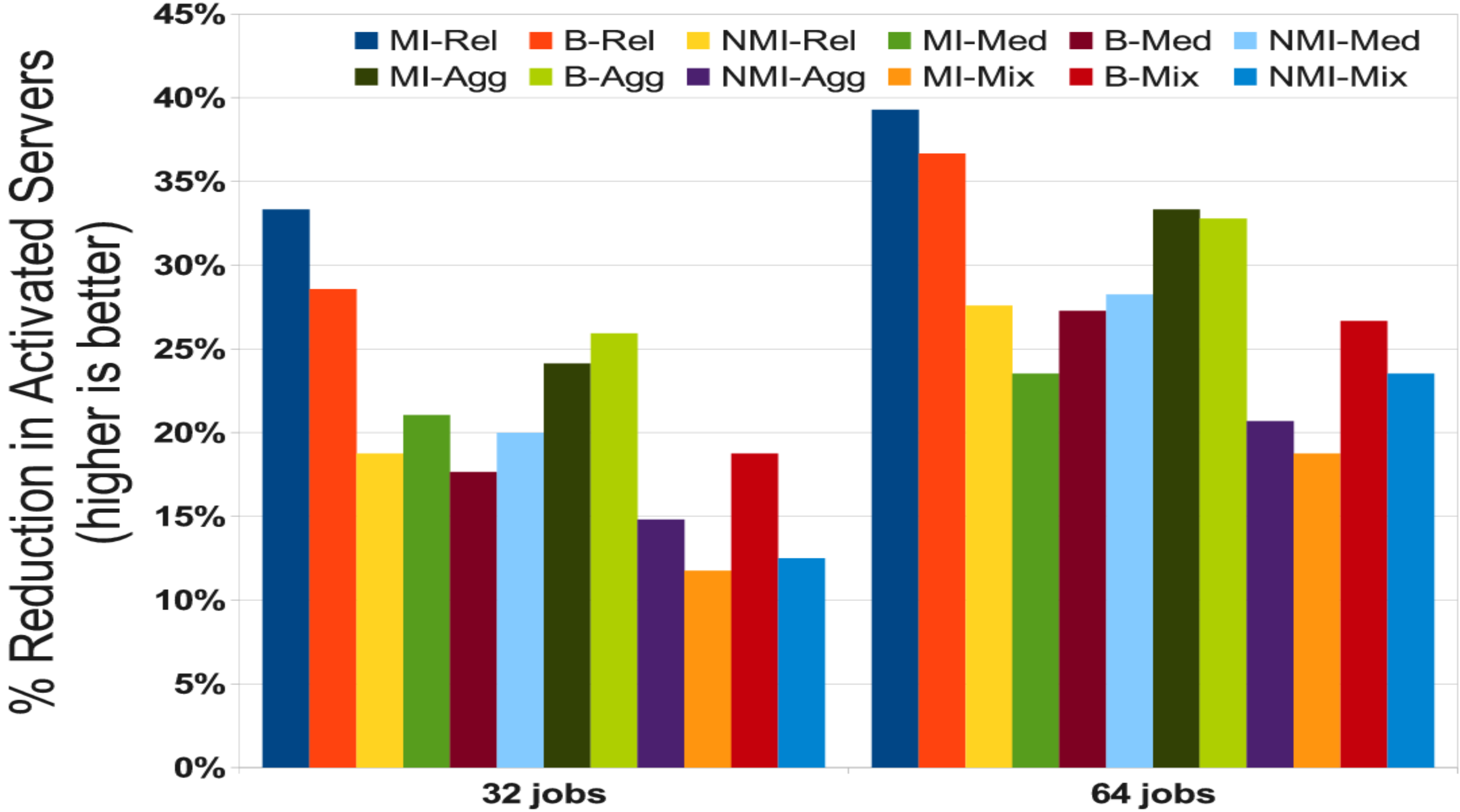

Performance Impact of Application Co-location
Normalized IPC
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
sap
tpc
sjas
sjbb
Alone
2 apps
3 apps
4 apps

Number of Activated Servers



UTAH ARCH

Impact of Job list Length



UTAH ARCH