GemStone/S Update

Norman R. Green

Director, R & D, VMware Inc.

ESUG Conference 2012

Gent, Belgium

Agenda

- GemStone/S 32
- GemStone/S 64 2.4.x
- GemStone/S 64 3.0
- GemStone/S 64 3.1

GemStone/S 32 Bit

- Yes, it's still in use, and in production!
- Last Major Release: 6.6.2 in Feb 2012
- 6.6.x Summary
  - Rudimentary multi-threaded garbage collection
    - Use native threads to perform I/O
  - Symbol Garbage Collection
    - Methods to locate and remove unreferenced symbols.
    - Non-canonical symbol detection
  - Methods to override commit-at-login behavior
    - lastLoginTime is committed at login time if UserProfile security is enabled.
    - Commit-at-login can now be disabled on a per-UserProfile basis.

GemStone/S 32 Bit

- 6.6.x Summary
  - Backup and Restore to/from NFS
    - Full backups may now be made to an NFS file system.
    - Full backups may be restored from an NFS file system.
    - Tranlogs may be restored from an NFS file system.
  - POWER7 Certification (AIX only)
    - Add additional memory barriers required by POWER7 Processor

GemStone/S 32 Bit

- End of Life
  - EOL for GemStone/S 32 bit is planned for Oct-2012.
  - What does that mean exactly?
    - No more major releases.
    - No new product sales.
    - Support for existing customers will continue for 3 more years (2015).
    - Maintenance renewals beyond 2015 will not be accepted.

GemStone/S 64 2.4.x

- Current Releases
  - 2.4.4.7
    - Critical bug fixes
    - A few minor new features
  - 2.4.5
    - Critical and non-critical bug fixes
    - Several minor new features
  - 2.4.5.1
    - Minor bug fixes
    - POWER7 certification
- Further Details
  - Release Notes
  - http://community.gemstone.com/display/GSS64/Documentation


COSCO
COSCO LONG BEACH

GemStone/64 3.0

- 3.0 shipped on June 15, 2011.
- 3.0 was a major step forward in features and performance.
- Adoption has been (predictably) slow.

- Native Code Virtual Machine
  - Converts byte codes to native instructions
  - JIT design – conversion occurs at first method invocation.
  - 25% to 100% speed improvement

- Foreign Function Interface (FFI)
  - Load shared libraries.
  - Call C functions without writing C code.
  - All coding can be done in Smalltalk
  - Martin presented FFI at STIC 2011

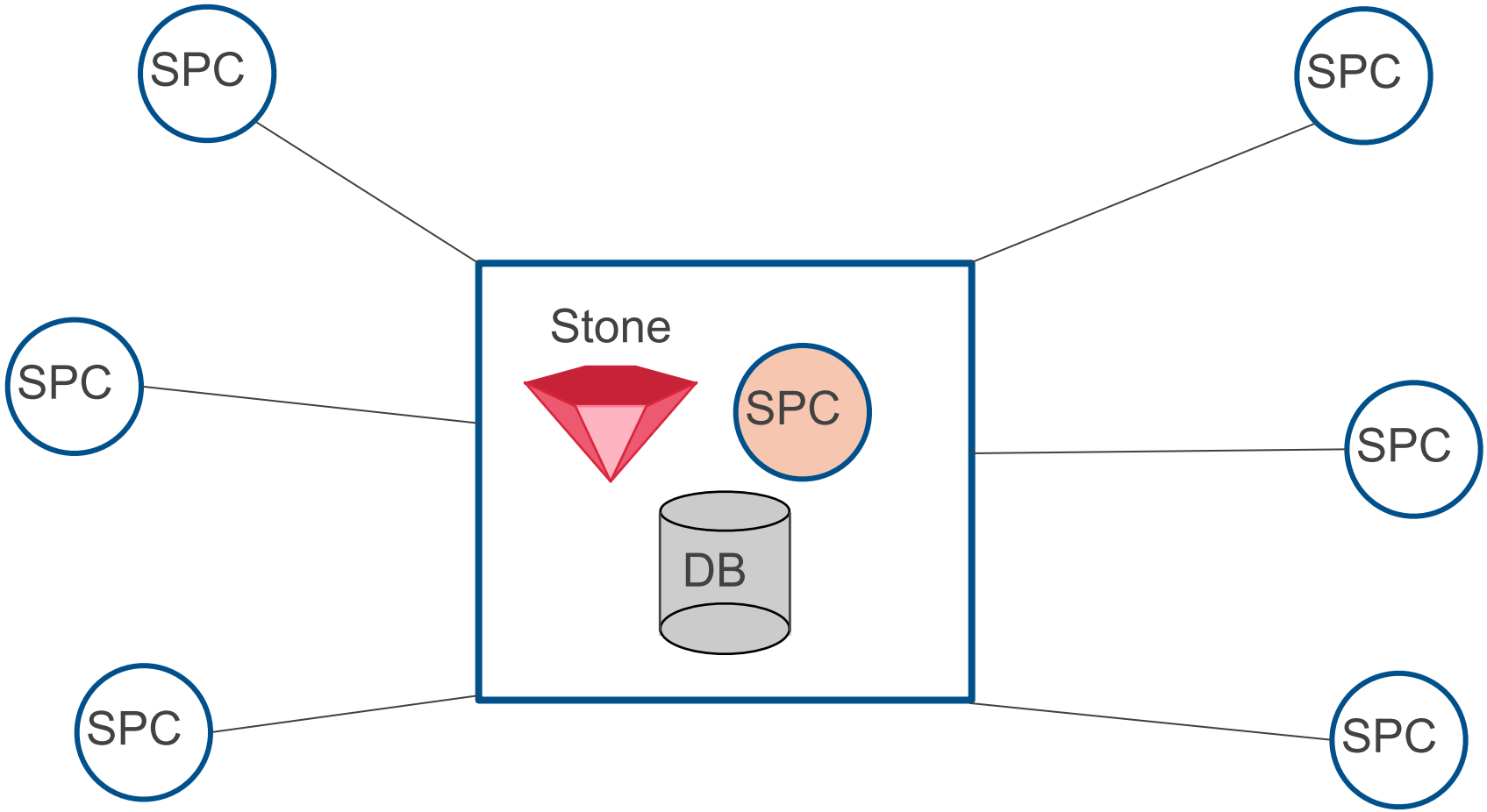
GemStone/S 64 3.1.x

- Current Releases
  - 3.1 - July 6, 2012
  - 3.1.0.1 - August 28, 2012
- Next Release
  - 3.1.1 - Q1, 2013
    - Bug fixes

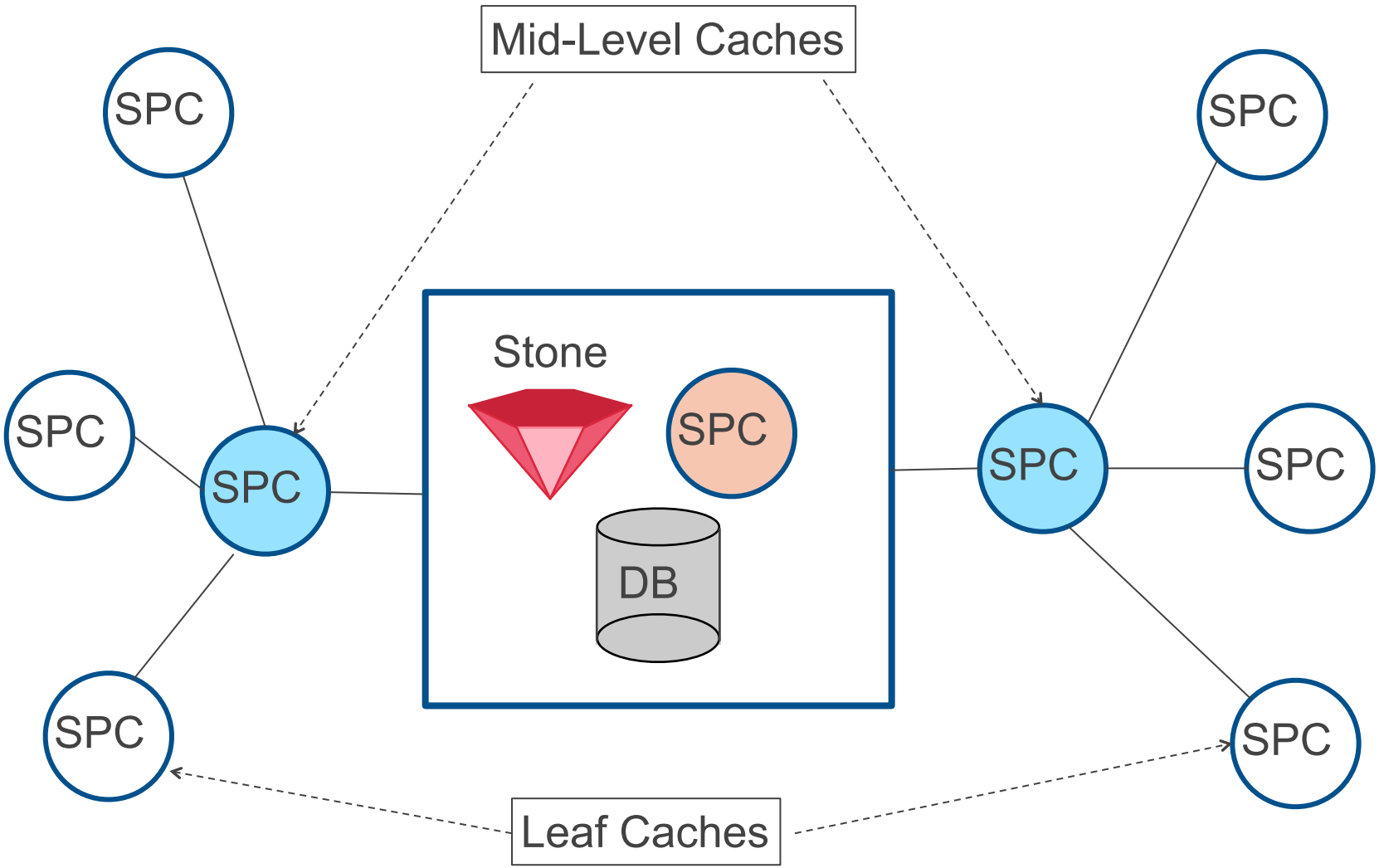
New Features

- Mid-Level Shared Caches
  - Caches that sit between the stone cache and a remote SPC
  - Benefits
    - Better caching.
      - 3 caches to look for pages rather than 2
    - Reduced Network I/O on Stone Server
      - 500 remote caches can red-line a 10 GB/s network!

Classic Remote SPC “Star” Configuration



Mid-Level Cache Configuration



New Features

- Mid-Level Shared Caches
  - Creating a Mid-Level Cache

1. Start the netldi on the remote (leaf cache) host, mid-level host, and stone server
2. From the leaf-cache host, do a normal remote login
3. Run this code to create the mid-cache:

```
System
    midLevelCacheConnect: midLevelHostName
    cacheSizeKB: aSize
    maxSessions: nSess
```

New Features

- Mid-Level Shared Caches
  - Connecting to an Existing Mid-Level Cache

1. From the leaf-cache host, do a normal remote login
2. Run this code to connect to the mid-cache:

```
System midLevelCacheConnect: hostName
```

New Features

- External Password Validation
  - Passwords may now be stored and managed outside of GemStone.
  - 2 New Options
    - LDAP = Lightweight (ha!) Directory Access Protocol
    - UNIX password

New Features

- External Password Validation
  - User ID Aliases are Supported
    - GemStone user ID may or may not match external user ID.
    - Example:
      - GemStone UID: ‘normg’
      - LDAP UID: ‘norm.green’

New Features

- External Password Validation
  - LDAP – Uses OpenLDAP and OpenSSL libraries (shipped with GemStone).
  - UNIX – Uses getpwnam() / getspnam() UNIX calls.

New Features

- External Password Validation
  - Enabling LDAP authentication for a UserProfile:

```
| u |
u := AllUsers userWithId: 'normg'.
u enableLDAPAuthenticationWithAlias: 'norm.green'
  baseDN: 'uid=%s,ou=LdapTesting,dc=gemstone,dc=com'
  filterDN: nil .
System commitTransaction .
```

New Features

- Multi-threaded Operations
  - The following operations are now multi-threaded:
    - markForCollection
    - objectAudit
    - findDisconnectedObjects
    - countInstances
    - listInstances
    - listReferences
    - GsObjectInventory
    - findObjectsLargerThan
    - Epoch Garbage Collection
    - Write Set Union Sweep (part of GC finalization)
  - Native (OS) threads are used.
  - Each thread is a unique GemStone session
  - Each thread takes a shared page cache slot.

New Features

- Multi-threaded Operations
  - Aggressiveness Is Controlled By 2 Parameters:
    - Number of threads
    - Percent CPU Active
      - Means total CPU load on the server across all cores.
      - Threads will start sleeping if the threshold is exceeded.
  - Both Parameters Can Be Adjusted On The Fly
    - `SystemRepository setMultiThreadedScanThreads: numThreads maxCpu: aPercent`

New Features

- Multi-threaded Operations
  - The default methods are NOT aggressive
  - If you want speed, use the fast* methods
    - **`markForCollection`** – uses 2 threads
    - **`fastMarkForCollection`** – uses up to 16 threads

New Features

- ProfMonitor Improvements
  - New implementation supports sample intervals down to 1 microsecond (1000 ns).
  - New Class Protocol
    - `ProfMonitor monitorBlock: aBlock downTo: hits intervalNs: ns`
    - `ProfMonitor monitorBlock: aBlock intervalNs: ns`

New Features

- Built-In Monticello Support
  - Monticello Support is now available “out of the box”
  - Protocols:
    - `GSPackageLibrary loadMczFile: aFile fromRepositoryPath: aPath`
    - `GSPackageLibrary loadLastVersionOf: packageName fromRepositoryPath: aPath`

vmware

Smalltalk Changes

- Array Literal Syntax
  - Run-time Constructor
    - `2.x:  #[ 1, 2, 3 ]`
    - `3.0: { 1 . 2 . 3 }`
  - Compile-time Constructor (unchanged)
    - `#( 1, 2, 3)`
- In 3.0, the old 2.x Array syntax can be enabled by running this code after login:
  - `System gemConfigurationAt:#GemConvertArrayBuilder put: true`
- ByteArray Literal Syntax (new)
  - `#[ 0 1 254 255 ]`

vmware

Smalltalk Changes

- Exception Handling
  - In 3.0, ANSI exceptions are fully supported.
  - ANSI exceptions are class-based.
  - Old GemStone Exception class is deprecated, but still works in some cases.
  - GemStone error numbers are deprecated.

Smalltalk Changes

- New Special Selectors
  - The following selectors are inlined by the compiler and may not be overloaded:
    - `ifNil:`
    - `ifNotNil:`
    - `ifNotNil:ifNil:`
    - `ifNil:ifNotNil:`
    - `_isSmallInteger`
    - `_isInteger`
    - `_isNumber`
    - `_isFloat`
    - `_isSymbol`
    - `_isExceptionClass`
    - `_isExecBlock`
    - `_isArray`

Smalltalk Changes

- Segment is renamed GsObjectSecurityPolicy
  - We've wanted to change that for a long time.

Smalltalk Changes

- Dynamic Instances Variables
  - Add instance variables on the fly!
  - No instance migration required.
  - Replaces object tags
  - Methods:
    - `Object dynamicInstVarAt: aSymbol`
    - `Object dynamicInstVarAt: aSymbol put: anObject`
    - `Object removeDynamicInstVar: aSymbol`

Smalltalk Changes

- LargeInteger Changes
  - New class: LargeInteger
    - Replaces both LargePositiveInteger and LargeNegativeInteger
    - Existing Instances:
      - LargePositiveInteger -> ObsoleteLargePositiveInteger
      - LargeNegativeInteger -> ObsoleteLargeNegativeInteger
    - Instances are automatically converted when loaded into object memory.
    - To convert manually, send:
      - `anObsoleteInt convert`

Smalltalk Changes

- ScaledDecimal Changes
  - New format:
    - mantissa (SmallInteger or LargeInteger)
    - scale (SmallInteger) – power of 10
  - New Literal Notation (ANSI compliant)
    - `1.23s2`
      - Mantissa: 123
      - Scale: 2

Smalltalk Changes

- ScaledDecimal Changes
  - GS/64 v2.x ScaledDecimal is now FixedPoint in 3.0
  - Instances are automatically converted
  - New Literal Syntax:
    - `1.23p2`
      - Numerator: 123
      - Denominator: 100
      - Scale: 2

Smalltalk Changes

- New Class/Metaclass Hierarchy

2.x

- Object
  - Behavior
    - Metaclass
    - Class

3.0

- Object
  - Behavior
    - ObsoleteMetaclass
    - Module
      - Metaclass3
        - Class

- After upgrade, existing Metaclasses become ObsoleteMetaclass
- Newly created classes will have a class of Metaclass3
- Applications which are sensitive to the superclass of Class or Metaclass will need to be adjusted.

Performance Improvements

- Aborting A Transaction
  - Round trips to stone reduced from 2 to 1

Performance Improvements

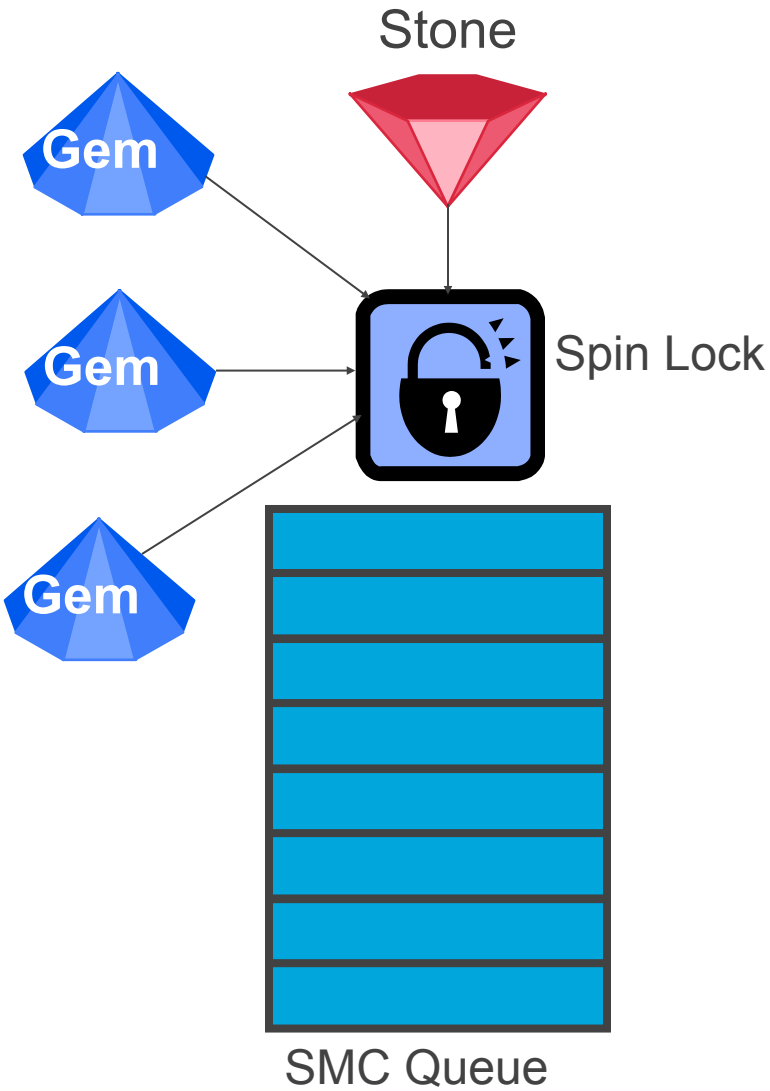
- Old SMC Design

Gem Procedure

- Get SMC queue lock
- Add self to SMC queue
- Release SMC queue lock
- Wait on semaphore for stone's response.

Stone Procedure

1. Get SMC queue lock.
2. Remove all requests from SMC queue.
3. Release SMC queue lock
4. Process Requests



vmware

Performance Improvements

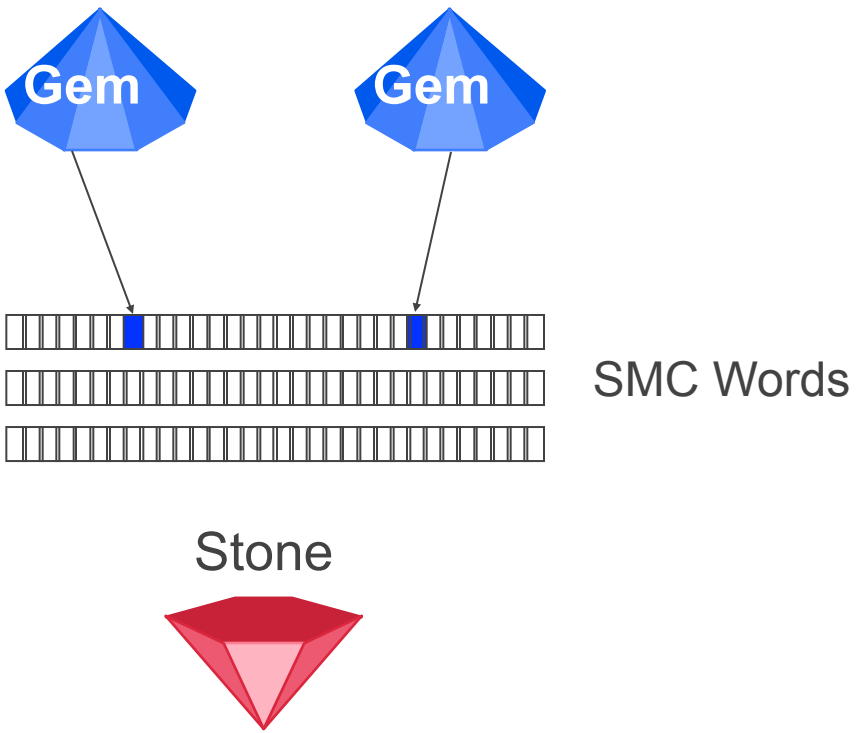
- New SMC Design – No Lock or Queue

Gem Procedure

- Set my bit (atomic OR)
- Wait on semaphore for stone's response.

Stone Procedure

1. For each 64-bit session word:
   - Read and clear all bits (atomic AND).
2. Process Requests



vmware

Performance Improvements

- Session Priority
  - Elimination of SMC queue means sequence of sessions requesting service from stone is not preserved.
  - Session Priority added to compensate.
  - Session Priority Values:
    - 0 – lowest (Reclaim and Admin GC gems)
    - 1 – low
    - 2 – medium (default)
    - 3 – high
    - 4 – highest
  - Session holding (or about to receive) the commit token always has highest priority.

Performance Improvements

- Session Priority Protocol
  - `System setSessionPriority: anInt forSessionId: aSessionId`
  - `System priorityForSessionId: aSessionId`

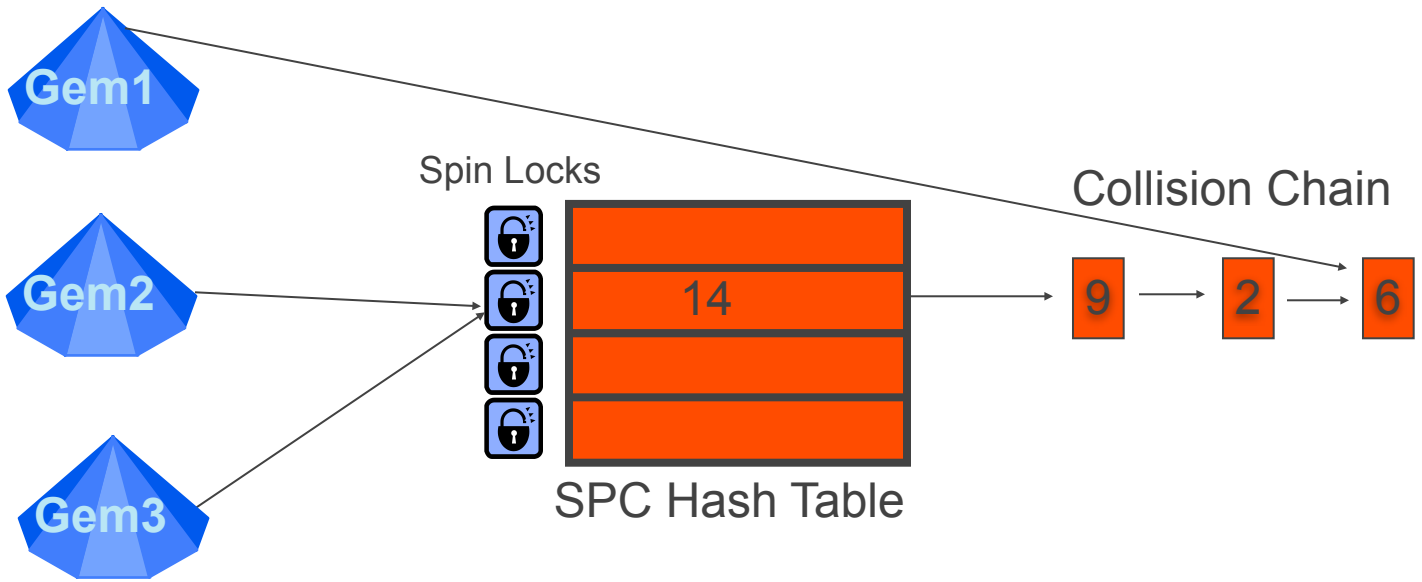
Performance Improvements

- Shared Cache Hash Table Read Locks
- SPC Hash Table is a dictionary in shared memory
  - `pageId -> cache offset (address)`

  - **2.x Design**
    - **All hash table row accesses were serialized by spin lock.**
    - **Result: All lookups are serialized.**
  - **3.0 Design**
    - **Each hash table row now has a reference count and a write lock.**
    - **Result: Lookups are done in parallel.**

Performance Improvements

- Serialized Hash Table Lookups In 2.x

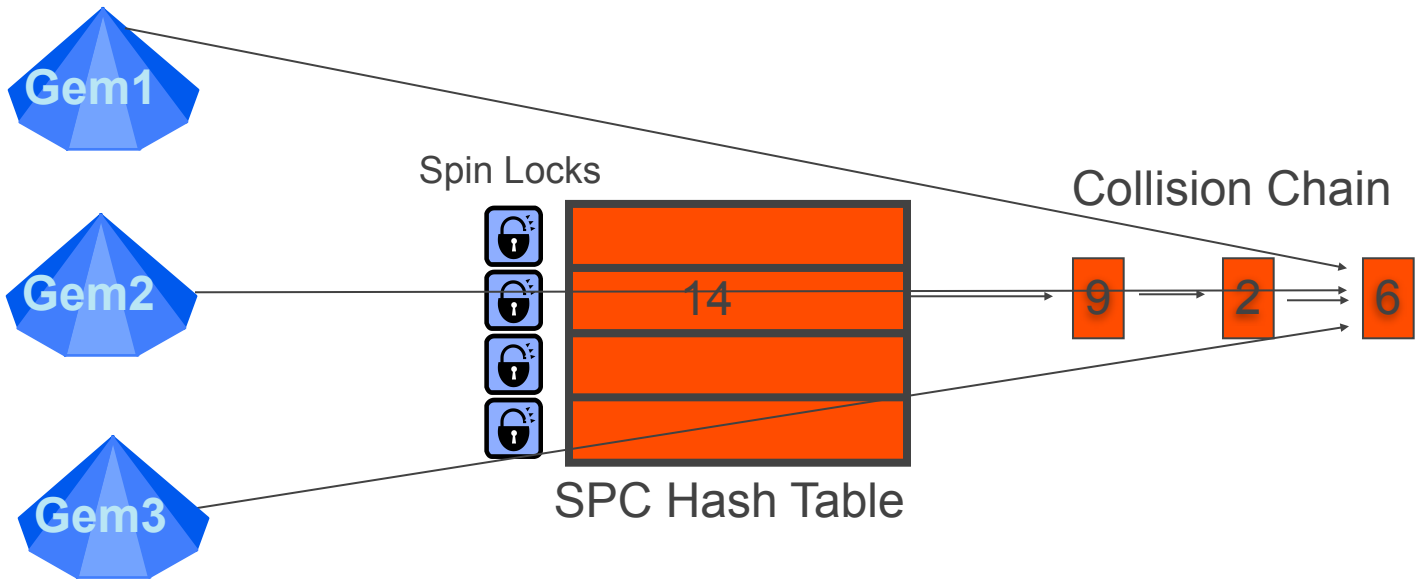


•3 Gems want to lookup page 6.
•Each gem must hold the lock to perform the lookup.

If the lookup time is *n*, then:
- •Gem1: *n*
- •Gem2: *2n*
- •Gem3: *3n*

Performance Improvements

- Parallelized Hash Table Lookups In 3.0



•3 Gems want to lookup page 6.
•All 3 lookups can be done simultaneously!

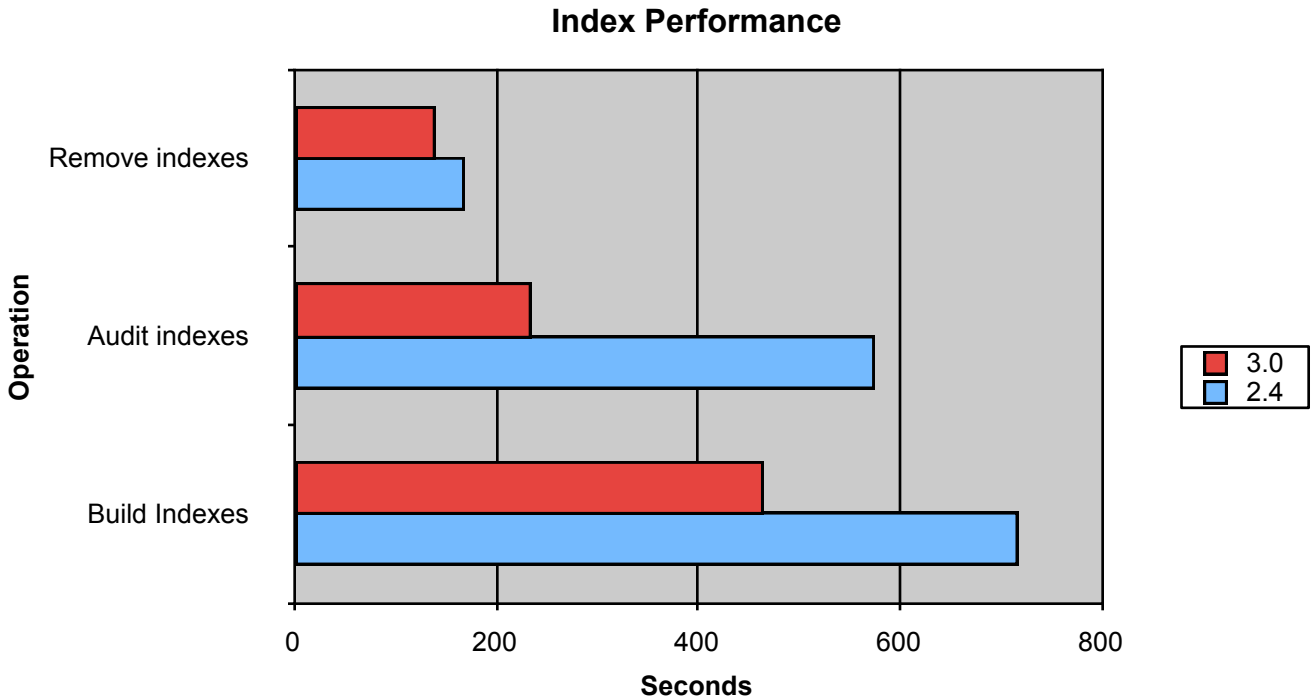
If the lookup time is *n*, then:
- •Gem1: *n*
- •Gem2: *n*
- •Gem3: *n*

vmware

Performance Improvements

- New Tranlog AIO System
  - POSIX AIO calls (aio_write()) removed.
    - Too many OS bugs
  - Replaced With Our Own Code
  - Based on Native OS Threads
  - New Config Parameter:
    - **STN_NUM_AIO_WRITE_THREADS** – Number of native threads started to perform tranlog writes.

Index Performance Improvement



- Solaris 10 SPARC
- Tranlogs on solid state disks
- Extents on raw partitions.
- 2M element collection
- 5 Equality Indexes

GemStone/S 64 3.1

- **Hot-standby database support**
  - Automatically synchronize 1 or more hot standby databases across a LAN or WAN.
  - Real-time tranlog replay
  - Failover support
  - Automatic reconnect support
- **Support for UTF & Locale-specific collation using ICU**
  - Interface to the International Components for Unicode (ICU) libraries.
  - ICU is a widely used open source set of libraries providing Unicode and globalization support for software applications.
- **Secure RPC logins using SSL**
  - All RPC logins now use Secure Socket Layer (SSL) and Secure Remote Password (SRP) to establish the initial connection between the GCI client and gem and to authenticate passwords.
  - Passwords are now stored in GemStone in the encrypted form used by SRP.

GemStone/S 64 3.1

- **Backup and restore reimplemented as multi-threaded**
  - For multi-file backups, the protocol that created or restored backup files in a sequence of commands has been removed
  - All backup files are created or restored simultaneously
  - Multi-file backups are now created or restored by a single command rather than repeated calls to continue* methods
  - Written in parallel, using separate threads to write to each file after initial material is written to the first file.
- **Symbol garbage collection**
  - If enabled, this occurs automatically in the background and requires no management.
- **Garbage Collection Enhancements**
  - Ability to defer reclaim under low free space conditions
  - Further optimizations to MFC
- **The FFI has a number of changes and improvements**

Nested Transactions

- Create up to 16 sub-transactions which may be independently aborted or committed.
- Protocol:
  - System beginNestedTransaction
    - Begin a new nested transaction
  - System abortNestedTransaction
    - Roll back changes made in this nested transaction only.
  - System commitNestedTransaction
    - Commits the modifications made in the current nested transaction to the next outer-level transaction.
    - Changes are NOT seen by other sessions until the outer-most transaction is committed.
    - Write-write conflicts with other session are not detected here.

IPv6 Support

- 3 Address protocols now supported:
  - IPv4
  - IPv6
  - IPv4-mapped IPv6
- New Config File Options:
  - STN_LISTENING_ADDRESSES – A list of 0 to 10 addresses upon which stone should listen for login connections .
- For More Info:
  - RFC 2373
  - RFC 4291
  - RFC 4038

GemStone/S 64 3.2

- Target Date
  - 3.2 - Q3, 2013
- New Features
  - Multi-threaded page reclaim sessions.
  - Additional multi-threaded garbage collection options.
  - Additional Unicode character features
  - Thread-safe GCI (Customer C Interface)
  - Optimize GemStone/S to run on ESXi.
  - Add protection for DoS attacks.
  - Load Balancer – balance loads across multiple nodes.

Questions ?

VMware Inc.
15220 NW Greenbrier Pkwy.
Suite 150
Beaverton, Oregon, 97006

www.vmware.com

Norman R. Green
Director, R&D
normg@vmware.com

(503) 533-3710 direct
(503) 804-2041 mobile

Ruby that Scales

GemStone/S HIstory

1982	32-bit	64-bit
.		
.		
.		
1986	1.0	
.		
.	1.5	
.		
1990	2.0	
.		
.	3.0	
.		
.	4.0	
.		
.	5.0	
.		
.	5.1	
.		
2000		
.		
.	6.0	
.	6.1	
.		
.		1.0
.		2.0
.	6.2	2.2
.	6.3	2.3
.	6.5	2.4
2010		
.	6.6	3.0
.		3.1

First GemStone customer, May 1986

The future is here, it's just not evenly distributed yet.
– William Gibson