A Study of Network Quality of Service in Many-Core MPI Applications

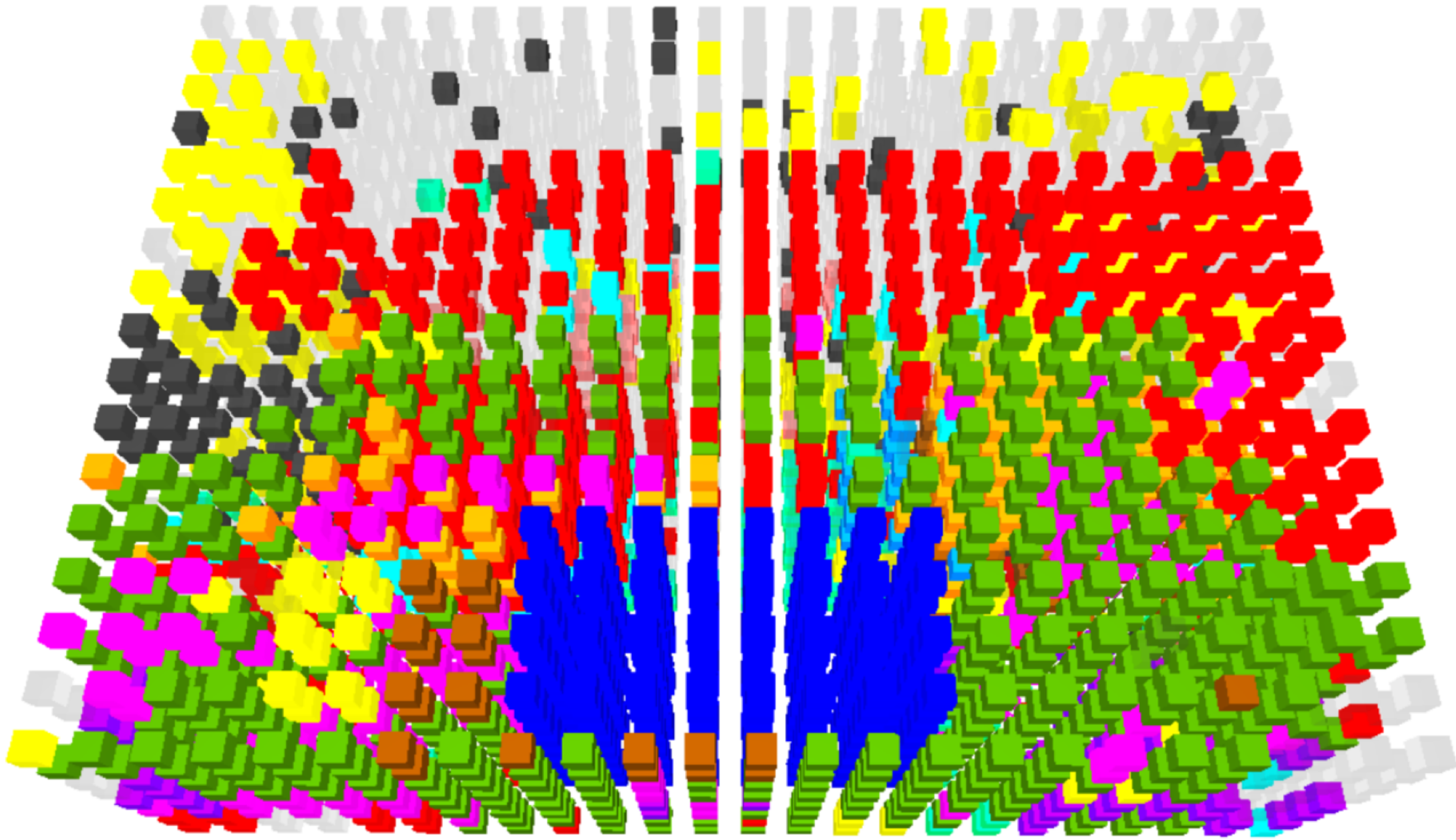
Lee Savoie[1], David Lowenthal[1], Bronis de Supinski[2], Kathryn Mohror[2]

[1]The University of Arizona, [2]Lawrence Livermore National Laboratory

Introduction

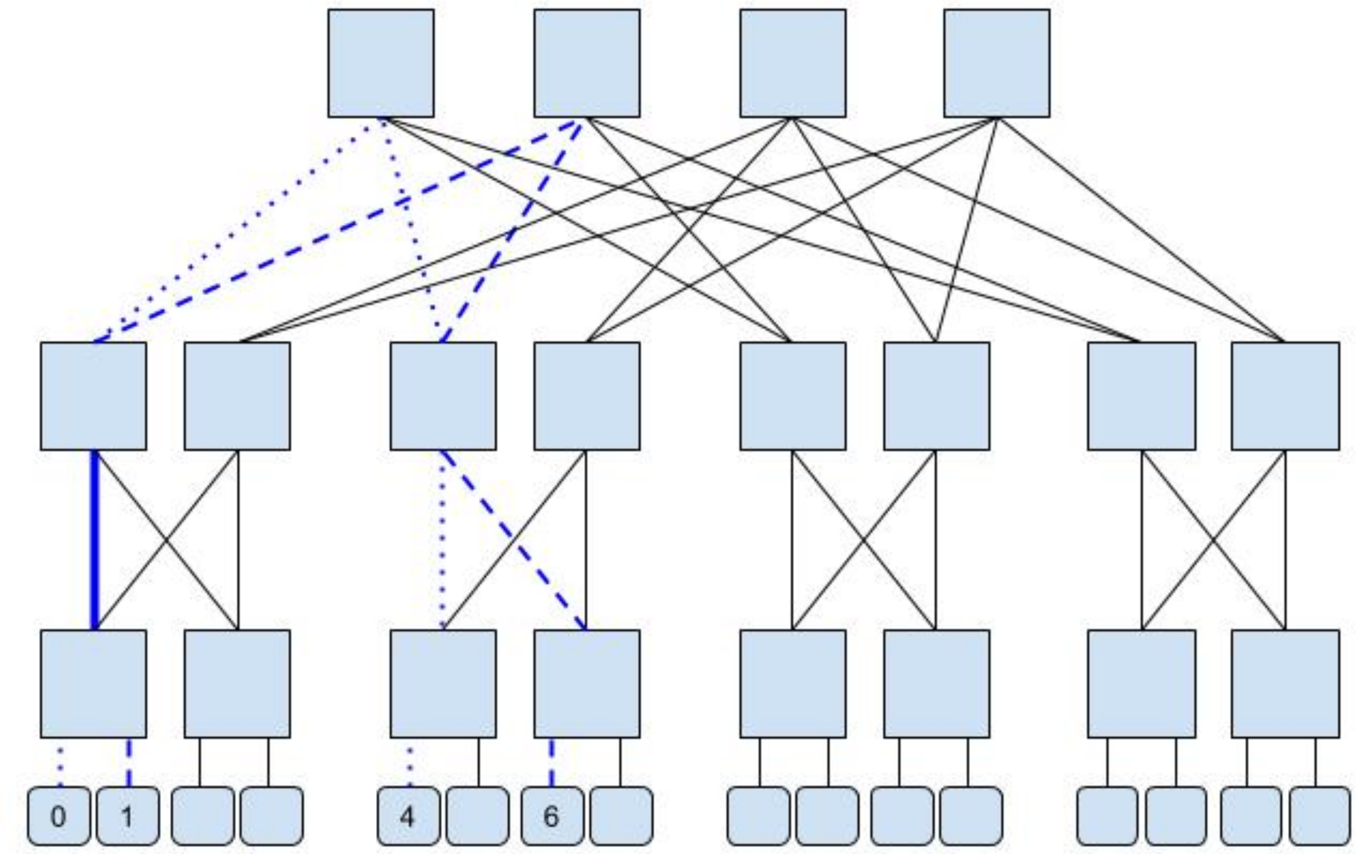
- Core counts increasing in high performance computing (HPC)
- Many machines already include many-core accelerators
- Many-core nodes process more data
- The network must work harder to transfer data between nodes

Network Contention



"There goes the neighborhood: performance degradation due to nearby jobs" (Bhatele et al., SC 13)

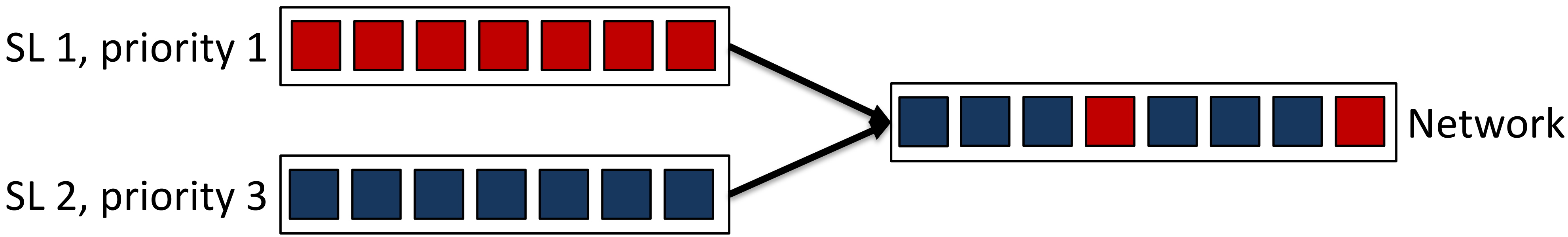
Fat-tree Contention



- HPC systems with many-core nodes need better network management

Quality of Service (QoS)

- Most networks provide QoS mechanisms for network management
- In Infiniband:
  - Packets are marked with a service level (SL)
  - Each SL has a priority



Research Question

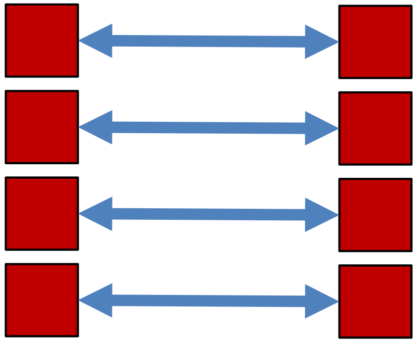
- Can we improve the performance of contending jobs on HPC systems using QoS?
  - This will enable HPC systems to handle the increased data demands of many-core nodes.
- This work focuses on per-job QoS
  - Each job runs in a separate service level
  - Each job is guaranteed a minimum amount of bandwidth

Experimental Set Up

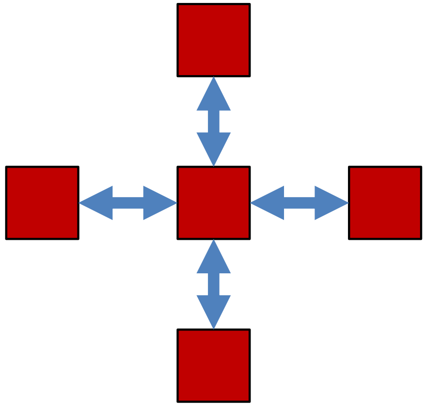
- 300 node machine
  - Left 20 nodes free in case of failures
  - No other jobs running
- Service levels with priorities 2286:254:9:1
- Applications
  - QBox
  - Crystal Router
  - MILC
  - pF3D
- Micro-benchmarks

Micro-Benchmarks

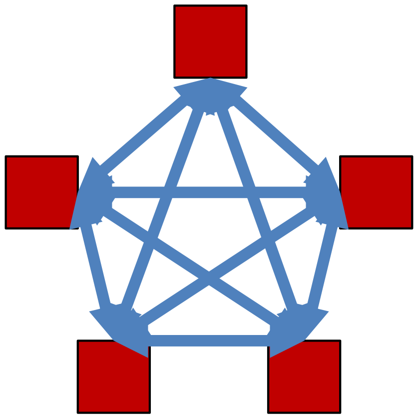
Flood-Pairs



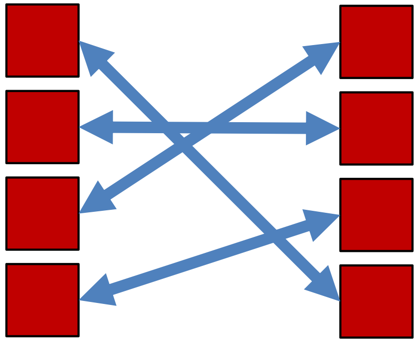
Nearest-Neighbor



All-to-all



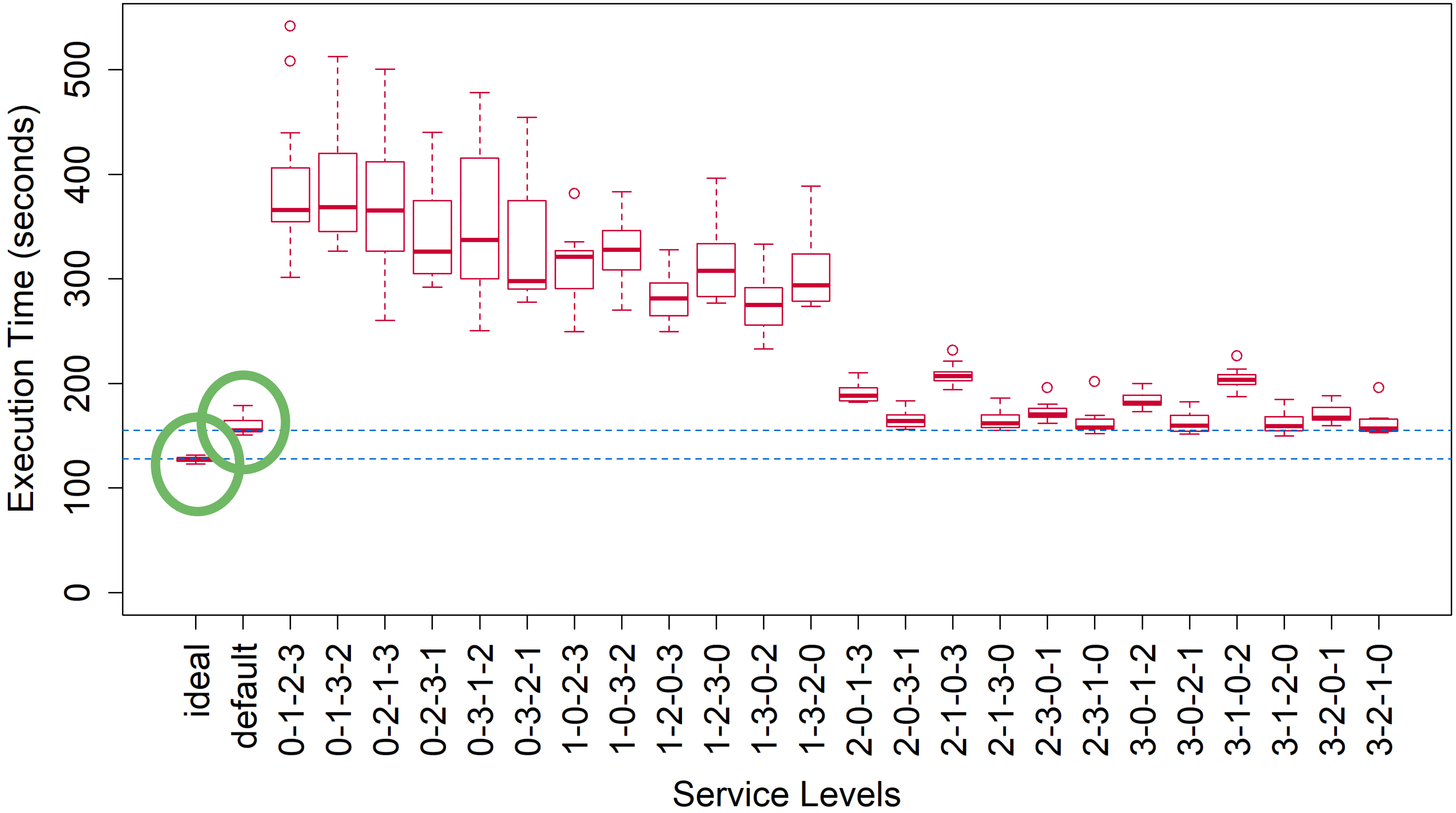
Random-Pairs



Methodology

- Ran 4 jobs at a time
  - 70 nodes each
  - 22 ranks per node
- Assigned nodes to jobs randomly
  - Repeated tests several times with different node assignments
- Restarted each job when it completed to maintain contention profile until all jobs completed at least once
- Ran the following tests
  - Ideal – each job running in isolation
  - Default – all jobs in the same service level
  - All assignments of jobs to 4 service levels

Results: Micro-Benchmarks



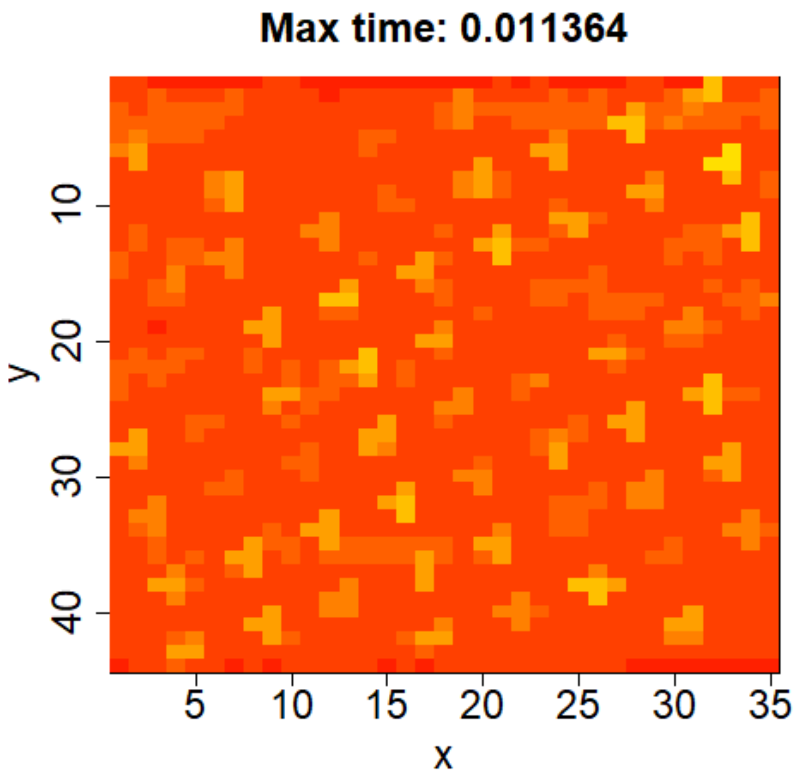
- Per-job QoS is insufficient to improve performance.

Flood-pairs Rank Timing

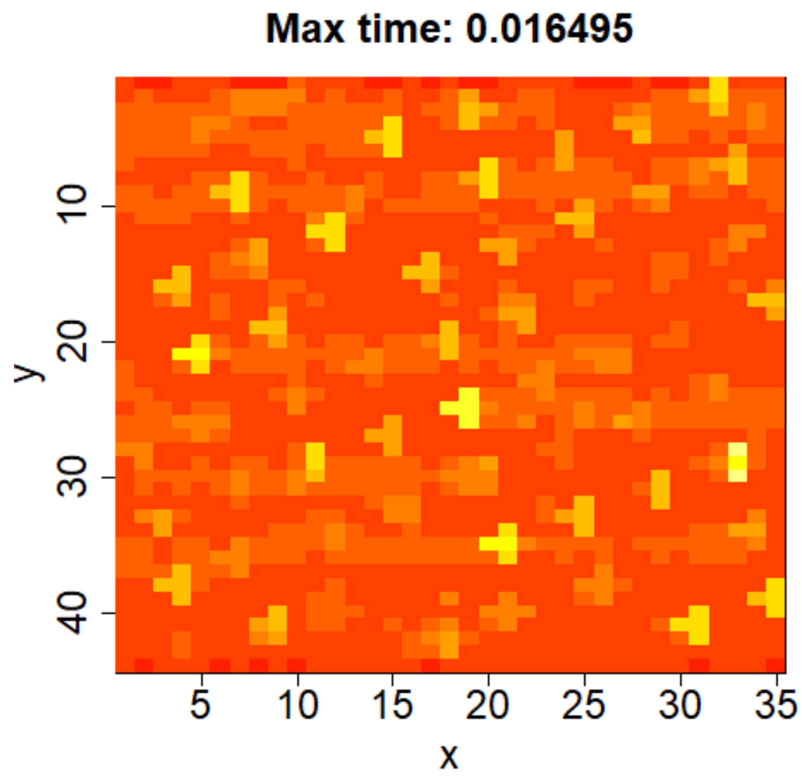
- Only a few ranks need to be prioritized.

Nearest-neighbor Rank Timing

High Priority

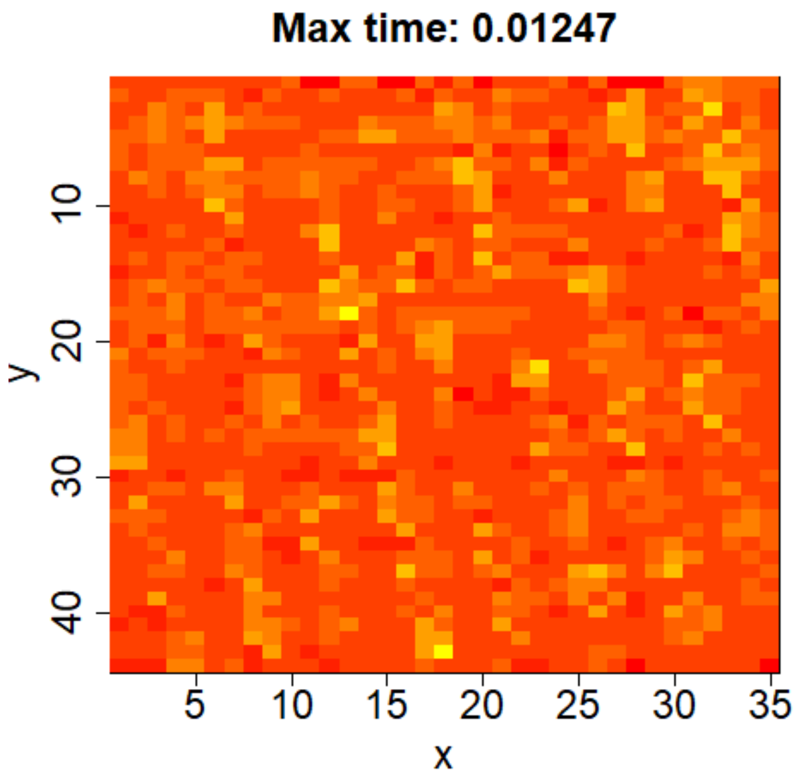


Contended

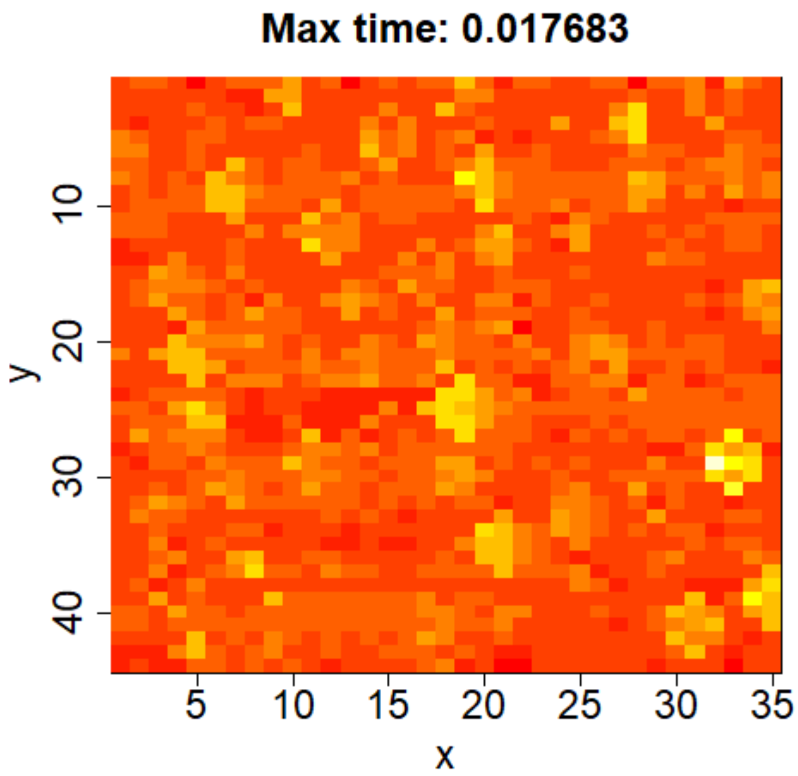


Nearest-neighbor Rank Timing

High Priority

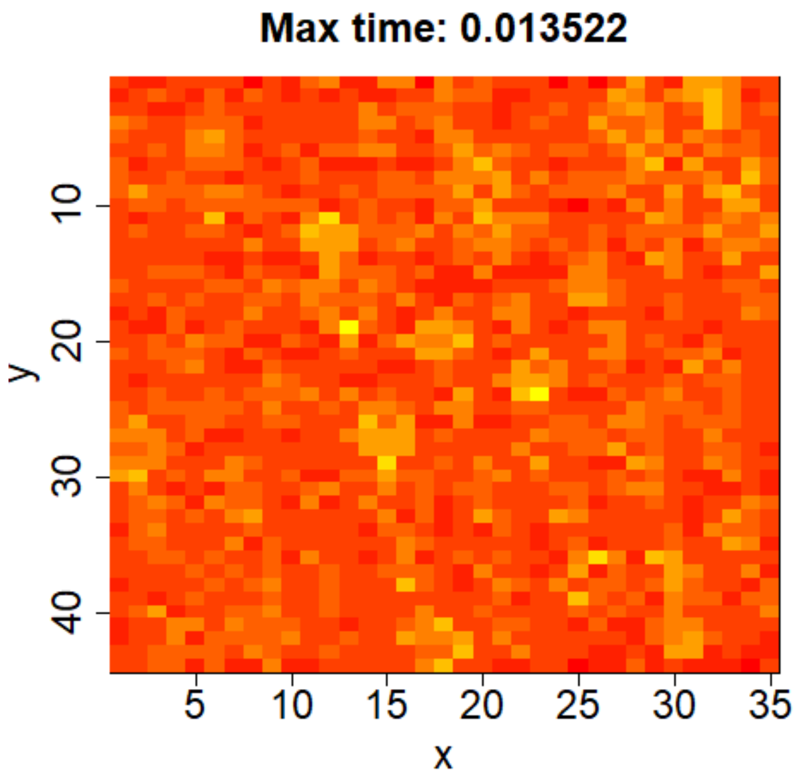


Contended

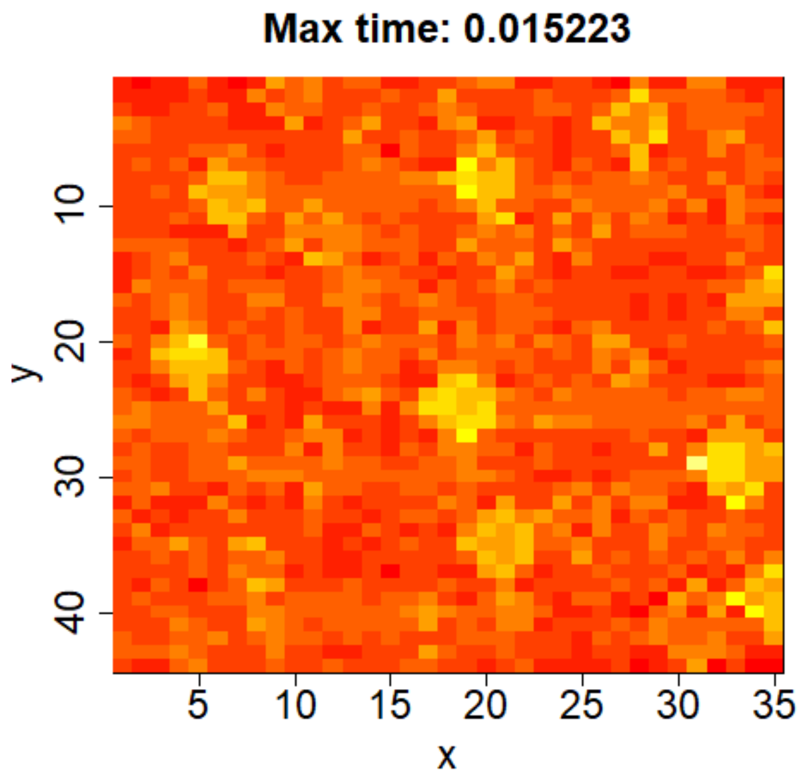


Nearest-neighbor Rank Timing

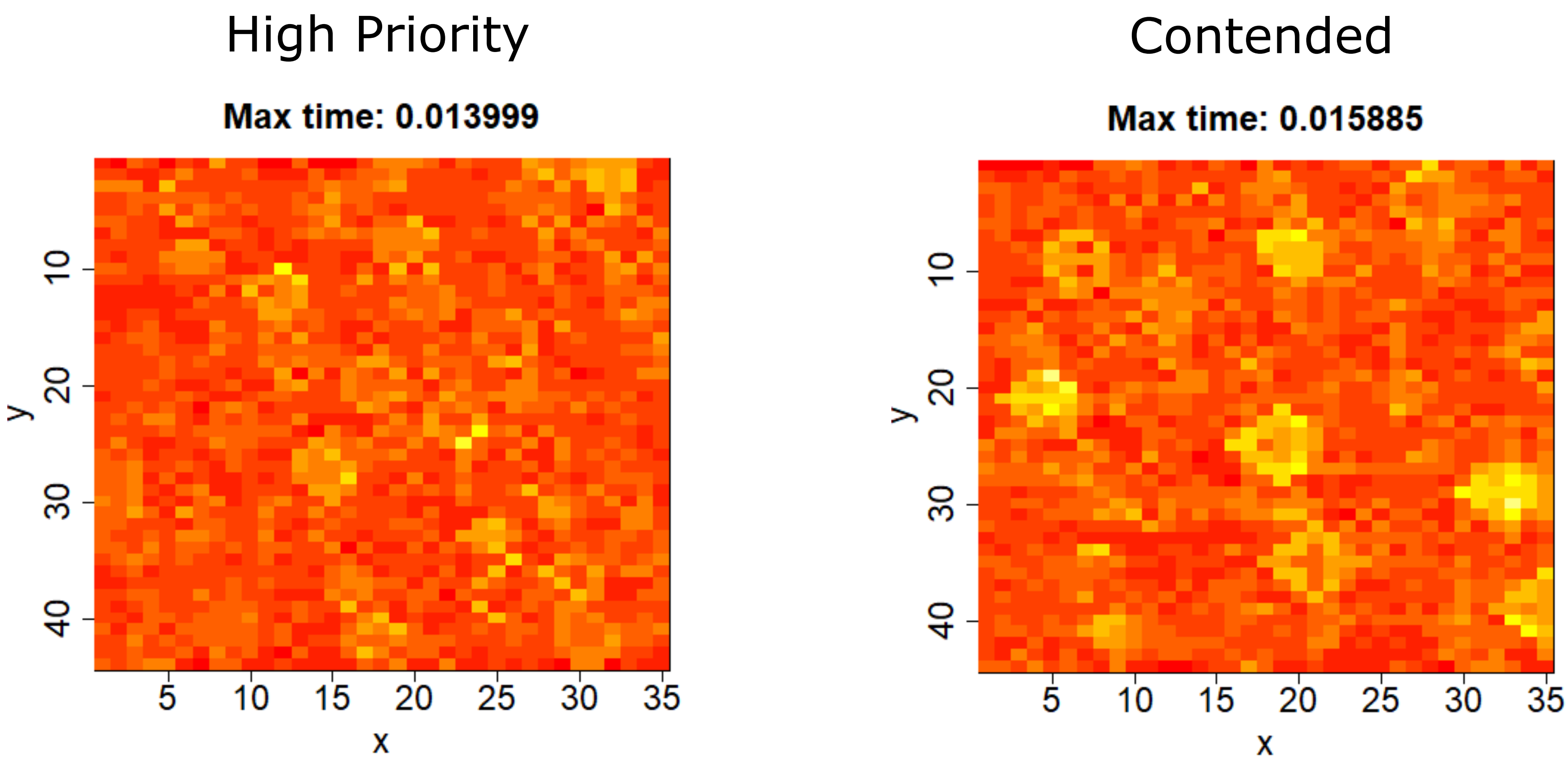
High Priority



Contended

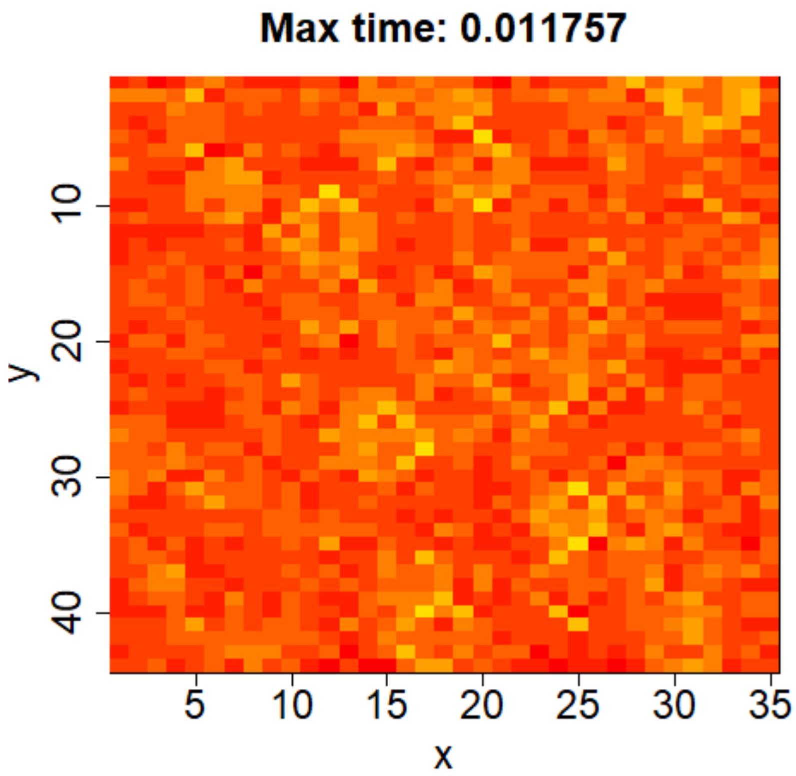


Nearest-neighbor Rank Timing

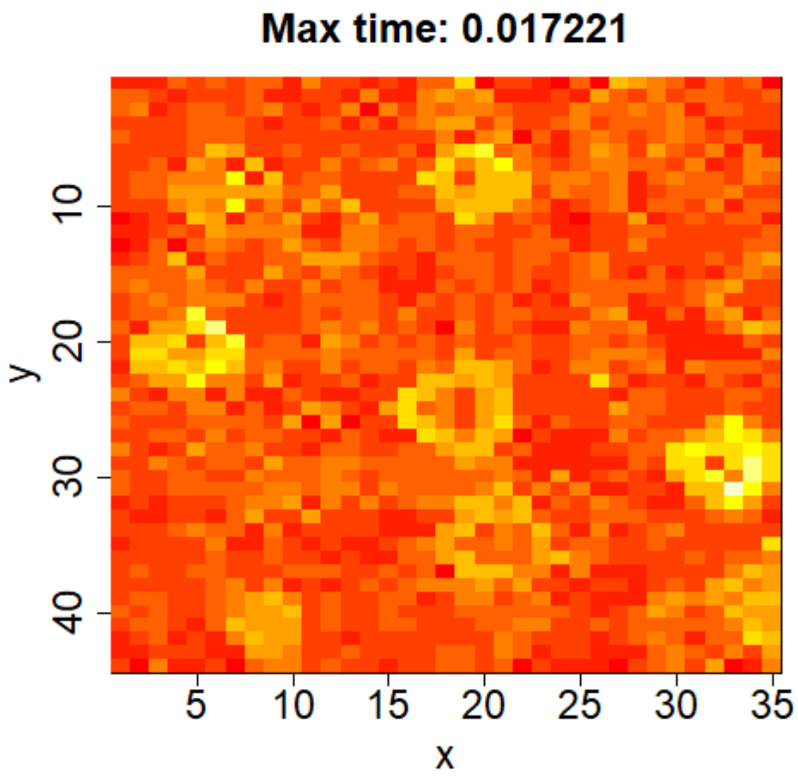


Nearest-neighbor Rank Timing

High Priority

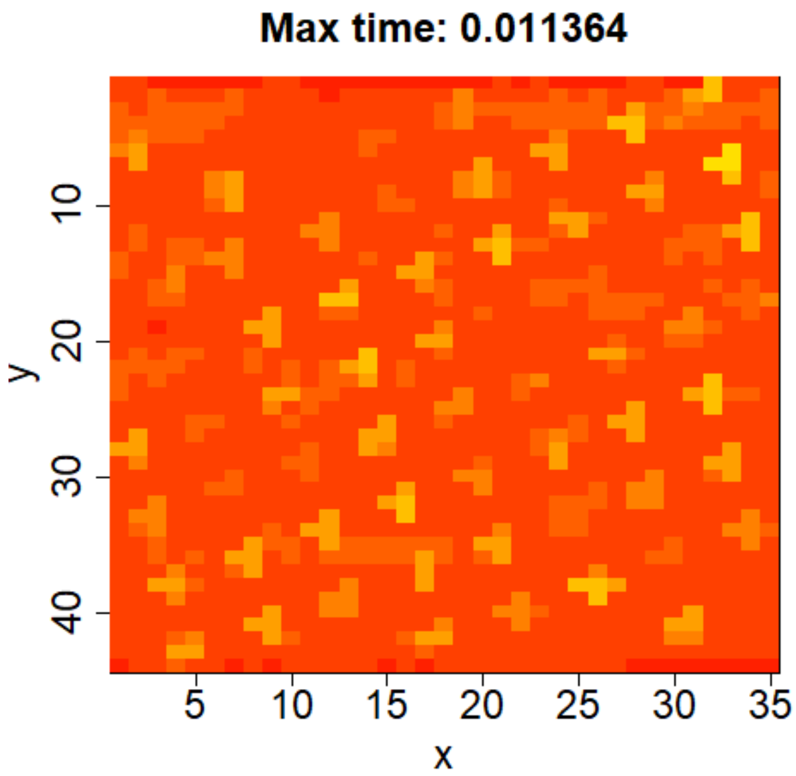


Contended

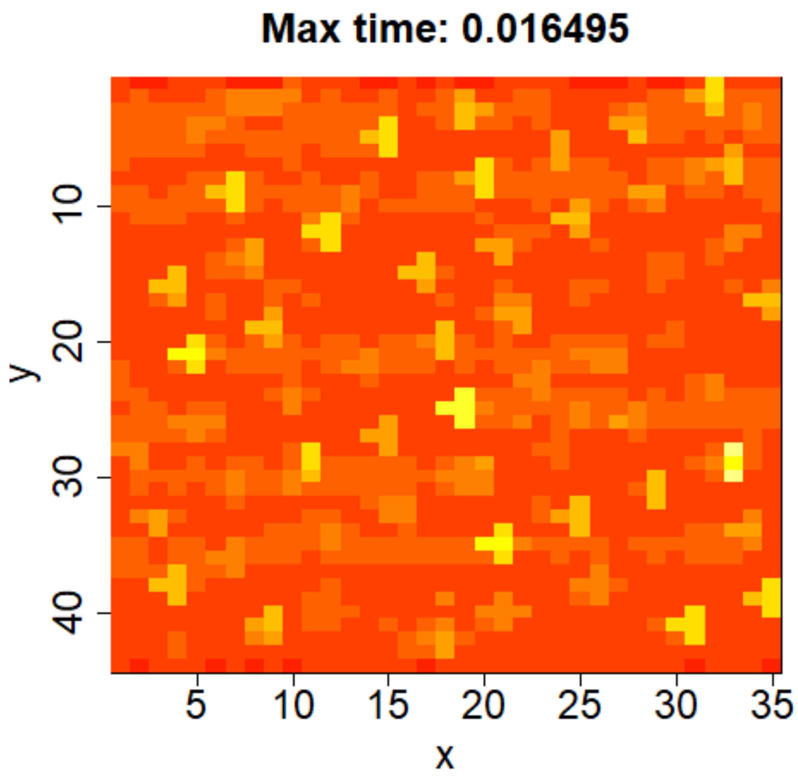


Nearest-neighbor Rank Timing

High Priority



Contended



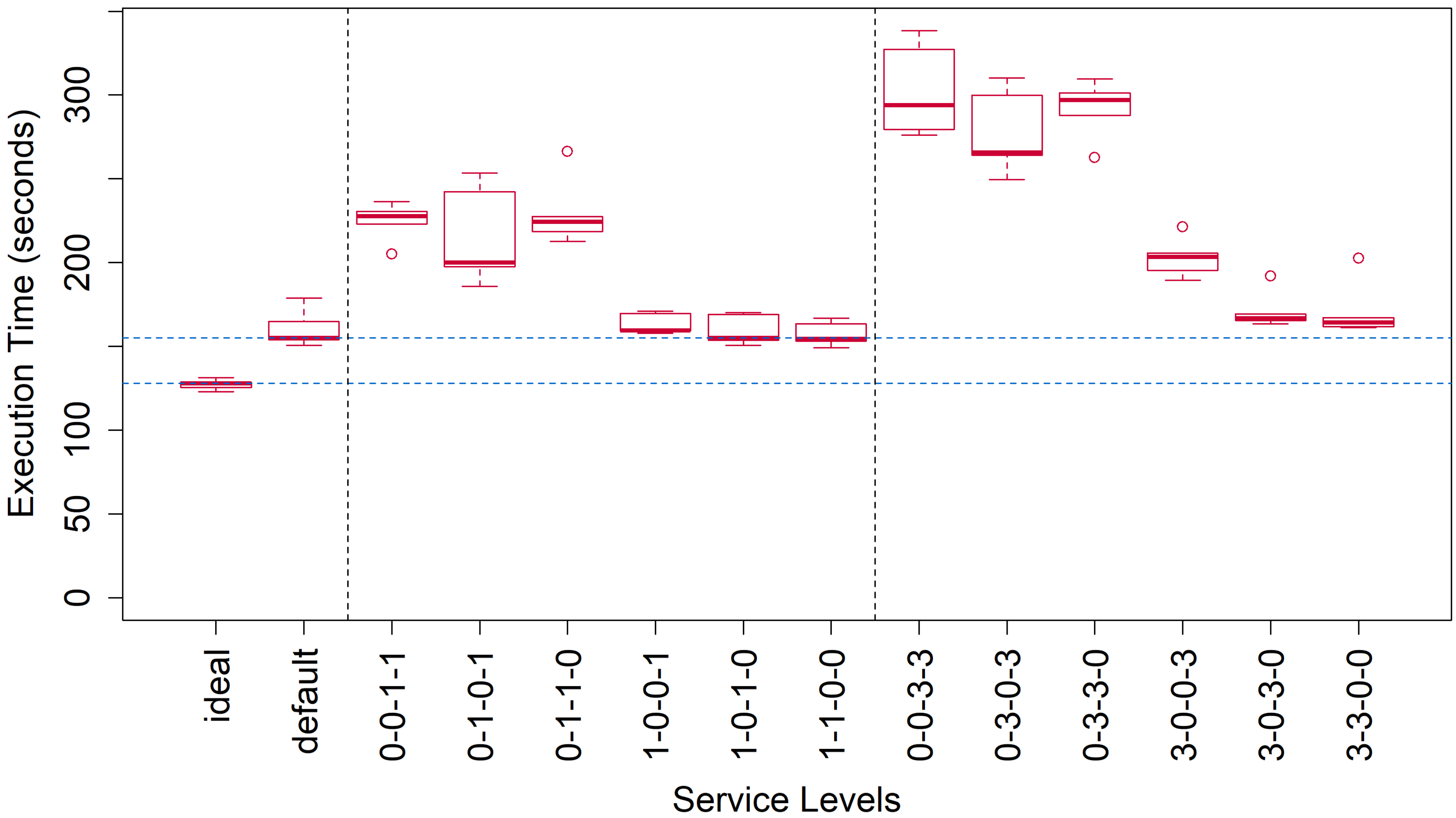
Per-Rank QoS

- Prioritizing an entire job gives high priority to some ranks that are already fast.
- This slows down other jobs, erasing any throughput improvement.
- What if we prioritize only the slowest ranks?
  - Requires prioritizing only ~10% of ranks
  - Same performance as prioritizing the entire job
  - Expect significant reduction in impact on other jobs
- This is the subject of ongoing research

Related Work

- QoS has been studied for a long time
- Jokanovic et al. (2012) came to opposite conclusions
  - Segregate jobs into SLs with different priorities
  - 59% contention reduction
  - Possible reasons for the difference:
    - Simulation vs hardware
    - Future vs current hardware
    - Different service levels

Different Service Levels



- QoS in HPC deserves more research

Conclusion

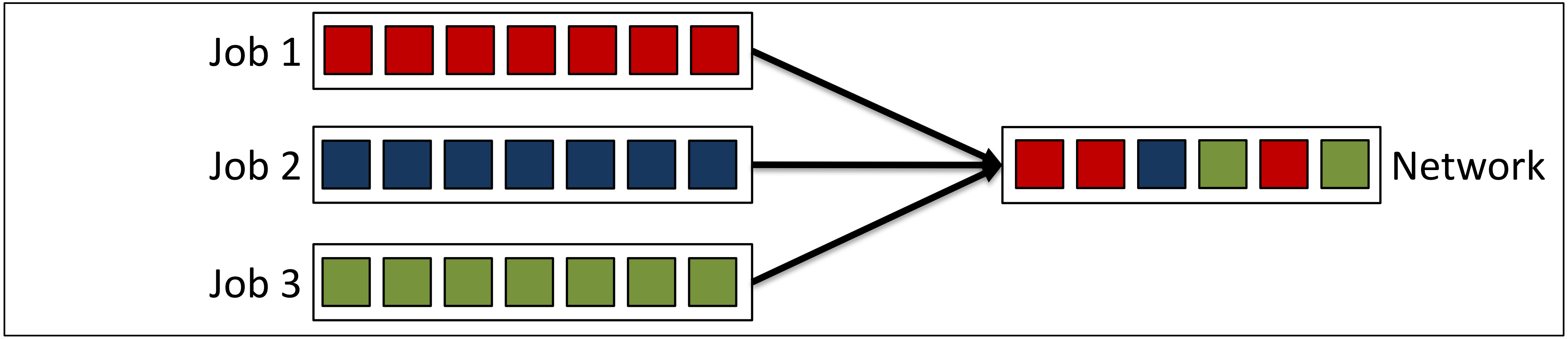
- Many-core nodes will require efficient networks to move data around
- Simple, per-job QoS is unlikely to improve performance
  - Differs from previous work
- Per-rank QoS is more promising
- Further research is needed to understand QoS in HPC

lsavoie@cs.arizona.edu
http://www.cs.arizona.edu/people/lsavoie/

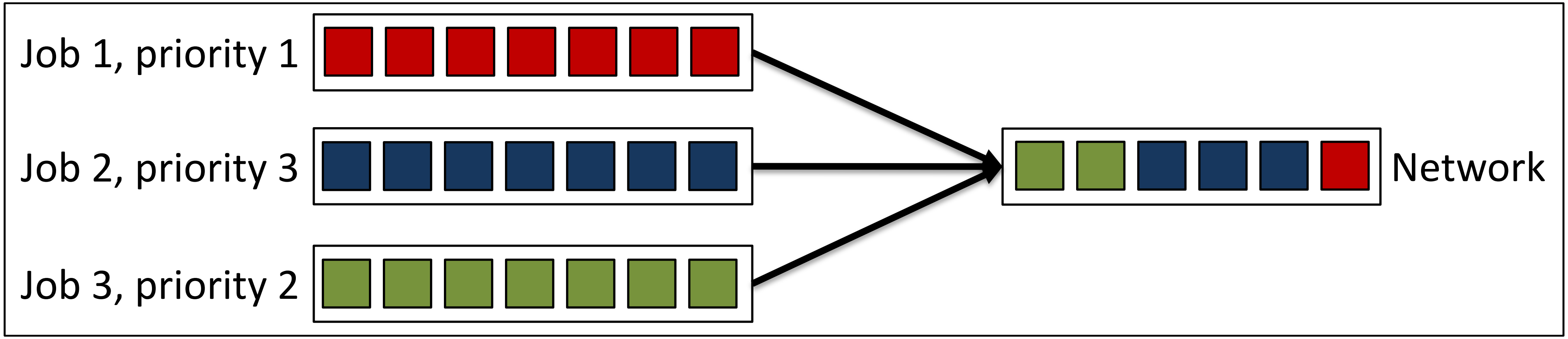
Backup

Per-Job QoS

No QoS:



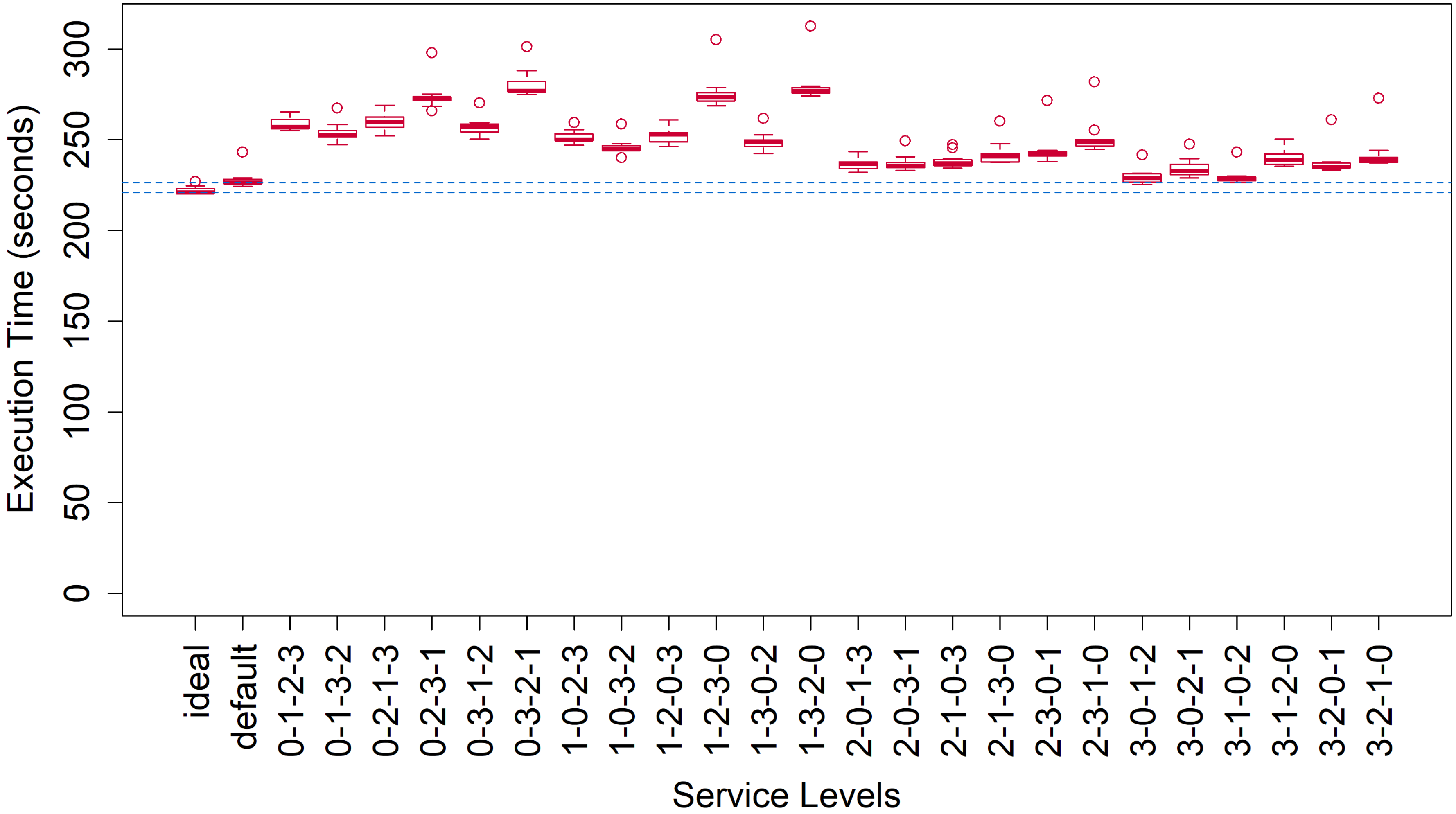
QoS:



Related Work

- QoS has been applied to:
  - The internet [Blake 1998]
  - Video streaming [Ke 2005, Kumwilaisak 2003]
  - Clouds and data centers [Voith 2012]
  - Wireless networks [Andrews 2001]
- Divide traffic across SLs with the same priority to avoid head of line blocking [Subramoni 2010, Guay 2011]
  - We use service levels with different priorities
- Other methods of dealing with contention
  - Adaptive routing [Jain 2014]
  - Job placement [Yang 2016, Jokanovic 2015]
  - These methods are complimentary to ours and insufficient on their own

Results: Applications



- Per-job QoS is insufficient to improve performance.