Brandenburg
University of Technology
Cottbus - Senftenberg

28th August 2017

HELP YOUR BUSY NEIGHBOURS

DYNAMIC MULTICASTS OVER STATIC TOPOLOGIES

Robert Kuban, Randolf Rotta, Jörg Nolte

Distributed Systems / Operationg Systems

OUR TARGET SCENARIO

objective: scalable multicasts

+ acknowledgement of completion
+ dynamic group membership (join/leave)

applications: cache invalidation, esp. TLB shootdown

hardware: many-cores like Intel XeonPhi, Tilera TilePro. . .

+ cache-coherent shared memory
+ point-to-point message passing

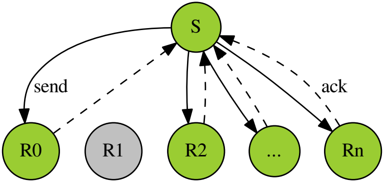
EXAMPLE: LINUX TLB SHOOTDOWN

Linux 4.11 x86 `smp_call_function_many()`

Brandenburg
University of Technology
Cottbus - Senftenberg

Initiator (Sender)

1. update page tables
2. enqueue invalidation tasklet at each thread
3. send IPI to each thread
4. wait on flag in each tasklet



Other CPU Threads

IPI handler processes tasklet:

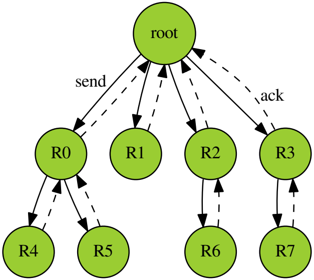
1. invalidate page(s) in TLB
2. set ACK flag in tasklet

⇒ flat topology

👍 fast join/leave via bit-mask

👎 $\mathcal{O}(n)$ latency

EXAMPLE: MULTICASTS IN BARRELFISH



- propagate along a tree topology
- use constraint solver
  for optimized topology
- proposed for TLB shootdowns[1]

👎 expensive join/leave
or interrupt ex-members

👍 $\mathcal{O}(\log n)$ latency

[1] Baumann et al., *The multikernel: A new OS architecture for scalable multicore systems*, 2009

DESIGN SPACE

	Multicasts (just members)	Broadcasts (over all threads)
Flat	👍 low latency for small groups 👎 high latency for large groups 👍 fast join/leave	👎 always high latency 👎 interrupts non-members
Tree	👍 always low latency 👎 costly join/leave	👍 good latency for large groups 👎 bad latency for small groups 👎 interrupts non-members

MULTICASTS ON A STATIC TOPOLOGY

Problem Statement: Combine…

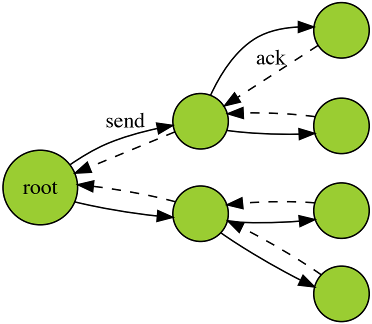
- fast join/leave like with flat topology
- low latency like in tree topologies (parallel propagation)

Solution Idea

- use static tree topology like in broadcasts
  (can be hand-crafted for the processor)
- membership as bit-mask for fast join/leave
- exploit shared memory to skip non-members,
  just message passing to actual members

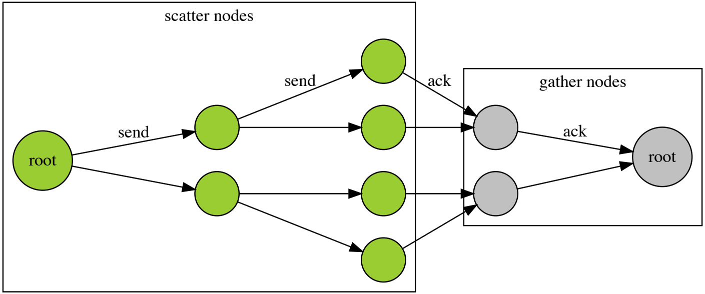
TREES WITH ACKNOWLEDGEMENT

Nodes = Cores; Two roles at each node

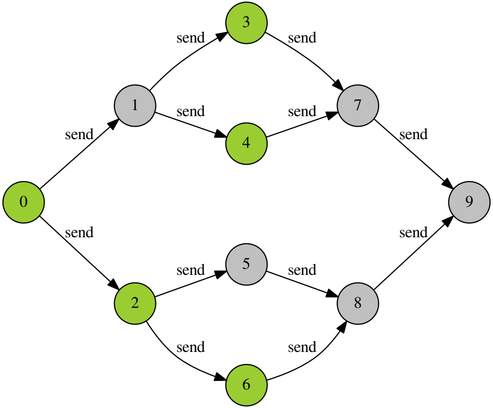


TREES WITH ACKNOWLEDGEMENT

Logical nodes for larger design space & simpler code

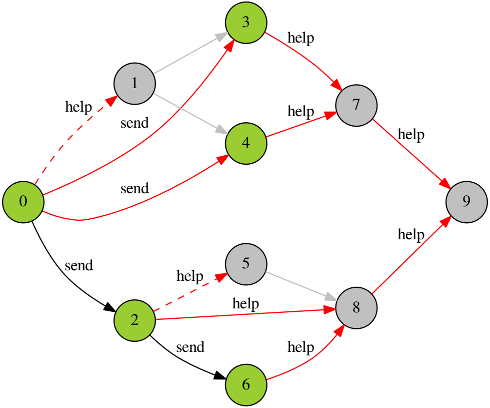


NON-MEMBER NODES IN BROADCASTS

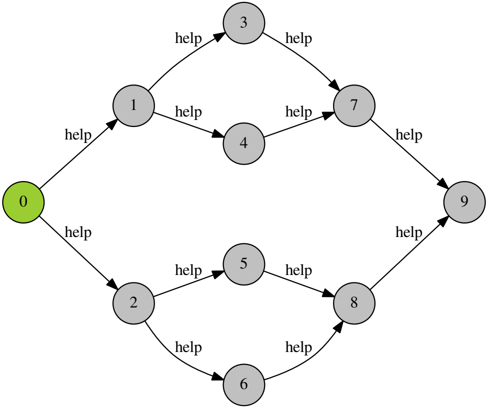


SOLUTION: HELPING

Skip non-member scatter nodes

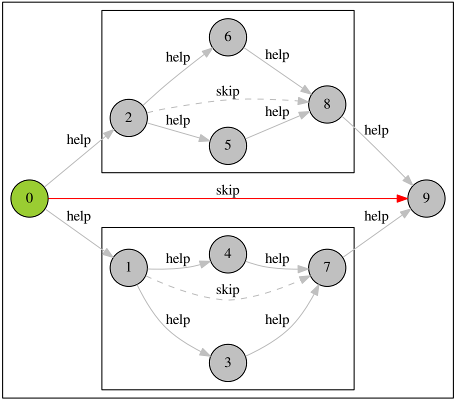


HUGE OVERHEAD FOR SMALL GROUPS :(



SOLUTION: SKIPPING

Jump over whole subtrees

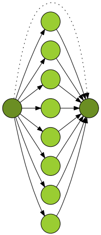


EVALUATION SETUP

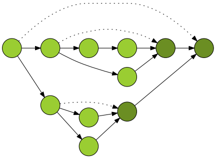
Setup

- Intel XeonPhi Knights Corner (1.053 GHz)
- 60 cores
- message passing via shared memory
- polling

Flat Topology

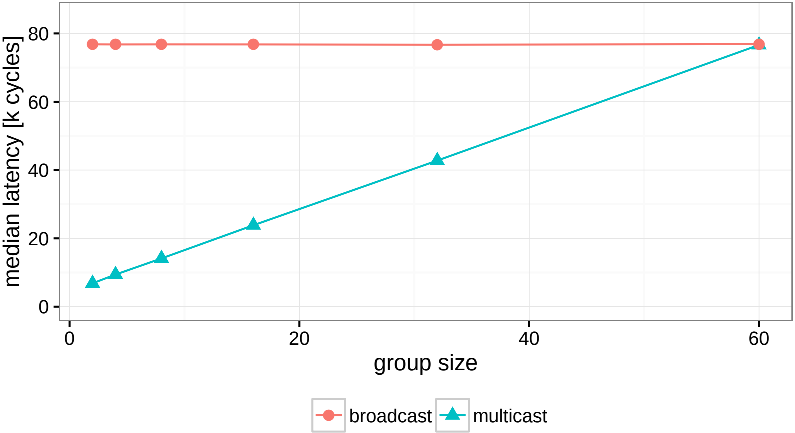


Binary Tree



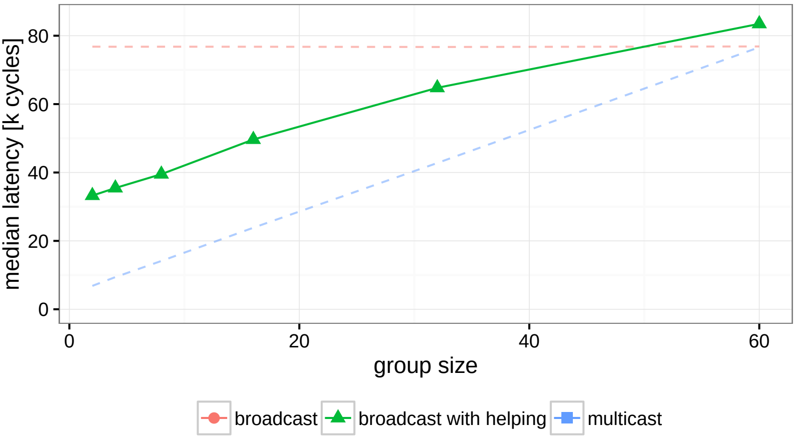
FLAT TOPOLOGY

multicast similar to Linux TLB shootdown



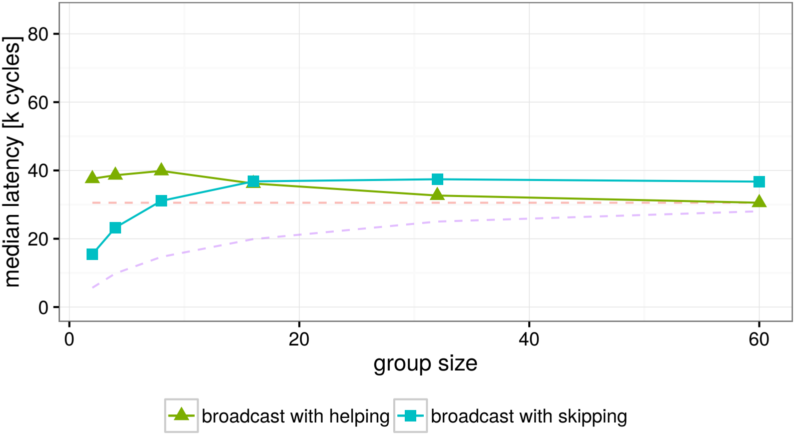
FLAT TOPOLOGY WITH HELPING

Overhead from membership tests and graph traversal



BINARY TREE WITH HELPING, SKIPPING

Brandenburg
University of Technology
Cottbus - Senftenberg



CONCLUSION

Scalable, acknowledged, dynamic multicasts for manycores:

Challenges: generating good topologies is costly,
flat topology not scalable,
non-members should not be interrupted

Solution: static optimized broadcast topology,
help and skip non-member cores

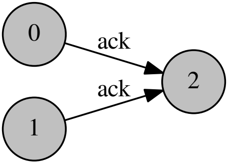
Result: success for large groups, alright for small

Implications: improve Linux TLB shootdown for Many-Core HPC apps

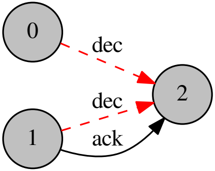
ACKNOWLEDGE VIA SHARED MEMORY

Decrement shared variable instead of message passing

Only message passing:

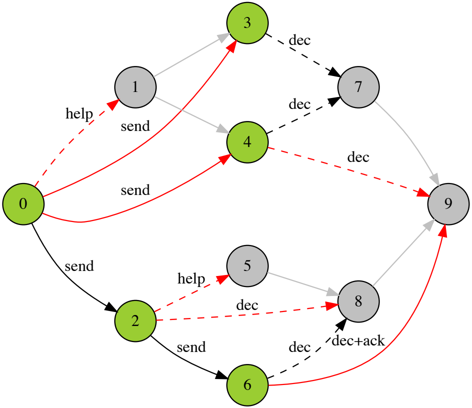


Using shared memory:



HELPING WITH SHARED MEM ACK

aka tree combining



INTERRUPT-FREE INVALIDATION

Many-core systems

- (incoherent) shared caches in cores/tiles
- abundance of cores
- symmetric multithreading

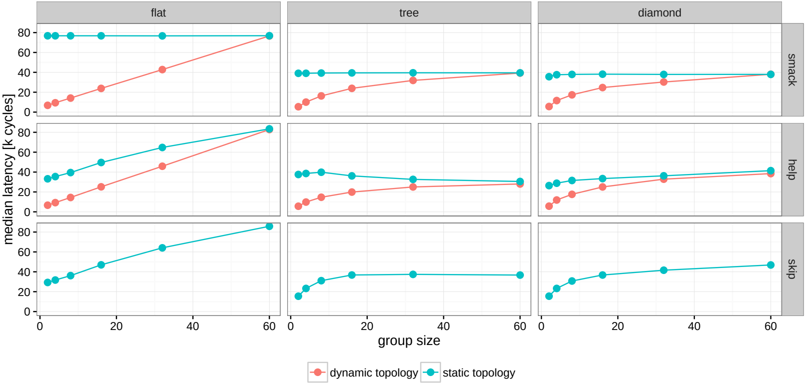
Assumption

- shared TLB can be invalidated by any sharer
- superfluous hardware threads per shared cache

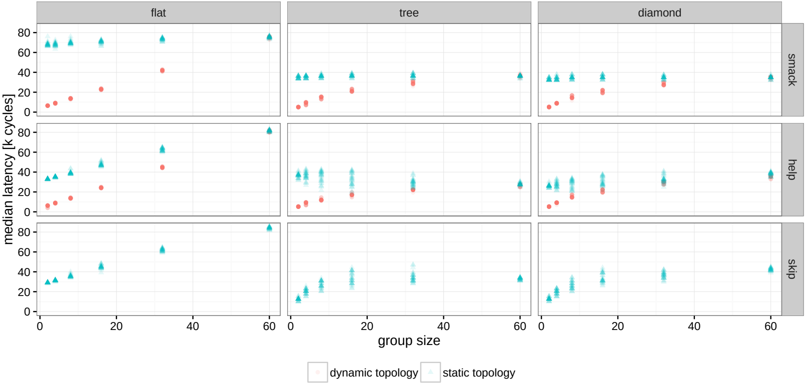
Algorithm

- one dedicated hardware thread per core/tile → still many cores
- polling/waiting for notifications → core can not sleep
- TLB invalidation for all threads in core/tile

EVALUATION


flat
tree
diamond
smack
help
skip
median latency [k cycles]
80
60
40
20
0
0
20
40
60
group size
dynamic topology
static topology

EVALUATION


flat
tree
diamond
smack
help
skip
median latency [k cycles]
0
20
40
60
80
group size
0
20
40
60
dynamic topology
static topology