Nexus: A New Approach to Replication in Distributed Shared Caches

Po-An Tsai, Nathan Beckmann, and Daniel Sanchez
Executive summary
Executive summary

- Data replication reduces the access latency of non-uniform caches (NUCA)
  - But replicating too aggressively leads to more cache misses
Executive summary

- Data replication reduces the access latency of non-uniform caches (NUCA)
  - But replicating too aggressively leads to more cache misses

- Prior adaptive techniques focus on **which data to replicate at each core**
  - Data that is not replicated locally still incurs high latency
Data replication reduces the access latency of non-uniform caches (NUCA)
- But replicating too aggressively leads to more cache misses

Prior adaptive techniques focus on which data to replicate at each core
- Data that is not replicated locally still incurs high latency

Nexus instead focuses on how much to replicate across the system
- Chooses the best number of replicas for the whole read-only working set
- Lets cores access replicas beyond their local bank
- Outperforms a state-of-the-art replication technique by up to 90%
The last-level cache (LLC) has become distributed and non-uniform (NUCA)
The last-level cache (LLC) has become distributed and non-uniform (NUCA)
The last-level cache (LLC) has become distributed and non-uniform (NUCA)
The last-level cache (LLC) has become distributed and non-uniform (NUCA)
The last-level cache (LLC) has become distributed and non-uniform (NUCA)

Key problem is what data to place and where to place it on chip
Static NUCA (S-NUCA) spreads data using a fixed line-to-bank mapping
Static NUCA (S-NUCA) spreads data using a fixed line-to-bank mapping
Static NUCA (S-NUCA) spreads data using a fixed line-to-bank mapping.
Static NUCA (S-NUCA) spreads data using a fixed line-to-bank mapping.

Threads  →  LLC data

Some near

Mostly far

Simple but large average distance
Replication reduces the distance to read-only data
Replication reduces the distance to read-only data
Replication reduces the distance to read-only data
Replication reduces the distance to read-only data
Replication reduces the distance to read-only data
Replication reduces the distance to read-only data
Replication reduces the distance to read-only data

Cache replicated read-only lines locally and check the local bank first. Upon a miss in the local bank, check the directory (at line’s original location).
Replication reduces the distance to read-only data

Cache replicated read-only lines locally and check the local bank first. Upon a miss in the local bank, check the directory (at line’s original location).
Replication reduces the distance to read-only data

Cache replicated read-only lines locally and check the local bank first. Upon a miss in the local bank, check the directory (at line’s original location).

Threads → LLC data

A, B, C, D are local, but replicated lines compete for cache capacity with other data.
Replication reduces the distance to read-only data

Cache replicated read-only lines locally and check the local bank first. Upon a miss in the local bank, check the directory (at line’s original location).

A, B, C, D are local, but replicated lines compete for cache capacity with other data.

Replicating too aggressively causes more cache misses than no replication.
Adaptive replication in directory-based dynamic NUCAs

ASR [Beckmann, MICRO 2006],
SP-NUCA [Dybdahl, HPCA 2007],
ECC [Herrero, ISCA 2010],
Locality-aware replication [Kurian, HPCA 2014]
Adaptive replication in directory-based dynamic NUCAs

Be **selective** about **which lines** to replicate, but **always** replicate them in the core’s local bank.

---

ASR [Beckmann, MICRO 2006],
SP-NUCA [Dybdahl, HPCA 2007],
ECC [Herrero, ISCA 2010],
Locality-aware replication [Kurian, HPCA 2014]
Adaptive replication in directory-based dynamic NUCAs

Be selective about which lines to replicate, but always replicate them in the core’s local bank.
Adaptive replication in directory-based dynamic NUCAs

Be **selective** about **which lines** to replicate, but **always** replicate them in the core’s local bank.

ASR [Beckmann, MICRO 2006],
SP-NUCA [Dybdahl, HPCA 2007],
ECC [Herrero, ISCA 2010],
Locality-aware replication [Kurian, HPCA 2014]
Adaptive replication in directory-based dynamic NUCAs

Be selective about which lines to replicate, but always replicate them in the core’s local bank.

A is nearby, but B, C, D are still far away

ASR [Beckmann, MICRO 2006], SP-NUCA [Dybdahl, HPCA 2007], ECC [Herrero, ISCA 2010], Locality-aware replication [Kurian, HPCA 2014]
Adaptive replication in directory-based dynamic NUCAs

Be selective about which lines to replicate, but always replicate them in the core’s local bank.

A is nearby, but B, C, D are still far away

Read-only data that is not replicated still causes high latency

ASR [Beckmann, MICRO 2006]
SP-NUCA [Dybdahl, HPCA 2007]
ECC [Herrero, ISCA 2010]
Locality-aware replication [Kurian, HPCA 2014]
Nexus spreads replicas across nearby banks to replicate more data.
Nexus spreads replicas across nearby banks to replicate more

Threads share a read-only data replica within a core group cluster
Nexus spreads replicas across nearby banks to replicate more

Threads **share** a read-only data **replica** within a core group cluster.
Nexus spreads replicas across nearby banks to replicate more.

Threads **share** a read-only data **replica** within a core group cluster.

A is local
B, C, D are just one hop away
Nexus spreads replicas across nearby banks to replicate more nearby. Threads **share** a read-only data **replica** within a core group cluster. A is local. B, C, D are just one hop away.
Nexus spreads replicas across nearby banks to replicate more

Threads **share** a read-only data **replica** within a core group cluster

All threads enjoy fast access to all read-only data by replicating beyond their local bank

A is local
B, C, D are just one hop away
An experiment to show why and when Nexus is better

A multithreaded workload that regularly scans over shared read-only data

![Graph showing access latency for different data footprints]

- Memory
- Remote LLC bank
- Local LLC bank

Access latency (lower is better)

Read-only data footprint

64KB (L1 cache)  512KB (Local LLC bank size)  6MB  72MB (Full LLC size)  256MB
An experiment to show why and when Nexus is better

A multithreaded workload that regularly scans over shared read-only data

![Diagram showing memory access latency comparison]

- **Access latency (lower is better)**
- **Remote LLC bank**
- **Local LLC bank**
- **64KB (L1 cache)**
- **512KB (Local LLC bank size)**
- **6MB**
- **72MB (Full LLC size)**
- **256MB**

**High average latency**
An experiment to show why and when Nexus is better

A multithreaded workload that regularly scans over shared read-only data

Data no longer fits in LLC
Previous replication techniques are ineffective when the read-only data does not fit in the local bank.
Previous replication techniques are ineffective when the read-only data does not fit in the local bank.

Access latency (lower is better)

Data fits in the local bank
Previous replication techniques are ineffective when the read-only data does not fit in the local bank.
Nexus allows replication even when read-only data cannot fit in the local bank.
Nexus allows replication even when read-only data cannot fit in the local bank.

Data fits in the local bank, each thread owns 1 replica.
Nexus allows replication even when read-only data cannot fit in the local bank.

Data fits in the local bank, each thread owns 1 replica

1 replica shared by every 4 neighbors

Access latency (lower is better)
Nexus allows replication even when read-only data cannot fit in the local bank.

Data fits in the local bank, each thread owns 1 replica.

1 replica shared by every 4 neighbors

1 replica shared by every 16 neighbors
Nexus allows replication even when read-only data cannot fit in the local bank.

Data fits in the local bank, each thread owns 1 replica

1 replica shared by every 4 neighbors

1 replica shared by every 16 neighbors

→ Same as S-NUCA

Access latency (lower is better)
Nexus allows replication even when read-only data cannot fit in the local bank.

Data fits in the local bank, each thread owns 1 replica

A significant latency reduction over prior work!

1 replica shared by all threads ➔ Same as S-NUCA
Recent directory-less dynamic NUCAs enable replication beyond the local bank
Recent directory-less dynamic NUCAs enable replication beyond the local bank.

Data placement is controlled using the virtual memory system and does not require a global directory.
Recent directory-less dynamic NUCAs enable replication beyond the local bank

Data placement is controlled using the virtual memory system and does not require a global directory.
Recent directory-less dynamic NUCAs enable replication beyond the local bank.

Data placement is controlled using the virtual memory system and does not require a global directory.

Data can be dynamically mapped to nearby banks and shared by arbitrary cores.
The number of replicas (*replication degree*) is important

Read-only Threads

- Threads: [Diagram showing multiple threads connected to read-only data]

16 MB LLC capacity

- Data: [Diagram showing data blocks connected to threads]
The number of replicas (replication degree) is important.

Read-only Threads

16 MB LLC capacity
The number of replicas (replication degree) is important.

Replicating 4 times works best
(4 x 4MB read-only = 16MB)

Read-only data (4MB)  Threads

16 MB LLC capacity
The number of replicas (replication degree) is important.

Replicating 4 times works best
(4 x 4MB read-only = 16MB)

Read-only data (4MB) → Threads

Choosing how much to replicate is more important than choosing which lines to replicate.
The number of replicas (*replication degree*) is important.
The number of replicas (*replication degree*) is important.
The number of replicas (*replication degree*) is important.

Replicating 8 times works best
(8 x 1MB read-only + 8MB other = 16MB)

Read-only data (1MB)

Other data (8MB)

16 MB LLC capacity

Threads

Diagram showing the replication process and data distribution.
The number of replicas (replication degree) is important

Replicating 8 times works best
(8 x 1MB read-only + 8MB other = 16MB)

Too few replicas cause extra network traversals,
while too many cause unnecessary cache misses
No adaptive replication in directory-less D-NUCAs
No adaptive replication in directory-less D-NUCAs

Reactive-NUCA (R-NUCA) [Hardavellas, ISCA 2009] always replicates instructions every 4 cores statically.
No adaptive replication in directory-less D-NUCAs

Reactive-NUCA (R-NUCA) [Hardavellas, ISCA 2009] always replicates instructions every 4 cores statically.

Other directory-less D-NUCAs do not replicate data.
Workloads have different preferences to *replication degrees*

- Study read-only data intensive workloads running on a 144-core system
- Apply different replication degrees for all read-only data
Workloads have different preferences to replication degrees

- Study read-only data intensive workloads running on a 144-core system
- Apply different replication degrees for all read-only data
Workloads have different preferences to replication degrees

- Study read-only data intensive workloads running on a 144-core system
  - Apply different replication degrees for all read-only data

**Observation 1:**
Applications prefer different degrees, requiring an adaptive approach.
Workloads have different preferences to replication degrees

- Study read-only data intensive workloads running on a 144-core system
  - Apply different replication degrees for all read-only data

Observation 1:
Applications prefer different degrees, requiring an adaptive approach.

Observation 2:
A few replication degrees suffice.
Nexus: enabling adaptive replication degrees in NUCA
Nexus: enabling adaptive replication degrees in NUCA

- Builds on top of **directory-less** D-NUCAs
  - Read-only data’s on-chip location and coherence are tracked via the virtual memory system
  - Cores access and **share closest replicas without directory** overheads
Nexus: enabling adaptive replication degrees in NUCA

- Builds on top of **directory-less** D-NUCA
  - Read-only data’s on-chip location and coherence are tracked via the virtual memory system
  - Cores access and **share closest replicas without directory** overheads

- **Nexus-R** builds on R-NUCA [Hardavellas, ISCA’09]
  - Supports flexible replication degrees for **all read-only data**
  - Leverages **set-sampling** to choose the best replication degree
Nexus: enabling adaptive replication degrees in NUCA

- Builds on top of **directory-less D-NUCAs**
  - Read-only data’s on-chip location and coherence are tracked via the virtual memory system
  - Cores access and **share closest replicas without directory** overheads

- **Nexus-R** builds on R-NUCA [Hardavellas, ISCA’09]
  - Supports flexible replication degrees for **all read-only data**
  - Leverages **set-sampling** to choose the best replication degree

- **Nexus-J** builds on Jigsaw
  - Extends Jigsaw’s configuration algorithm to select the best replication degree
  - Outperforms Nexus-R in multi-program workloads
Nexus: enabling adaptive replication degrees in NUCA

- Builds on top of directory-less D-NUCAs
  - Read-only data’s on-chip location and coherence are tracked via the virtual memory system
  - Cores access and share closest replicas without directory overheads

**Focus of this talk**

- **Nexus-R** builds on R-NUCA [Hardavellas, ISCA’09]
  - Supports flexible replication degrees for all read-only data
  - Leverages set-sampling to choose the best replication degree

- **Nexus-J** builds on Jigsaw
  - Extends Jigsaw’s configuration algorithm to select the best replication degree
  - Outperforms Nexus-R in multi-program workloads
Nexus-R: Applying Nexus to R-NUCA
Nexus-R: Applying Nexus to R-NUCA

- Nexus uses the virtual memory system to classify pages into three types.
  - Similar to R-NUCA, but differentiates all read-only data (not just instructions)
Nexus-R: Applying Nexus to R-NUCA

- Nexus uses the virtual memory system to classify pages into three types.
  - Similar to R-NUCA, but differentiates all read-only data (not just instructions)

![Diagram showing time and threads with labels X, Y, Z and an Unknown section]
Nexus-R: Applying Nexus to R-NUCA

- Nexus uses the virtual memory system to classify pages into three types.
  - Similar to R-NUCA, but differentiates all read-only data (not just instructions)
Nexus-R: Applying Nexus to R-NUCA

- Nexus uses the virtual memory system to classify pages into three types.
  - Similar to R-NUCA, but differentiates **all read-only data** (not just instructions)

```
Thread 0
Read X
Thread 1

X
Unknown
First TLB miss

Y
Z
Thread Private

Shared Read-only

Shared Read-write
```
Nexus-R: Applying Nexus to R-NUCA

- Nexus uses the virtual memory system to classify pages into three types.
  - Similar to R-NUCA, but differentiates all read-only data (not just instructions)

<table>
<thead>
<tr>
<th>Thread 0</th>
<th>Thread 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read X</td>
<td>Read Y</td>
</tr>
</tbody>
</table>

Time

First TLB miss

Unknown

Thread Private

X

Y

Shared Read-only

Shared Read-write
Nexus-R: Applying Nexus to R-NUCA

- Nexus uses the virtual memory system to classify pages into three types.
  - Similar to R-NUCA, but differentiates all read-only data (not just instructions)

```
<table>
<thead>
<tr>
<th>Thread 0</th>
<th>Thread 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read X</td>
<td>Read X</td>
</tr>
<tr>
<td>Read Y</td>
<td>Read Y</td>
</tr>
<tr>
<td>Read Y</td>
<td>Read Y</td>
</tr>
</tbody>
</table>
```

- First TLB miss
- Read TLB miss from other thread
- Unknown
- X (Thread Private)
- Y (Shared Read-only)
- Z (Shared Read-write)
Nexus-R: Applying Nexus to R-NUCA

- Nexus uses the virtual memory system to classify pages into three types.
  - Similar to R-NUCA, but differentiates all read-only data (not just instructions)

```
Thread 0  Thread 1
Read X    Read Y
Read Y
```

```
Thread Private
Unknown
First TLB miss

Read TLB miss from other thread

X

Y

Nexus-R replicates this

Shared Read-only

Shared Read-write
```

Time
Nexus-R: Applying Nexus to R-NUCA

- Nexus uses the virtual memory system to classify pages into three types.
  - Similar to R-NUCA, but differentiates all read-only data (not just instructions)

<table>
<thead>
<tr>
<th>Thread 0</th>
<th>Thread 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read X</td>
<td>Read Y</td>
</tr>
<tr>
<td>Read Y</td>
<td>Read Z</td>
</tr>
<tr>
<td>Read Y</td>
<td>Unknown</td>
</tr>
</tbody>
</table>

- First TLB miss

- Read TLB miss from other thread

- X: Private
- Z: Read-write
- Y: Shared Read-only

Nexus-R replicates this
Nexus-R: Applying Nexus to R-NUCA

- Nexus uses the virtual memory system to classify pages into three types.
  - Similar to R-NUCA, but differentiates all read-only data (not just instructions)

```
 Thread 0
  Read X
  Read Y
  Read Z

 Thread 1
  Read X
  Read Y
  Write Z
```

```
X
  Thread Private
  First TLB miss

Y
  Shared Read-only
  Write TLB miss from other thread

Z
  Shared Read-write
  Read TLB miss from other thread
```

Nexus-R replicates this
Nexus-R: Applying Nexus to R-NUCA
Nexus-R: Applying Nexus to R-NUCA

- Supports flexible replication degrees via flexible cluster sizes
  - R-NUCA always uses the cluster size of 4; Nexus-R supports reconfigurable sizes
Nexus-R: Applying Nexus to R-NUCA

- Supports flexible **replication degrees via flexible cluster sizes**
- R-NUCA always uses the cluster size of 4; Nexus-R supports reconfigurable sizes
Nexus-R: Applying Nexus to R-NUCA

- Supports flexible **replication degrees via flexible cluster sizes**
  - R-NUCA always uses the cluster size of 4; Nexus-R supports reconfigurable sizes
Nexus-R: Applying Nexus to R-NUCA

- Supports flexible **replication degrees via flexible cluster sizes**
  - R-NUCA always uses the cluster size of 4; Nexus-R supports reconfigurable sizes

Private data: Always local
Nexus-R: Applying Nexus to R-NUCA

- Supports flexible replication degrees via flexible cluster sizes
  - R-NUCA always uses the cluster size of 4; Nexus-R supports reconfigurable sizes
Nexus-R: Applying Nexus to R-NUCA

- Supports flexible **replication degrees via flexible cluster sizes**
  - R-NUCA always uses the cluster size of 4; Nexus-R supports reconfigurable sizes

Replication **degree of 9 on 36 cores** →
cluster with **size of 4 (36 divided by 9)**

- **Shared read-only data:** Replicated clusters
- **Private data:** Always local
- **Shared read-write data:** Always like S-NUCA
Nexus-R: Applying Nexus to R-NUCA

- Supports flexible **replication degrees via flexible cluster sizes**
  - R-NUCA always uses the cluster size of 4; Nexus-R supports reconfigurable sizes

Replication **degree of 9 on 36 cores** →
cluster with **size of 4** (36 divided by 9)

Shared read-only data: Replicated clusters
Nexus-R: Applying Nexus to R-NUCA

- Supports flexible **replication degrees via flexible cluster sizes**
  - R-NUCA always uses the cluster size of 4; Nexus-R supports reconfigurable sizes

Replication **degree of 9 on 36 cores** → cluster with **size of 4** (36 divided by 9)

Replication **degree of 4** → cluster with **size of 9**

**Shared read-only data:**
Replicated clusters
Nexus-R leverages set-sampling to select the best degree
Nexus-R leverages set-sampling to select the best degree

- Enhances set-sampling to monitor the performance of different degrees
Nexus-R leverages set-sampling to select the best degree

- Enhances set-sampling to monitor the performance of different degrees
- Compares the cumulative access latency of each degree from sampled sets
Nexus-R leverages set-sampling to select the best degree

- Enhances set-sampling to monitor the performance of different degrees
- Compares the cumulative access latency of each degree from sampled sets
Nexus-R leverages set-sampling to select the best degree

- Enhances set-sampling to monitor the performance of different degrees
- Compares the cumulative access latency of each degree from sampled sets

1. **L1 Miss**

Address to Bank/Set Lookup Logic

- L1s
- Core
Nexus-R leverages set-sampling to select the best degree

- Enhances set-sampling to monitor the performance of different degrees
- Compares the cumulative access latency of each degree from sampled sets
Nexus-R leverages set-sampling to select the best degree

- Enhances set-sampling to monitor the performance of different degrees
- Compares the cumulative access latency of each degree from sampled sets
Nexus-R leverages set-sampling to select the best degree

- Enhances set-sampling to monitor the performance of different degrees
- Compares the cumulative access latency of each degree from sampled sets

Counters record the latency difference between degrees

1. L1 Miss Address to Bank/Set Lookup Logic
2. Sampled access for degree of 4
3. Sampled access returns

1/4 1/9 1/36 4/9 4/36 9/36
Nexus-R leverages set-sampling to select the best degree

- Enhances set-sampling to monitor the performance of different degrees
- Compares the cumulative access latency of each degree from sampled sets

1. L1 Miss
2. Sampled access for degree of 4
3. Sampled access returns
4. Update counters

Counters record the latency difference between degrees
Nexus-R leverages set-sampling to select the best degree

- Enhances set-sampling to monitor the performance of different degrees
- Compares the cumulative access latency of each degree from sampled sets

Counters record the latency difference between degrees

1. L1 Miss
2. Sampled access for degree of 4
3. Sampled access returns
4. Update counters
5. Vote
Nexus-R leverages set-sampling to select the best degree

- Sampled sets spread across several banks
  - Threads share sampled sets if they share a read-only replica cluster
Nexus-R leverages set-sampling to select the best degree

- Sampled sets spread across several banks
- Threads share sampled sets if they share a read-only replica cluster
Nexus-R leverages set-sampling to select the best degree

- Sampled sets spread across several banks
  - Threads share sampled sets if they share a read-only replica cluster

![Diagram of LLC bank and threads with sampled sets for cluster size]
Nexus-R leverages set-sampling to select the best degree

- Sampled sets spread across several banks
- Threads share sampled sets if they share a read-only replica cluster
Nexus-R leverages set-sampling to select the best degree

- Sampled sets spread across several banks
- Threads share sampled sets if they share a read-only replica cluster

![Diagram showing LLC bank, sampling sets for cluster size, and threads sharing sampled sets.](image)
Nexus-R leverages set-sampling to select the best degree

- Sampled sets spread across several banks
- Threads share sampled sets if they share a read-only replica cluster

Sampling sets for cluster size:

<table>
<thead>
<tr>
<th>LLC bank:</th>
<th>Sampling sets</th>
<th>Threads:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>&gt;</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16</td>
<td></td>
</tr>
</tbody>
</table>
Nexus-R leverages set-sampling to select the best degree

- Sampled sets spread across several banks
- Threads share sampled sets if they share a read-only replica cluster

**LLC bank:**

- 1
- 4
- 9
- 16

**Sampling sets for cluster size:**

**Threads:**

For another thread in the system
Nexus-R takes coordinated decision across threads
Nexus-R takes coordinated decision across threads

- Uncoordinated decisions work poorly
  - “Tragedy of the commons”: Each thread wants itself to replicate more, but others to replicate less
Nexus-R takes coordinated decision across threads

- Uncoordinated decisions work poorly
  - “Tragedy of the commons”: Each thread wants itself to replicate more, but others to replicate less

- Nexus-R makes the whole process agree on the best replication degree by using per-process total latency for each degree
  - The OS reads latency counters periodically and sets the best degree for a process
Nexus-R adds small overheads over R-NUCA
Nexus-R adds small overheads over R-NUCA

- Overheads of applying Nexus to R-NUCA:
  - 1.5% of the LLC used for set-sampling
  - ~100 bits per core for hardware counters
  - 10s of instructions per context switch for the OS support
Jigsaw groups partitions from different banks to create virtual caches (VCs).
Nexus-J: Applying Nexus to Jigsaw

- Jigsaw groups partitions from different banks to create *virtual caches (VCs)*

![Diagram of 4x4 mesh NUCA LLC]

<table>
<thead>
<tr>
<th>LLC Bank</th>
<th>L1I</th>
<th>L1D</th>
<th>Core</th>
</tr>
</thead>
</table>

4x4 mesh NUCA LLC
Nexus-J: Applying Nexus to Jigsaw

- Jigsaw groups partitions from different banks to create **virtual caches (VCs)**

![Diagram showing LLC Bank, L1I, L1D, Core, VC1, VC2, VC3, 4x4 mesh NUCA LLC](image)

23
Nexus-J: Applying Nexus to Jigsaw

- Jigsaw groups partitions from different banks to create *virtual caches (VCs)*
Jigsaw groups partitions from different banks to create virtual caches (VCs).

Jigsaw manages capacity among applications and data types, outperforming many D-NUCA techniques.
Nexus-J: Applying Nexus to Jigsaw
Nexus-J: Applying Nexus to Jigsaw

- Jigsaw outperforms R-NUCA’s simple heuristics with better data placement
  - Especially in **multi-programmed workloads**
  - But Jigsaw **never replicates** data!!
Nexus-J: Applying Nexus to Jigsaw

- Jigsaw outperforms R-NUCA’s simple heuristics with better data placement
  - Especially in **multi-programmed workloads**
  - But Jigsaw **never replicates** data!!

- Nexus-J implements adaptive replication degree on Jigsaw
  - Combines the ability to allocate capacity between apps with adaptive replication
  - Enhance Jigsaw’s software runtime to select the best replication degree
Nexus-J: Applying Nexus to Jigsaw

- Jigsaw outperforms R-NUCA’s simple heuristics with better data placement
  - Especially in multi-programmed workloads
  - But Jigsaw never replicates data!!

- Nexus-J implements adaptive replication degree on Jigsaw
  - Combines the ability to allocate capacity between apps with adaptive replication
  - Enhance Jigsaw’s software runtime to select the best replication degree

- See the paper for implementation details
Evaluation

- **Modeled system**
  - 144 Silvermont-like OOO cores
  - 12x12 mesh
  - 32KB L1I/D caches
  - 72MB LLC (0.5MB per core)

- **Multithreaded workloads**
  - Scientific workloads: SPECOMP2012, PARSEC, SPLASH2, BioParallel
  - Server workloads: TailBench [Kasture, IISWC’16]
  - With various input sizes
## Evaluation

- Compared 6 schemes

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-NUCA</td>
<td>No replication (baseline).</td>
</tr>
<tr>
<td>R-NUCA</td>
<td>Replicate instructions at a fixed degree.</td>
</tr>
<tr>
<td>Jigsaw</td>
<td>Allocate capacity across processes. No replication.</td>
</tr>
<tr>
<td>Locality-aware replication</td>
<td>State-of-the-art directory-based D-NUCA. Selectively replicate cache lines in local bank.</td>
</tr>
<tr>
<td>[Kurian, HPCA’14]</td>
<td></td>
</tr>
<tr>
<td>Nexus-R</td>
<td>Nexus on R-NUCA.</td>
</tr>
<tr>
<td>Nexus-J</td>
<td>Nexus on Jigsaw.</td>
</tr>
</tbody>
</table>
Nexus outperforms prior selective replication techniques

- Single-program workloads running with 144 threads

<table>
<thead>
<tr>
<th>R-NUCA</th>
<th>Locality-aware</th>
<th>Nexus-J</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jigsaw</td>
<td>Nexus-R</td>
<td></td>
</tr>
</tbody>
</table>
Nexus outperforms prior selective replication techniques

- Single-program workloads running with 144 threads

Workloads with small read-only footprint → Nexus matches prior work
Nexus outperforms prior selective replication techniques

- Single-program workloads running with 144 threads

- Workloads with small read-only footprint → Nexus matches prior work

- Workloads with medium read-only footprint → Nexus outperforms prior work
Nexus outperforms prior selective replication techniques

- Single-program workloads running with 144 threads

- Workloads with small read-only footprint → Nexus **matches** prior work
- Workloads with medium read-only footprint → Nexus **outperforms** prior work
- Workloads with large read-only footprint → Nexus **does not hurt** performance
Nexus outperforms prior selective replication techniques

- Workloads with small read-only footprint → Nexus matches prior work
- Workloads with medium read-only footprint → Nexus outperforms prior work
- Workloads with large read-only footprint → Nexus does not hurt performance

Single-program workloads running with 144 threads

- R-NUCA
- Locality-aware
- Jigsaw
- Nexus-J
- Nexus-R
Nexus-J performs best with multi-programmed workloads

- Workload mixes with 4 different apps running with 36 threads each
Nexus-J performs best with multi-programmed workloads

- Workload mixes with 4 different apps running with 36 threads each
Nexus-J performs best with multi-programmed workloads

- Workload mixes with 4 different apps running with 36 threads each

Replication-sensitive $\rightarrow$

**Nexus-R and Nexus-J**
are better
Nexus-J performs best with multi-programmed workloads

- Workload mixes with 4 different apps running with 36 threads each

Replication-sensitive $\rightarrow$ **Nexus-R and Nexus-J are better**

Capacity-sensitive $\rightarrow$ **Jigsaw and Nexus-J are better**
Nexus-J performs best with multi-programmed workloads

- Workload mixes with 4 different apps running with 36 threads each

<table>
<thead>
<tr>
<th>Replication-sensitive</th>
<th>Capacity-sensitive</th>
<th>Sensitive to both</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nexus-R and Nexus-J</td>
<td>Jigsaw and Nexus-J</td>
<td>Nexus-J performs the best</td>
</tr>
</tbody>
</table>

Graphs showing speedup vs. S-NUCA for different workloads.

- Red: R-NUCA
- Green: Jigsaw
- Brown: Locality-aware
- Orange: Nexus-R
- Blue: Nexus-J
See paper for more results

- Performance of 60 apps between Nexus-R and Locality-aware replication
- Dynamic replication degree vs. static degrees
- Result of 20 Multi-program workloads
- Sensitivity study to
  - System sizes
  - Different cache hierarchies
- Dynamic data reclassification
Data replication can improve the performance of NUCA systems

Replication requires balancing the latency and capacity tradeoff in NUCA
Data replication can improve the performance of NUCA systems
- Replication requires balancing the latency and capacity tradeoff in NUCA

We propose Nexus, a new approach to adaptive replication
- Unlike prior work, Nexus focuses on **how much to replicate** the read-only data
- We present two implementation of Nexus: **Nexus-R** and **Nexus-J**
Conclusion

- Data replication can improve the performance of NUCA systems
  - Replication requires balancing the latency and capacity tradeoff in NUCA

- We propose Nexus, a new approach to adaptive replication
  - Unlike prior work, Nexus focuses on how much to replicate the read-only data
  - We present two implementation of Nexus: Nexus-R and Nexus-J

- Nexus outperforms the state-of-the-art adaptive scheme
  - By up to 90% and 23% on average for replication sensitive workloads
Data replication can improve the performance of NUCA systems
- Replication requires balancing the latency and capacity tradeoff in NUCA

We propose Nexus, a new approach to adaptive replication
- Unlike prior work, Nexus focuses on how much to replicate the read-only data
- We present two implementation of Nexus: Nexus-R and Nexus-J

Nexus outperforms the state-of-the-art adaptive scheme
- By up to 90% and 23% on average for replication sensitive workloads