HP-UX 11i Knowledge-on-Demand

HP technical Webcast series: software optimization

Technology for better business outcomes

HP-UX 11i v3 Knowledge-on-Demand

- Objective: Support software development partners and customers in achieving better business outcomes with HP-UX 11i.

- What HP is providing: a series of technical on-demand training Webcasts

  – Focused on helping developers increase performance through application optimization for HP-UX 11i v3 on HP Integrity servers

  – Access to HP for follow-up questions

  – Available at www.hp.com/go/knowledgeondemand

HP-UX 11i v3 Knowledge-on-Demand Webinars – planned curriculum

- Foundation Track
  - Module 1: How to upgrade to HP-UX 11i v3
  - Module 2: HP-UX open source resources
  - Module 3: Unified file cache
  - Module 4: Caliper
  - Module 5: NUMA Tuning: Getting the Most Out of Your Cellular Server by using NUMA
  - Module 6: The Mercury Library – Increasing Application Performance
  - Module 7: Software Transition Kit's (STK's) for HP-UX 11i v3
- Java Developers Track
  - Module 8: Java Memory Management - Internals and Performance
  - Module 9: HPjmeter – measure Java application performance on HP-UX 11i
  - Module 10: Solving Java performance problems
- C/C++ Developers Track
  - Module 11: pthreads enhancements in HP-UX 11i v3
  - Module 12: Kernel tracing & profiling tools (internal tools)
  - Module 13: Using compilers to get optimal performance
  - Module 14: HP Code Advisor: A Powerful New C/C++ Analysis Tool for HP-UX
  - Module 15: Montecito Hyper-Threading on HP-UX 11i v3

Additional Webinars published going forward!

HP-UX 11i v3 Training

Related HP-UX 11i v3 resources

- All developers' resources
  - HP-UX 11i developers' content
    www.hp.com/go/hpuxdev
  - HP-UX 11i v3 news, functionality, product download and services resources
    www.hp.com/go/hpux11i
  - HP Integrity server ISV resources for DSPP members
    www.hp.com/go/dspp_integrity
  - HP Integrity server product information
    www.hp.com/go/integrity

- Software partner promotional opportunity
  - HP promotion for HP-UX 11i v3-ready software partner application
    www.hp.com/go/v3promotion

hp invent

Enjoy this Knowledge-on-Demand topic!

Thank you for taking time to learn about HP-UX 11i v3 and related technologies.

Please send comments on today's topic and/or requests for future topics to:

hpuxquestions@hp.com

NUMA tuning

An HP-UX 11i Knowledge-on-Demand software optimization Webcast

Technology for better business outcomes

Introducing today's speaker

- Rajesh Desai is a senior engineer in HP-UX kernel development.

Agenda

- What is NUMA?

- When to use NUMA

- How to deploy applications with NUMA

- How to develop applications for NUMA

- NUMA roadmap—a glimpse

Current status

- NUMA is a memory technology used to build large memory systems. HP-UX cellular servers are built using NUMA memory:
  - In 11i v2, HP-UX introduced a configuration option called Cell Local Memory that allows users to take advantage of this technology.
  - In 11i v3, HP-UX kernel memory has been optimized for NUMA.
- Benchmarking shows across-the-board benchmark performance improvements. Key highlights:
  - No user application changes are required.
  - Oracle support is already available.
- But NUMA is not general-purpose OR out-of-the-box:
  - It can only help some customers.
  - It needs services.
- More NUMA capabilities are being considered:
  - Partition configuration support
  - Tools support

NUMA performance data

Benchmark	vs ILM	vs IBM	% CLM
TPC-C high-end	+20%	+1.5%	90%
TPC-C low-end	+50%	~0%	90%
TPC-H	+10%	~0%	50%
SAP—2-tier	+50%	+30%	90%
specJBB—multiple AS	+20%	~0%	90%

- All data is based on a 64-socket HP Superdome.
- SAP and specJBB are projections from smaller configurations.
- TPC-C low-end is an estimation of 4-socket soft partition with NUMA-based CPU assignments and CLM configuration compared to a 4-socket soft partition on a vanilla HP Integrity Superdome.
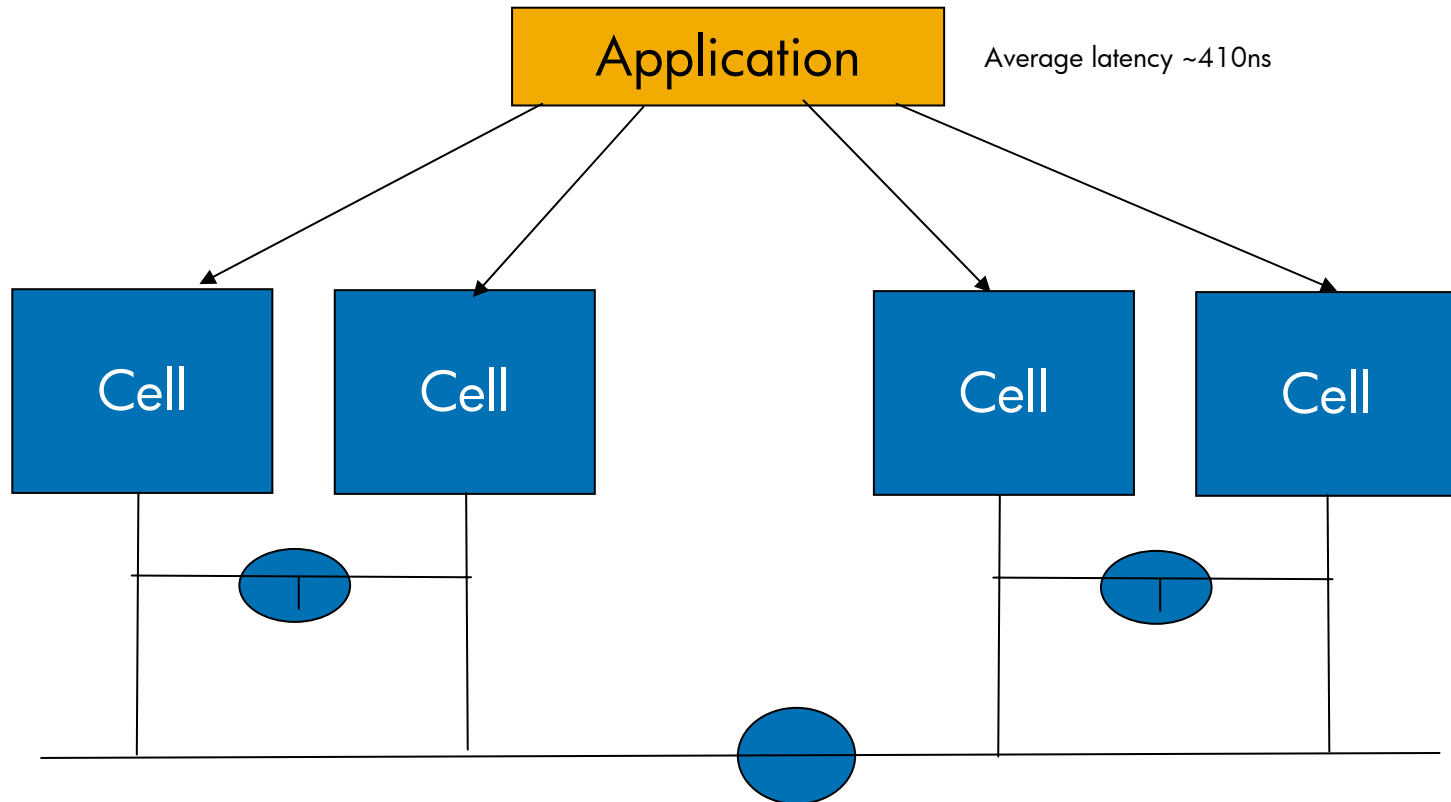
Technology overview—hardware

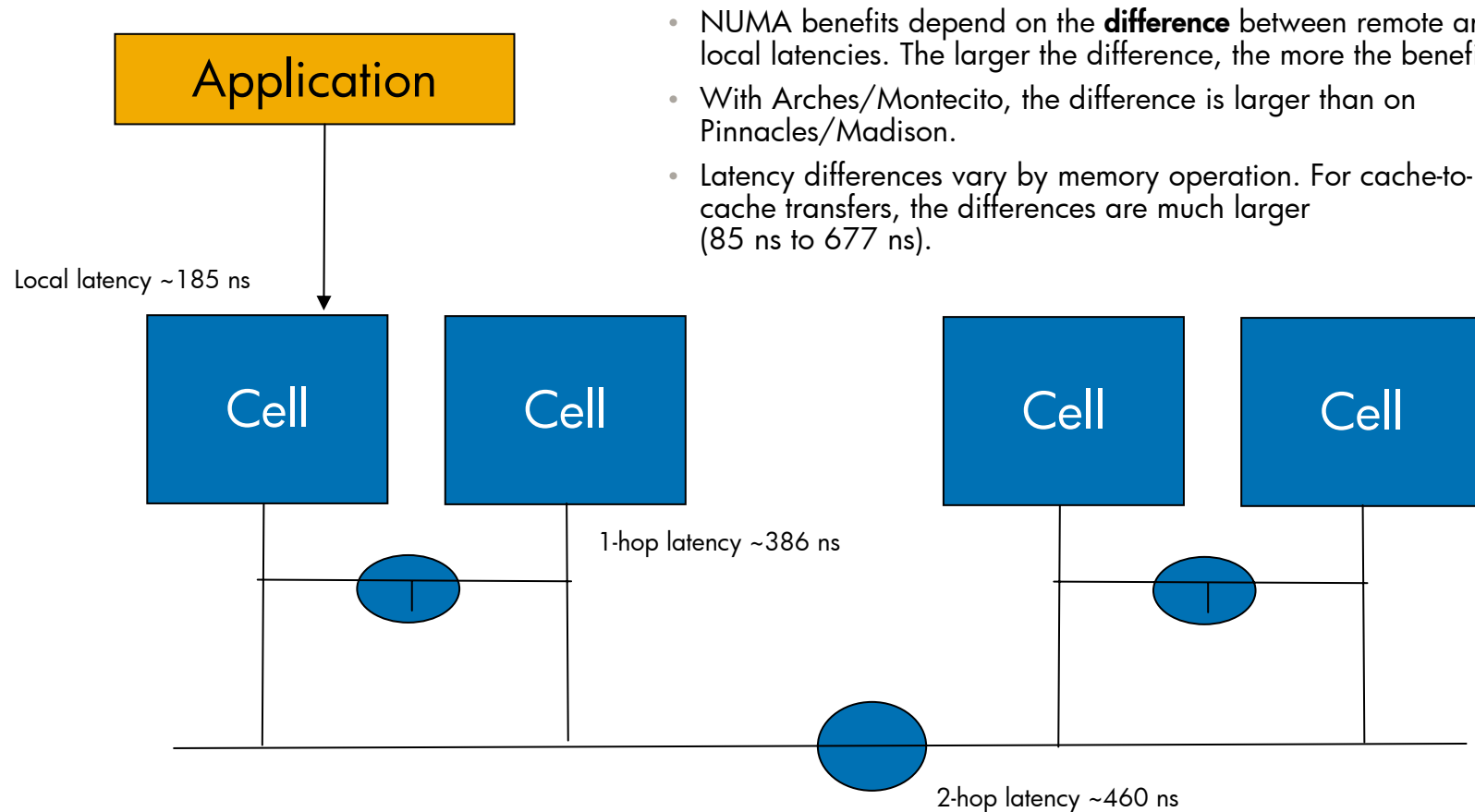Local: 185 ns
1-hop: 386 ns
2-hop: 460 ns

Local latency 185 ns

Cell   Cell   Cell   Cell

1-hop latency ~386 ns

2-hop latency ~460 ns
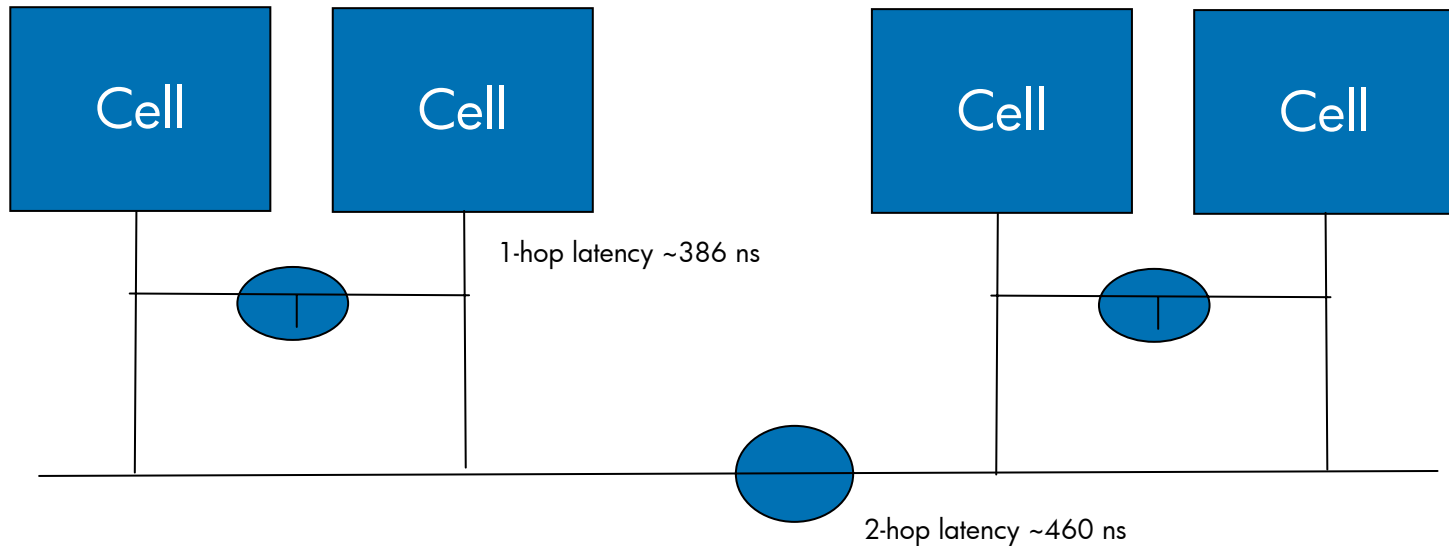
Technology overview—interleaved

Application

Average latency ~410ns

Cell

Cell

Cell

Cell

Technology overview—NUMA

Application

- NUMA benefits depend on the **difference** between remote and local latencies. The larger the difference, the more the benefit.
- With Arches/Montecito, the difference is larger than on Pinnacles/Madison.
- Latency differences vary by memory operation. For cache-to-cache transfers, the differences are much larger (85 ns to 677 ns).

Local latency ~185 ns

Cell   Cell   Cell   Cell

1-hop latency ~386 ns

2-hop latency ~460 ns

NUMA, pros and cons

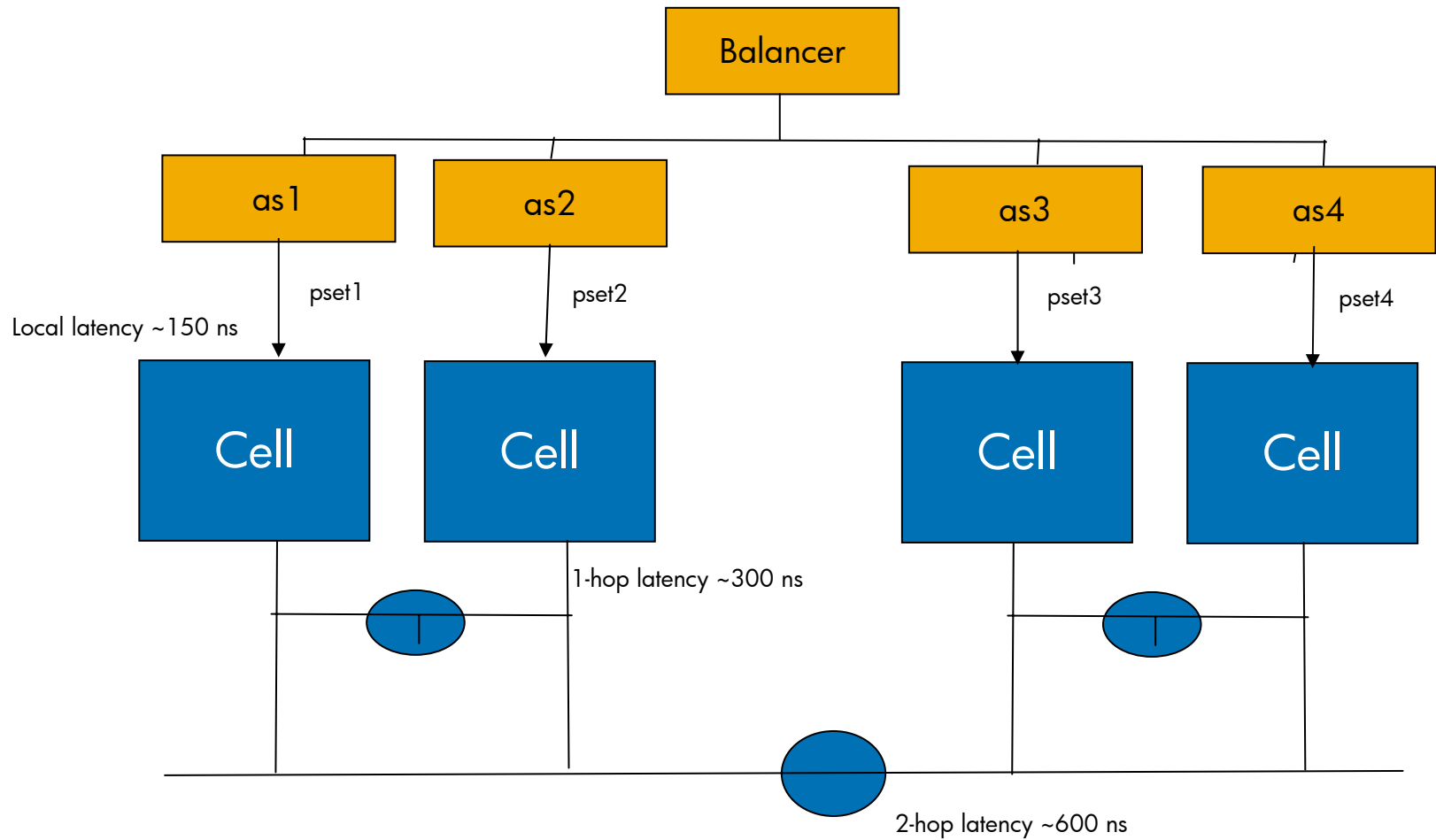| Pros	Cons
• Improved customer satisfaction; leverages existing HW capability untapped by customers • Improved HP-UX sales; 11.23 installed base get more reasons to switch to 11.31 • Improved Integrity sales; closes gap between customer gain and benchmark numbers	• Breaks the traditional SMP programming model • Memory latency becomes a problem that must be taken into account when designing large applications that require a lot of shared data.

Agenda

- What is NUMA?

- **When to use NUMA**
  - Case studies
  - What applications will benefit from NUMA?
  - What servers should be configured with NUMA?

- How to deploy applications with NUMA

- How to develop applications for NUMA

- NUMA roadmap—a glimpse

J2EE case study

Balancer

as1   as2   as3   as4

pset1   pset2   pset3   pset4

Local latency ~150 ns

Cell   Cell   Cell   Cell

1-hop latency ~300 ns

2-hop latency ~600 ns

TPC-C case study

- Start the database with default settings; don't try to outsmart it!
- Start one listener process per locality domain.
- Give the listener processes the HP-UX 'packed' launch policy.
- Distribute the load evenly among the listener processes (using Oracle load balancing or application-level load balancing).
- The server processes will stay in their locality domain because they inherit the 'packed' policy.
- It works well for TPC-C because all clients connect at the beginning of the benchmark and stay connected for the duration of the benchmark.
- Allocate enough CLM to hold the server processes. If the OS runs out of local memory for an SGA and has to allocate space from another cell or ILM, performance can degrade.
- Lock the dbwr processes and the lgwr process in their own CPUs within the locality domain where they were started.

Application segment

- Ideal for Oracle applications—e.g., SAP
  - Use NUMA option if Oracle 10g R2 is in use
- Ideal for server consolidation
  - Use processor set for each workload
  - Build processor sets to minimize number of localities
- Ideal for PRM/WLM with small workloads
  - PRM does select CPUs to minimize number of localities
- Not recommended for large applications—especially ones that are highly threaded or use System V shared memory

Server segment

- Use "static" cellular servers
  - No NUMA support with gWLM
  - No NUMA support with iCAP
- Use with robust memory configurations
  - Memory pressure in one cell can cause performance degradation even if there is memory available on the rest of the system.

Agenda

- What is NUMA?
- When to use NUMA
- **How to deploy applications with NUMA**
  - Partition configuration
  - Application launching
  - Load balancing
- How to develop applications for NUMA
- NUMA roadmap—a glimpse

Partition configuration

- How much CLM to configure?
  - It may be worthwhile to try multiple settings:
    - With vPars—100%
    - For the rest:
      - Configuring a little bit (1/4) would speed up the access to private data (e.g., stack, heap)
      - Configuring a lot (7/8) would speed up the access to shared data and private data (e.g., system V shared memory)

- How to configure:
  - parmodify
  - vparmodify

Load balancing

- For server consolidation—use PRM:
  - PRM is already optimized for NUMA.
- For Oracle:
  - See slides by Tor at the back of this presentation.
- For Java Application Server:
  - See Tom Benson's presentation.
- For SAP:
  - A white paper will be coming to a Web site near you.

Application launching

- Enable NUMA allocations:
  - Oracle options
  - New kernel tunable to avoid source code changes
- Create locality-aware processor sets:
  - mpsched –s
  - psrset –c
- Launch applications in appropriate processor set:
  - psrset –b
  - mpsched –P

How to develop for NUMA

- Application ISVs—no changes are needed for most applications:
  - SAP did not need any changes
- Middleware ISVs—may need to use HP-UX APIs:
  - Options to madvise(), mmap(), fadvise(), shmget()
  - Chatr options

Agenda

- What is NUMA?
- When to use NUMA
- How to deploy applications with NUMA
- How to develop applications for NUMA
- **NUMA roadmap—a glimpse**

VSE support

- Currently, partition configuration tools reduce TCO by minimizing the number of CPUs needed from a partition.

- The opportunity is to further reduce TCO by selecting the **right** CPUs to add and remove from a partition:
  - NUMA extensions to iCAP
  - NUMA extensions to gWLM

Reduce memory requirements

- Currently, NUMA delivers higher performance, but it may use extra memory to ensure there is no paging or short-term imbalance of memory requirement across cells.

- The goal is to implement robust NUMA features like those in Tru64 and SGI to ensure that we do not need extra memory:
  - User memory migration
  - NUMA-aware pager

Reduce services needed

- Currently, NUMA needs specialized server configuration services in order to be deployed.

- The goal is to make NUMA configuration easy enough that it can be deployed with something as simple as a boot option.

Summary

- NUMA improves performance.
- NUMA is not general purpose.
- NUMA does not need a new application version.
- NUMA needs services.
- More NUMA features in the product are on the way.