



SunNetwork
2003 Conference and Pavilion

TUNING DB2 IN A SOLARIS ENVIRONMENT

Tom Bauch
Technical Lead, IBM

Jignesh Shah
Software Engineer, Sun Microsystems

<http://sun.com/db2>



About Tom Bauch

- Senior Consulting IT Specialist, responsible for pre-sales technical support of IBM's 'DB2 Universal Database on Sun Solaris'
- Currently the technical lead for the DB2 Ranger Team
- Co-author of the upcoming book DB2 V8.1 on Solaris; The Official Guide
- In 2000, was a co-author of the book DB2 in the Solaris Operating Environment
- In 1999, helped develop and teach the UDB EEE for Windows NT course
- From 1995 through 1998, was an instructor with IBM Education and Training, in the OS/2 and DB2 UDB curriculums, teaching OS/2 Warp and Warp Server as well as DB2 UDB Administration on OS/2 and NT, net.data, and the UDB Extenders course
- Been with IBM for 23 years, serving in a variety of technical support roles, from mainframe VSE/VM, VTAM/NCP, SQL/DS, and DLI to OS/2 Warp, DB2/2, and Warp Server; Spent 7 years on international assignment supporting IBM systems for the US government in Europe

About Jignesh Shah

- IBM Certified Database Administrator—DB2 UDB V8.1
- IBM Certified Solution Designer—DB2 Content Manager V8
- Sun Certified System Administrator for Solaris 8
- Sun Certified Network Administrator for Solaris 8
- Sun Certified Programmer for the Java 2 Platform
- Sun Certified Developer for the Java 2 Platform
- Sun Certified Web Component Developer for the J2EE Platform
- Working on tight integration and performance enhancement of DB2 on Solaris Operating Environment
- Previously worked on various ERP implementation and their performance on Solaris and other flavors

Agenda

- General Considerations
- Memory Usage and Bottlenecks
- Disks and I/O Considerations and Bottlenecks
- Tuning DB2 V8.1 in Solaris 9
- Other Stuff...

You Are Here

- General Considerations
- Memory Usage and Bottlenecks
- Disks and I/O Considerations and Bottlenecks
- Tuning DB2 V8.1 in Solaris 9
- Other Stuff...

Supported Solaris Versions

- Refer to the Quick Beginnings manual and Fixpak Readme's for patch requirements
- DB2 UDB V6.1 is supported on the following Solaris versions:
 - Solaris 2.5.1, 2.6, 7 and 8 (Fixpak 3 and above)
- DB2 UDB V7.1 and 7.2 are supported on the following Solaris versions:
 - Solaris 2.6, 7, 8 and 9
 - 32 bit and 64 bit DB version in shrink-wrap or 64 bit version via Fixpak
 - 32 bit DB tested and supported on Solaris 7, 8 in 64-bit O/S mode
- DB2 UDB V8 is supported on the following Solaris versions:
 - Solaris 7, 8 and 9

Supported Solaris Matrix

	Solaris 2.5.1	Solaris 2.6	Solaris 7	Solaris 8	Solaris 9
DB2 V6*	Build Level	Supported	Supported	Supported	Not Supported
DB2 V7.1, V7.2	Not Supported	Build Level	Supported	Supported	Supported
DB2 V8.1	Not Supported	Not Supported	Build Level	Supported	Supported

Note 1: V6 is no longer being distributed and is only supported with a special support agreement

Kernel Settings

- Kernel settings tool (db2osconf) should be used on the machine where DB2 will be running for initial kernel suggestions
 - Sum of all applications may be required if multiple apps sharing the domain/server
 - Restart the system (you knew that...)
 - DB2 instance creation checks for certain minimum values, and will fail if those values are not set

Patches

- Patches:
 - DB2 Workgroup Server Edition is supported on:
 - Solaris 7 (32-bit) + 107226-17 + 107153-01 + 106327-10
 - Solaris 8 (32-bit) + 108921-12 + 108940-24 + 108434-03 and 108528-12
 - Solaris 9 (32-bit)
 - DB2 Enterprise Server Edition is also supported on:
 - Solaris 7 (64-bit) + 107226-17 + 107153-01 + 106300-11
 - Solaris 8 (64-bit) + 108921-12 + 108940-24 + 108435-03 and 108528-12
 - Solaris 9 (64-bit)
 - All should have “Recommended and Security Patches”
- The J2SE Solaris Patch Clusters are also required, and the SUNWlibC software is required to install DB2 on Solaris (this is included in the prerequisite patches above)

Installation Methods

- Two options for installing DB2 on Solaris
 - Choose the code you wish to install; server, client, connect
- `db2_install`
 - Runs Solaris 'pkgadd' to install DB2 executable code
 - Puts DB2 code into `/opt/IBM/db2/V8.1` with option to place binaries elsewhere
 - Requires root access, DB2 tailoring done after install
 - `db2_install` script does not get installed

Installation Methods

- db2setup
 - Java-based GUI
 - Two basic parts
 - Part 1: Install DB2—uses pkgadd
 - Part 2: Tailor install—creates users, groups, instances
 - kernel configuration and reboot necessary before part 2
 - Requires root access, db2setup installs into '/opt/IBM/db2/V8.1' only

Before You Start Looking...

- DB2 ‘out of the box’ is tuned for the minimum required system
 - Small memory machine (256 MB)
 - Single disk
- Memory and I/O parameters are optimized for that small environment

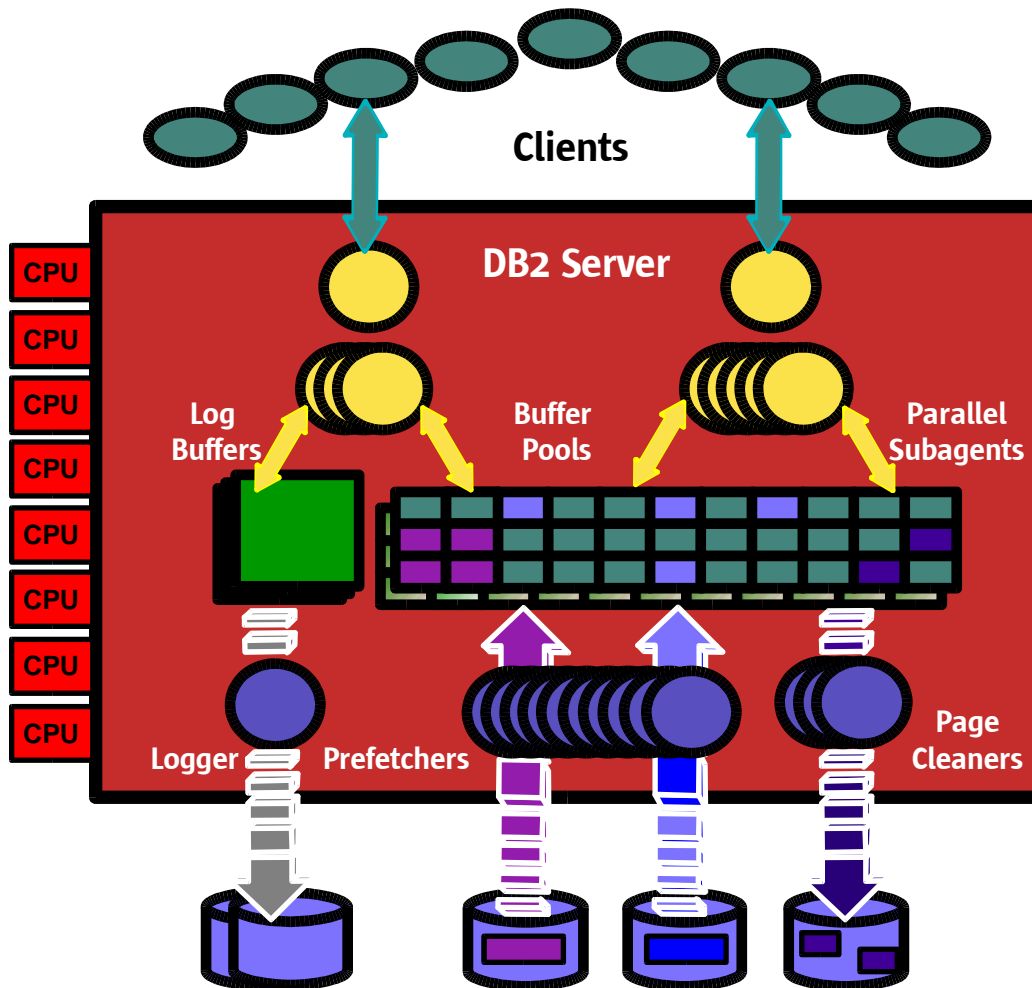
Configuration Wizard

- Shipped with DB2 is a ‘Configuration wizard’
 - Configures database parameters for your environment
 - Analyzes your system based on RAM, disk subsystems, number of processors
 - Should be run prior to starting tuning efforts
 - Allows input to define maximum system resources this database allowed to use
 - Suggestions can be implemented immediately or deferred for your evaluation prior to implementation
- Run the Configuration wizard ‘up front’ to make your baseline appropriate for your system!

What Is DPF Option?

- Data Partitioning Feature
 - Required in order to partition your database,
 - Either within a single server, or across multiple servers
 - The DPF is a license-only and does not require any additional products on top of DB2 UDB ESE to be installed on your database server to support database partitioning

DB2 UDB ESE



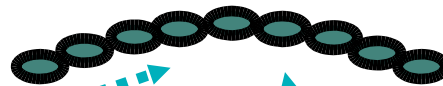
- **Full Parallelism**
 - SQL and utilities
 - Inter- and intra-partition parallelism
 - Cost-based optimizer with query rewrite
 - Dynamic throttling based on load
- **Full SMP Exploitation**
 - All CPUs exploited through OS threads and processes
- **Very Large Memory**
 - 64 bit support
 - I/Os fully buffered
 - Multiple buffer pools
- **Highly optimized I/O Subsystem**
 - Asynchronous
 - Parallel I/O
 - Automatic intelligent data striping with parallel I/O
 - Big block I/O
 - Scatter/Gather I/O

DB2 UDB ESE With DPF

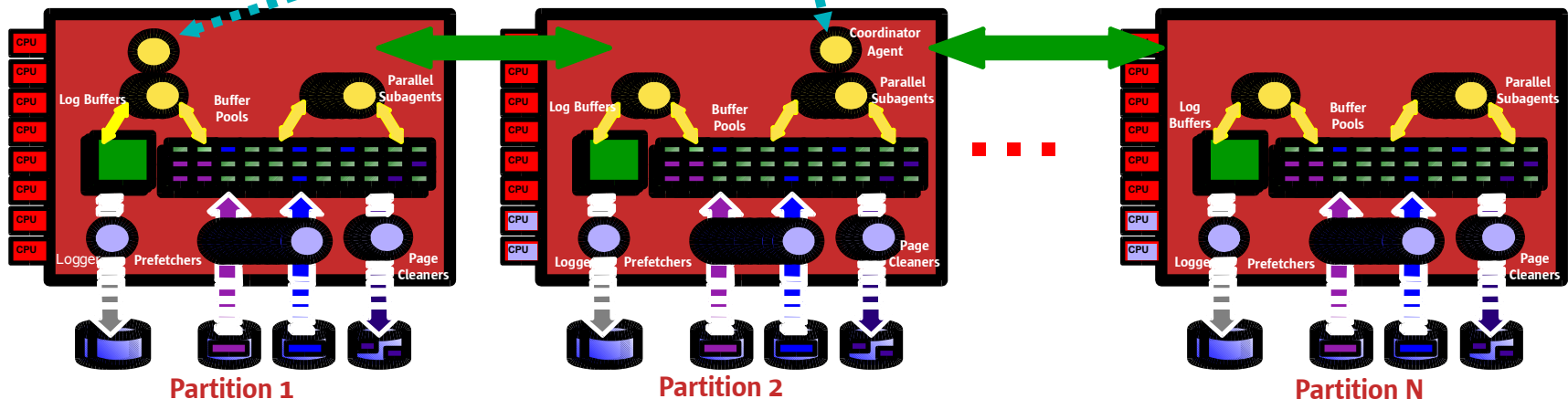
- Shared nothing architecture allows virtually unlimited scalability
 - Each partition owns it's resources (buffer pool, locks, disks, ...)
 - Avoids common limits on scalability:
 - No need for distributed lock manager or buffer coherence protocols
 - No need to attach disks to multiple nodes
- Flexible configuration
 - Any number of partitions can be created on a single physical machine (works extremely well with NUMA architectures)
- Partitions communicate only necessary tuples
 - Using shared memory (when partitions on same machine)
 - Using high speed comm protocol (on different machines) -e.g., VIA, GigaEthernet



SELECT * FROM T1,T2 WHERE...



Single Database View;
Transparent Intra- and Inter-Partition Parallelism



When to Use DPF?

- Large data volume (> 200GB Raw data)
- DSS type applications
- Large volume/connections OLTP
- Enhanced Parallelism (memory, IO — logger and tablespace)
- Need more than 4GB bufferpools in 32-bit instances

You Are Here

- General Considerations
- **Memory Usage and Bottlenecks**
- Disks and I/O Considerations and Bottlenecks
- Tuning DB2 V8.1 in Solaris 9
- Other Stuff...

Global/Instance Memory

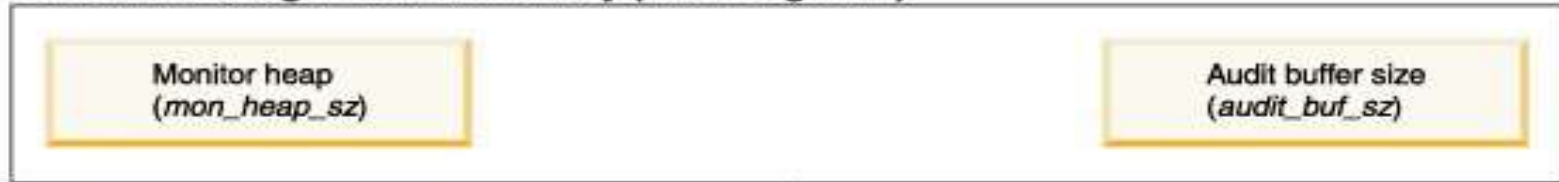
- Database memory
 - Each 32 bit database has a 4 gig limit, about 3.5 gig usable
 - Global memory use is limited by shmmax and is ISM (pinned)
 - 64 bit instance removes 4 gig memory limit
 - 32 bit and 64 bit instances can coexist on the same machine
 - ‘Instance Owning’ userid is the address space used for an instance

Global/Instance Memory

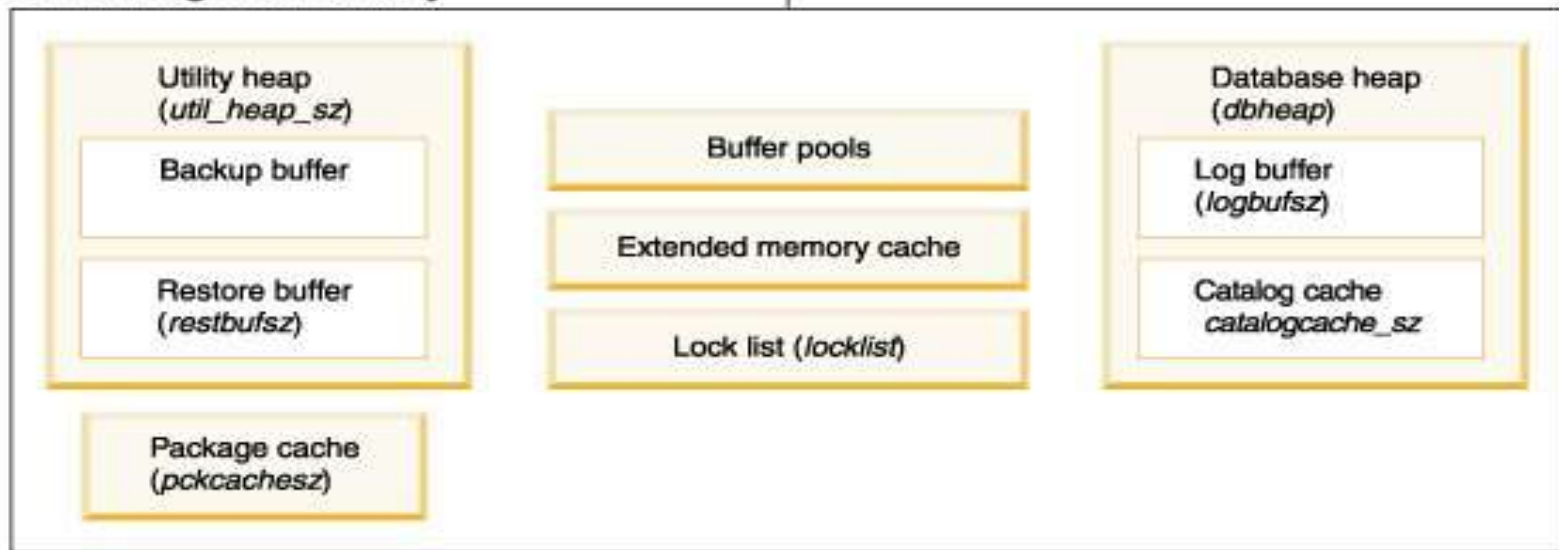
- Client/Agent memory
 - Each connection to the database is a process
 - Many functions within DB2 are processes
 - Logger
 - Prefetchers
 - Bufferpool cleaners
 - Other
 - Each agent/process requires a semaphore set
 - Each agent has private and shared memory
 - Agent shared memory is also ISM (pinned)

Memory Usage

Database manager shared memory (including FCM)



Database global memory

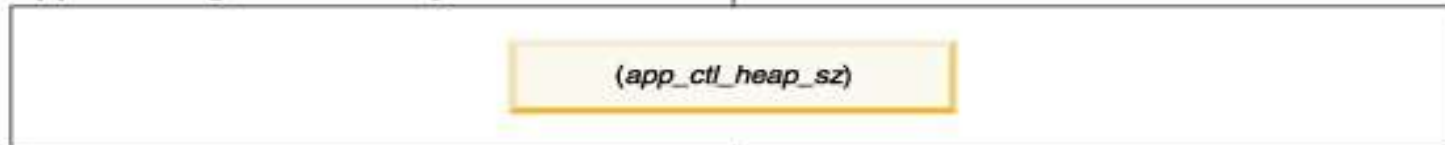


Application global memory

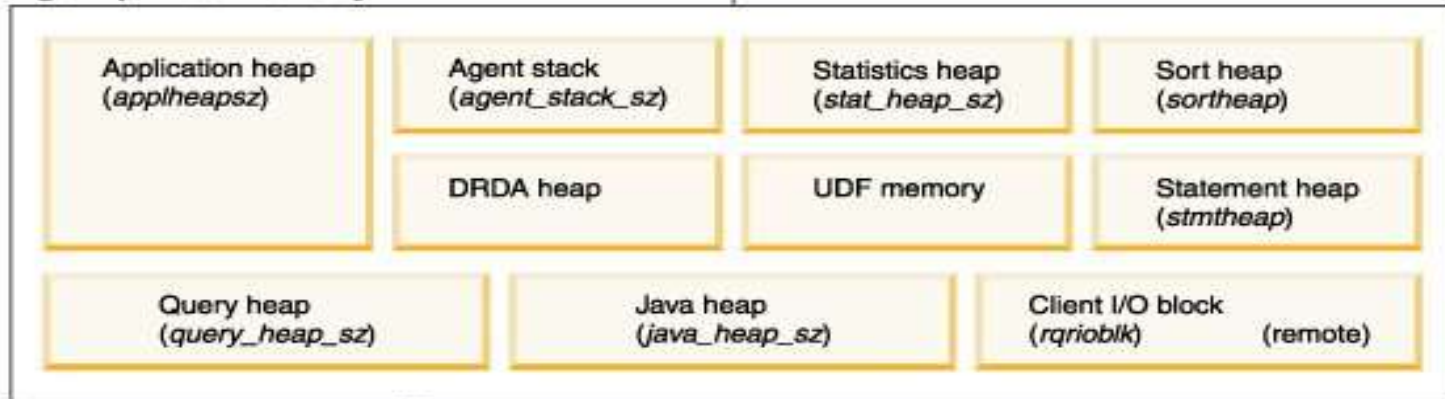


Memory Usage

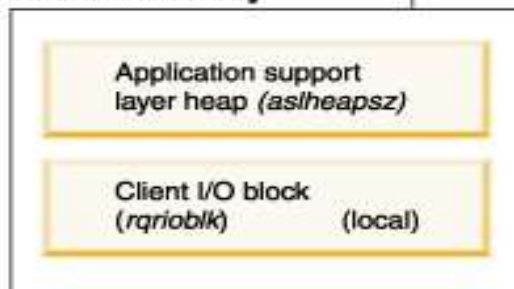
Application global memory



Agent private memory

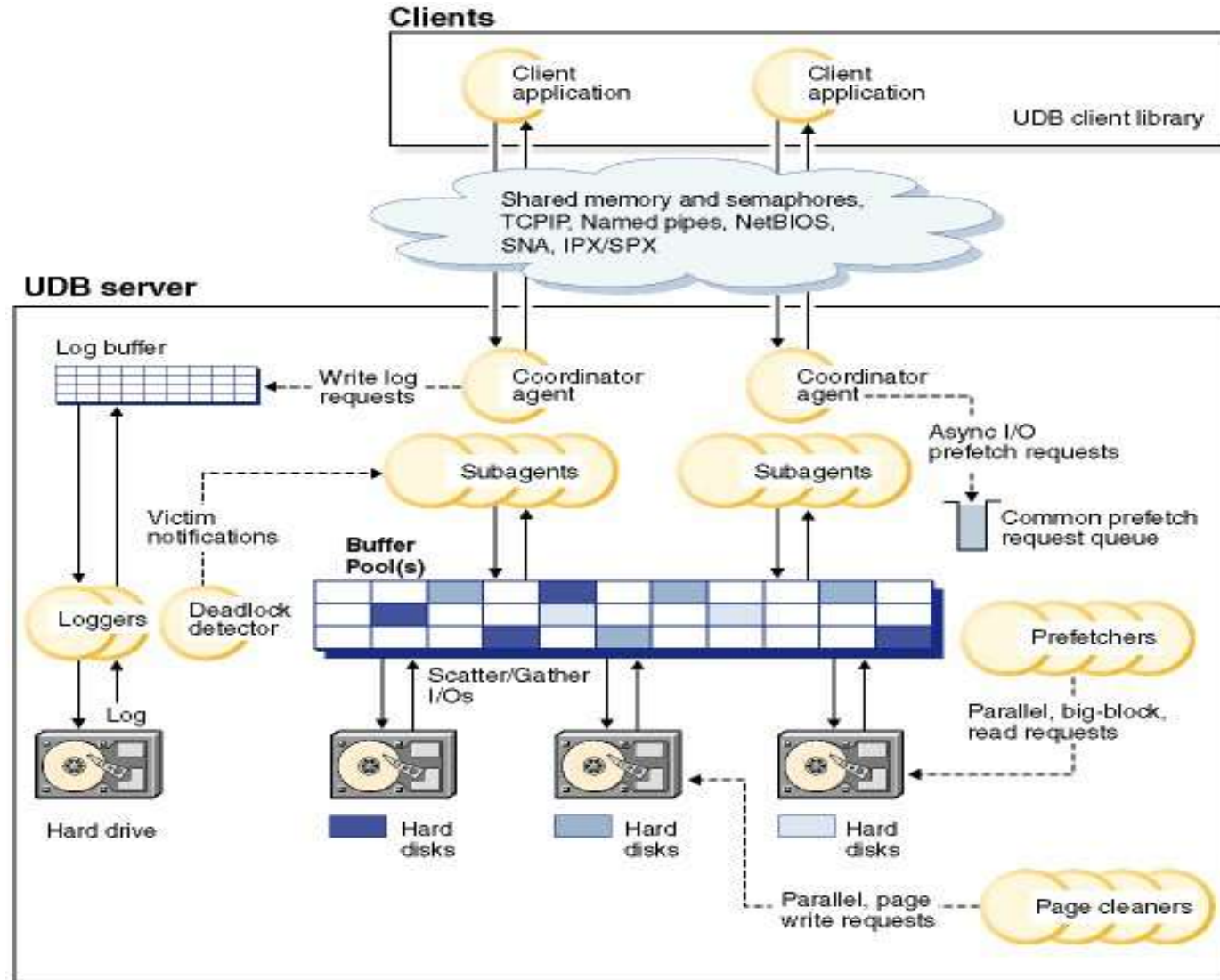


Agent/Application shared memory



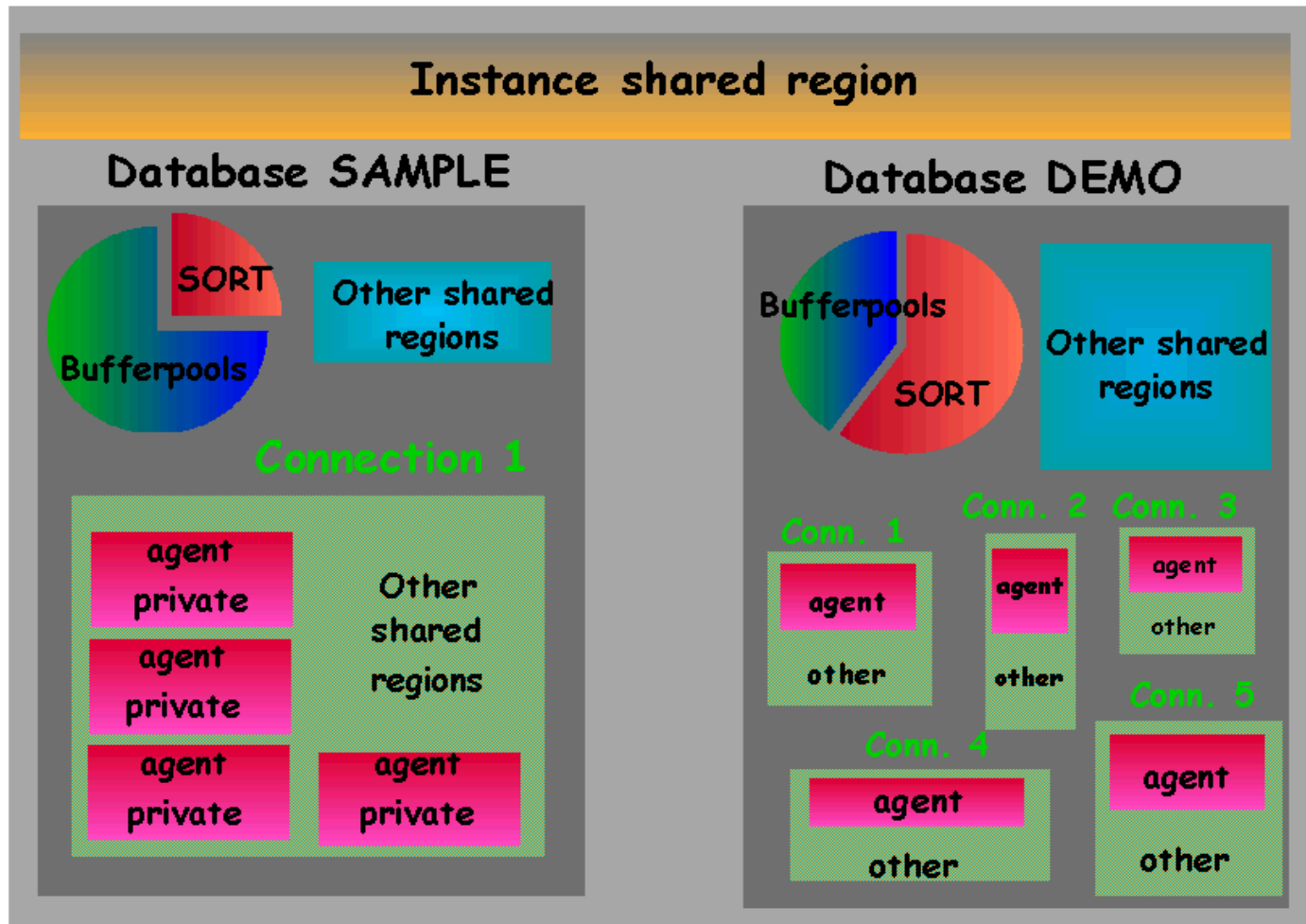
Note: Box size does not indicate relative size of memory.

Process Model



Memory Logical View

Instance db2inst1



Memory Bottleneck

- Not enough
 - DB2 can use all the memory you can afford
 - Tradeoffs are required to determine where the memory should be allocated
 - Usually memory shortage seen as a load on Solaris (i.e., swapping)
- Swapping?
 - ISM usage requires awareness of memory usage
 - Review memory definitions in DB2
 - Buffer pools
 - sortheapthresh
 - If lots of agents (processes), each has private allocation, consume swappable memory, consider using agent pooling
 - Adequate Swap?

Memory Bottleneck

- Slow query performance due to memory?
 - Use DB2 Snapshot tool, or vendor tools to view DB2 memory usage
 - Heavy sorting? Perhaps add indexes
- Memory tradeoff is usually between buffer pools and sorting or utilities
 - Can be dynamically shifted within the shared segment to favor utilities or applications
 - Off-shift memory usage can be shifted to batch style definitions
 - Multiple bufferpools allows flexibility of keeping indexes or specific objects in memory

Memory Bottleneck

- Memory issues seen as slow query performance
 - Use DB2 snapshot tool to see effectiveness of your bufferpools
 - Hit ratio
 - Multiple bufferpools may help
 - Block-based bufferpool definitions can assist in mixed workload systems
 - Look for sorting going to disk (spilling) due to inadequate sortheap
 - Tradeoff between sorting memory and bufferpools
 - Snapshot will indicate what your sorting activity is doing
 - Check the access plan generated by the optimizer
 - ‘Explain’ the queries
 - Look for performance issues, index usage (or not)
 - Multiple sub-agents with high levels of parallelism

You Are Here

- General Considerations
- Memory Usage and Bottlenecks
- Disks and I/O Considerations and Bottlenecks
- Tuning DB2 V8.1 in Solaris 9
- Other Stuff...

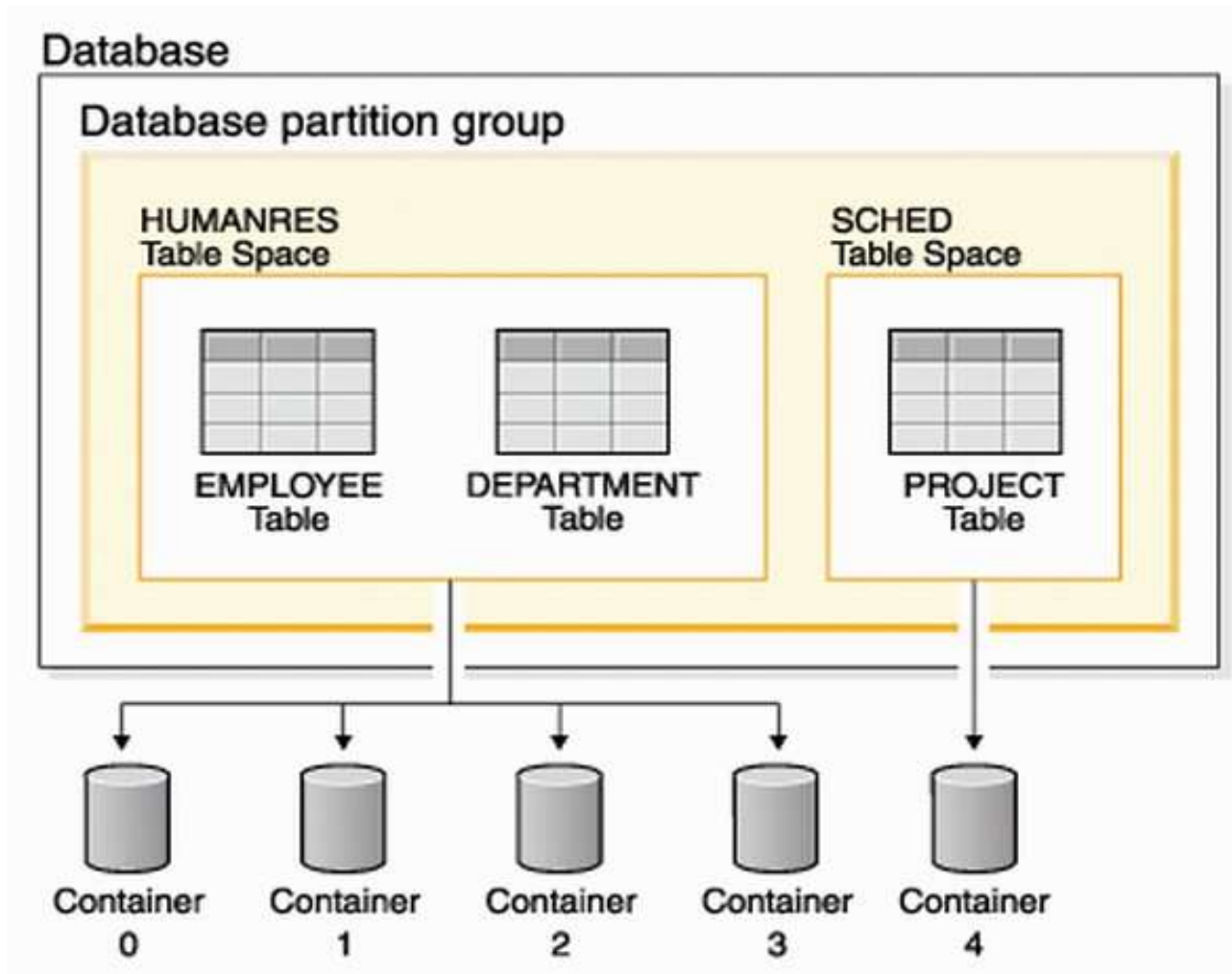
Disk and I/O Considerations

- DB2 I/O and optimization
 - Lots of small disks generally better than a few large disks (same for any DB)
 - SANs and Storage Subsystems are supported, but ‘Memory Cache’ in that subsystem does not necessarily improve database performance
 - DB2 optimizer is ‘I/O subsystem aware’
 - Allows for access plan definition based on actual I/O
 - I/O parallelism is standard, and the amount is dictated by the optimizer
- SAN/Raid Considerations
 - DB2 cannot see the actual disk configuration in a SAN/Raid implementation
 - Registry settings available to provide the optimizer with the necessary information to parallelize I/O activity against SAN/Raid devices

Disk and I/O Considerations

- Disk Layout
 - DB2 layout is defined at the tablespace layout
 - Tablespace are defined by ‘containers’
 - One or multiple containers per tablespace; raw devices, files, or directories
 - DB2 ‘stripes’ data across the containers, in a round-robin fashion
 - Parallel/concurrent access to multiple containers is the default, both within a single sql query and across multiple queries
 - DB2 registry parameters

Disk/Tablespace Layouts



I/O Performance

- Lots of disks
 - DB2's internal striping allows for concurrent access to multiple disks
 - Define tablespaces with containers on all disks available (?)
 - Indexes can be separated from table data (in a different tablespace)
- Tablespaces are associated with a bufferpool
 - To keep an individual table in memory, put it in a separate tablespace, and associate a separate bufferpool with that tablespace
 - Indexes and Table data can use separate tablespaces, which can then be associated with different bufferpools

I/O Performance

- Logging I/O
 - Database logging is a ‘synchronous’ operation
 - Place the logpath on a separate physical disk (if possible) not used by the database
 - A write cache can improve performance on the log, but insure it is equipped with a good battery backup, and the battery is changed frequently
- SAN/RAID
 - Use DB2 facilities to tell the optimizer that tablespaces are not on single disks to enable parallel/concurrent access of raid or SAN devices (multiple concurrent I/O operations)
 - Align DB2 extents (contiguous blocks) with physical disk stripe

Disk Performance

- Always know where your containers really are!
 - Make a drawing of the physical disk to the volume to the mount point
 - Often this layering of definitions can hide the true layout of your disk geometry
 - DB2 has an extremely efficient I/O capability, executing multiple I/O's in parallel, which can cause disk contention if the tablespaces appear to be on separate disks, but behind the scenes are actually on the same physical mechanism...
 - This is true for all databases
- Use `iostat` to identify disk bottlenecks
 - `iostat -xnez 5`
 - Will display devices that have activity, look for disks that have high service times or high busy indicators

Disk Performance

- I/O for DB2 is defined at the tablespace level
 - Make sure that your containers are really on separate disks
 - If SAN or RAID, make sure there are adequate containers to allow DB2 to perform multiple concurrent I/O operations, or that the parallel I/O registry settings are enabled
 - Containers can be added to move the I/O load (make sure you know where they go)
 - Temp space should be defined on multiple disks, make sure they are not really the same disk behind the scenes

You Are Here

- General Considerations
- Memory Usage and Bottlenecks
- Disks and I/O Considerations and Bottlenecks
- **Tuning DB2 V8.1 in Solaris 9**
- Other Stuff...

New Features of Solaris 9 Exploited by DB2 V8.1 fp 4

- **Projects-based Resource Pool Management**
 - Prelude to “Zones” which will be introduced in Solaris 10
 - Uses `/etc/project` and `/etc/pooladm.conf`
 - Can be set in `db2nodes.cfg` (in DB2 V8.1 fixpack 4)
 - Uses processor bindings in background with resource pools, exploits MPO, improves performance, eases management by reporting usage by projects also in `prstat` utility (`prstat -j`)
- **Memory Placement Optimization (U1)**
 - Giving preference to local memory (Sun 12000 and Sun Fire 15000)
 - In `/etc/system` add `set lgrp_mem_pset_aware=0x1`
 - Use projects to bind DB2 nodes to CPU Boards or use custom scripts to bind processes to CPU board
 - Improves performance by improving latency due to memory access

New Features of Solaris 9 Exploited by DB2 V8.1 fp 4

- Use bundled Solaris Volume Manager (specially in SAN environment)
 - Map containers to soft volume partitions using Solaris Volume Manager (useful if you use RAID Controllers where it presents itself as a big LUN to DB2)
 - Now you can use iostat tool to get IO rates per container (Solaris 9 iostat tool shows all SVM volumes also)
 - Makes analyzing problems easier as you can get IO information by containers in iostat

Example Tuning in Solaris 9

- Say you have a DB2 V8.1 with Solaris 9 update 3 on a single domain of a 24-way Sun Fire 6800
- You have multiple partitions or instances running on same physical server
- You have tuned your DB2 and Solaris kernel parameters as advised by db2osconf and the Configuration Wizard
- You are looking at Solaris to give you some extra boost to your system and make management more easier



Example Tuning in Solaris 9

- Use the Project database and resource pools to manage resources for each partitions of instances
 - Uses processor bindings in background
 - Exploits Memory Placement Optimization(MPO)
 - Creates virtual isolation of instances/partitions
 - Can now track individual project resource consumption

Setting Up Projects

- Add to `/etc/system` to make Solaris local memory aware
 - `set lgrp_mem_pset_aware=0x1`
- Modify `/etc/project`
 - `system:0::::`
 - `user.root:1::::`
 - `noproject:2::::`
 - `default:3::::`
 - `group.staff:10::::`
 - `db1_0:5000:DB2 Instance 1 Node 0 project:db2inst1,root::project.pool=db0_pool`
 - `db1_1:5001:DB2 Instance 1 Node 1 project:db2inst1,root::project.pool=db1_pool`
 - `db2:5002:DB2 Instance 2 project:db2inst2,root::project.pool=db2_pool`
 - `db3:5003:DB2 Instance 3 project:db2inst3,root::project.pool=db3_pool`
 - `db4:5004:DB2 Instance 4 project:db2inst4,root::project.pool=db4_pool`
- To identify your current project
 - `$ id -p`

Setting Up Resource Pools

- Create a pool.host text file
 - create system myhost
 - create pset pset_default (uint pset.min = 1)
 - create pset dbo_pset (uint pset.min = 4; uint pset.max = 4)
 - create pset db1_pset (uint pset.min = 4; uint pset.max = 4)
 - create pset db2_pset (uint pset.min = 4; uint pset.max = 4)
 - create pset db3_pset (uint pset.min = 4; uint pset.max = 4)
 - create pset db4_pset (uint pset.min = 4; uint pset.max = 4)
 - create pool pool_default (string pool.scheduler="TS"; boolean pool.default = true)
 - create pool dbo_pool (string pool.scheduler="TS")
 - create pool db1_pool (string pool.scheduler="TS")
 - create pool db2_pool (string pool.scheduler="TS")
 - create pool db3_pool (string pool.scheduler="TS")
 - create pool db4_pool (string pool.scheduler="TS")
 - associate pool pool_default (pset pset_default)
 - associate pool dbo_pool (pset dbo_pset)
 - associate pool db1_pool (pset db1_pset)
 - associate pool db2_pool (pset db2_pset)
 - associate pool db3_pool (pset db3_pset)
 - associate pool db4_pool (pset db4_pset)

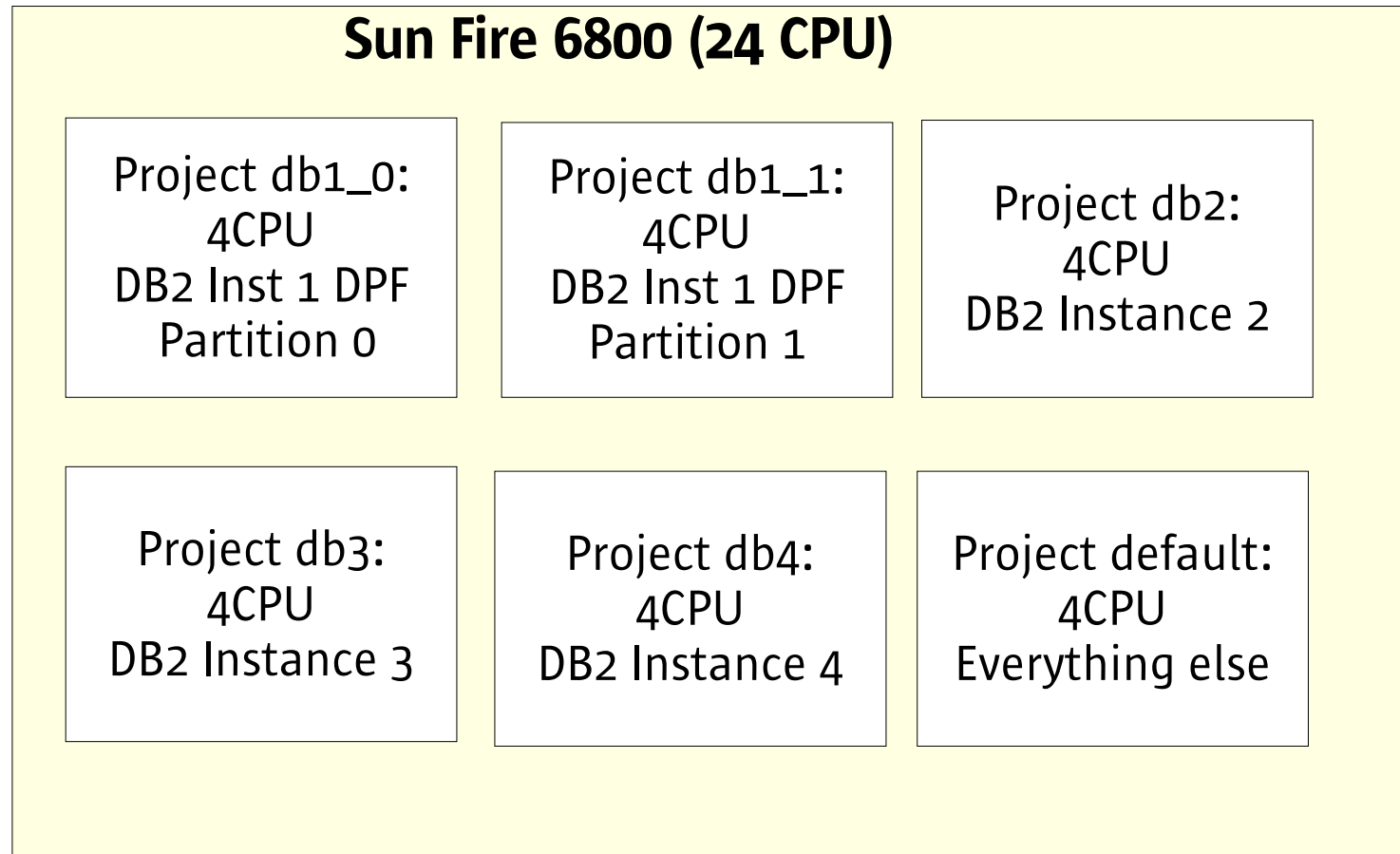
Setting Up Resource Pools

- Creating pool configuration
 - # /usr/sbin/poolcfg -f pool.host
- Activating pool configuration
 - # /usr/sbin/pooladm -c
- If you need to deactivate the configuration
 - # /usr/sbin/pooladm -x
- Use the following to view your active pool configuration
 - \$ /usr/sbin/pooladm

Setting Up DB2 to Use Projects

- Need DB2 V8.1 fixpack 4
- You can set the project information in `db2nodes.cfg` (Solaris 9 only)
- The modified `db2nodes.cfg` format is
 - `nodenum hostname logical port netname projectName`
- Example of `db2nodes.cfg` for `db2inst1`
 - `0 myhost 0 myhost db1_0`
 - `1 myhost 1 myhost db1_1`
- Now when you start `db2inst1`, node 0 starts in the `db1_0` project while node 1 starts in the `db1_1` project

System View

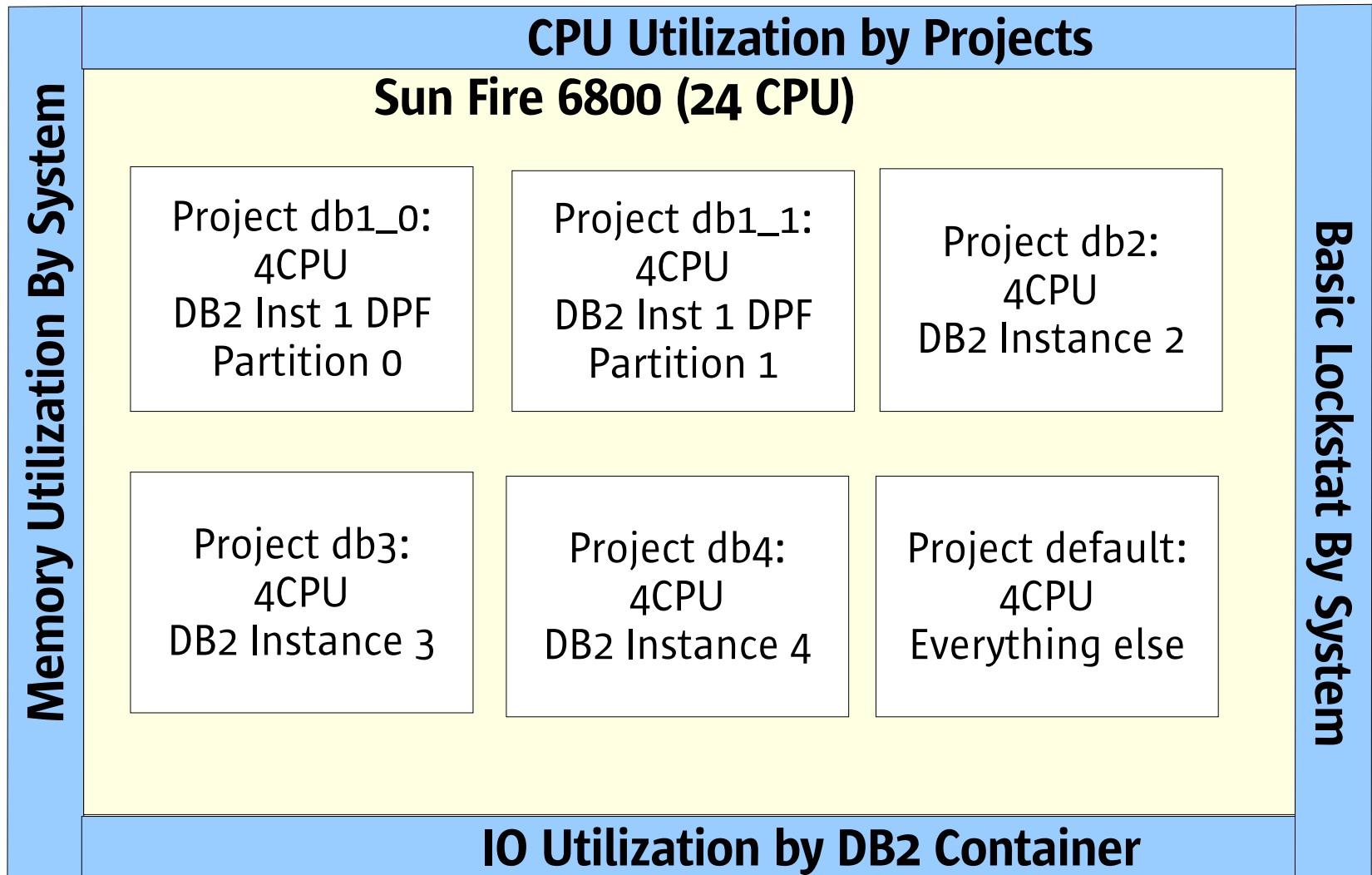


Note: This is just an example and not an actual recommendation for Sun Fire 6800. Actual distribution should consider all applications running on the system.

Monitor Resources

- Use following tools to narrow your search
 - `prstat -j` (or drill down using `prstat -j -j db1`)
 - `vmstat 5` (check total memory consumption)
 - `iostat -xnez 5` (narrow down to the containers)
 - `lockstat sleep 5` (as root to get kernel locking statistics)

Basic Monitoring



What Does prstat Tell You?

- Using `prstat -J` and `prstat -J -j projname` you can get the relative CPU usage by each project
- It will show you relative memory consumption too but it will add up to more than 100% (shared memory)
- This will give a global view w.r.t. projects which in this case will represent DB2 nodes (or instances) and the default system activity

What Does vmstat Tell You?

- You need to monitor the free swap and free memory space primarily
- If you cannot explain the used memory then try using db2mem to analyze memory usage
- If excessive paging is occurring in system

(HINT: If you are using INTRA_PARALLEL = NO then try setting SHEAPTHRES_SHR to a small value like 250 since SORT will be done in client private memory anyway)

What Does iostat Tell You?

- Containers having I/O problems
- Imbalance in Containers IOPs
- Containers can be traced back to the tablespaces and can give an indication of which database and the tables related to the tablespace could be having problems

(HINT: If you are use Solaris Volume Manager's soft partitions to map to each DB2 containers then you can use iostat to monitor each containers separately and trace it back to DB2 Tablespace)

What Does lockstat Tell You?

- Any reader/writer blocks going on (very expensive — need to be resolved)
- Any excessive adaptive mutex spins going on (If it is DB2 related, try to control using DB2SPINLATCHES registry value)

What to Do Next?

- Take a baseline number
- Compare your current readings and baseline number
- If current readings are better than your baseline number update your baseline number
- Iteratively try one change at a time

Things to Try on DB2

- Make sure db2set contains
 - DB2_STRIPED_CONTAINERS=ON
 - DB2_PARALLEL_IO="*"
- Use DB2 monitor switches if you doubt any DB2 record locking issues
- Refer the DB2 manual to tweak the right tuning parameters according to the output of “snapshots”

Things to Try on Solaris

- If you are using DMS on UFS try the “forcedirectio” mount option (use lockstats to verify the difference too)

You Are Here

- General Considerations
- Memory Usage and Bottlenecks
- Disks and I/O Considerations and Bottlenecks
- Tuning DB2 V8.1 in Solaris 9
- Other Stuff...

How Can We Influence DB2 Optimizer Without Hints?

- Using DB2 parameters
- Available memory
 - To sort
 - To read and write data
 - To maintain consistency (lock)
- Expected number of concurrent queries
- Hardware characteristics
 - Processor speed
 - Communication speed
 - Disk speed
- Eventually, desired parallelism
 - Parallel limits
 - Parallel defaults
 - Parallel deactivation
- Sophistication for optimization
 - Optimization level

How Can We Influence DB2 Optimizer Without Hints?

- Database layout
 - Tablespace design
 - Table constraints
 - Table alteration
 - "Volatile" (favor index scans)
 - No row locking
 - Table partitioning
- Enabling/disabling special features
 - Sort algorithm
 - Correlation detection
 - Search algorithm for varchar
 - ...
- Collecting statistics
 - RUNSTATS
- There are no hints in DB2!

Other Good Stuff...

- Setting up projects and resource pools in Solaris 9
 - <http://docs.sun.com/db/doc/817-0204>
 - <http://docs.sun.com/db/doc/817-0204/6mg168bsn?a=view>
- www.redbooks.ibm.com
 - DB2 UDB V8.1 Performance Tuning Guide
 - DB2 UDB V8.1 Porting Guide
 - Migrating to IBM Replication Solution (from Sybase)
 - Migrating to DB2 UDB V8.1 in a Visual Warehouse Environment
 - ESS Solutions for Open Systems Storage: Compaq AlphaServer, HP, and SUN
- DB2 on Sun Install Guidelines whitepaper — contact Ranger Team
 - DB2sun@us.ibm.com

Other Good Stuff...

- Papers
(<http://www7b.software.ibm.com/dmdd/library/techarticle/>)
 - Clustering with Sun Cluster:
www.ibm.com/software/data/pubs/papers/suncluster/suncluster.pdf
 - Clustering with Veritas:
<http://www.ibm.com/support/manager.wss?rs=71&rt=0&org=SW&doc=1045033>
 - Quick Reference for Tuning:
<http://www7b.software.ibm.com/dmdd/library/techarticle/0205parlapalli/0205parlapalli.html>
- Free download of 2-day “FastPath to UDB for Experienced DBA’s” Course
 - <http://www-3.ibm.com/software/data/education/ct28v8.html>



SunNetwork
2003 Conference and Pavilion

TUNING DB2 IN A SOLARIS ENVIRONMENT

Tom Bauch
bauch@us.ibm.com

Jignesh Shah
j.k.shah@sun.com

<http://sun.com/db2>

