New Features in Sun Grid Engine 6.2

- **Roland Dittel, Lubomir Petrik**
- Software Engineers
- Sun Microsystems

Advance Reservation

- "An advance reservation is a possibly limited or restricted delegation of a particular resource capability over a defined time interval, obtained by the requester from the resource owner through a negotiation process."
  - Grid Resource Allocation Agreement Protocol WG (GRAAP-WG)

Advance Reservation

- Enables users to schedule compute time
  - Schedule around people, places and resources
- Just like calling a restaurant:
  "I'd like a table for **4 person** at **6:00PM** on **Tuesday** for **2 hours** in **room Helsinki** and we'll need **silver cuterly**"

  ↓

  "I'd like **4 nodes** on at **6:00PM** on **Tuesday** for **2 hours** on **lx24-amd64** and I'll need the **silver library**"
- Entitled users can create and delete their own reservations
  - The scheduler makes sure everything fits

Advance Reservation Details

- Jobs can be submitted to a reservation
  - Scheduled within reservation boundaries
  - Terminated when reservation ends
- Reservations can be shared
  - Multiple users and/or groups
  - Declared when requesting the reservation
- Backfill before reservation
  - Other reservations
  - Jobs with run time limits
- New tools for reservation administration
  - qrsub, qrstat, qrdel

Advance Reservation Scheduling

- All requested resources must be available
- All allowed users must have access
- Unbounded jobs are kept apart
  - Unbounded jobs have no respectively infinite run time limit
  - Cannot be scheduled before an advance reservation
  - An advance reservation cannot be scheduled after
- Free to use resources as desired
  - May be shared by multiple users
  - May be used for multiple jobs
  - May go unused
- Resource Quotas are not considered

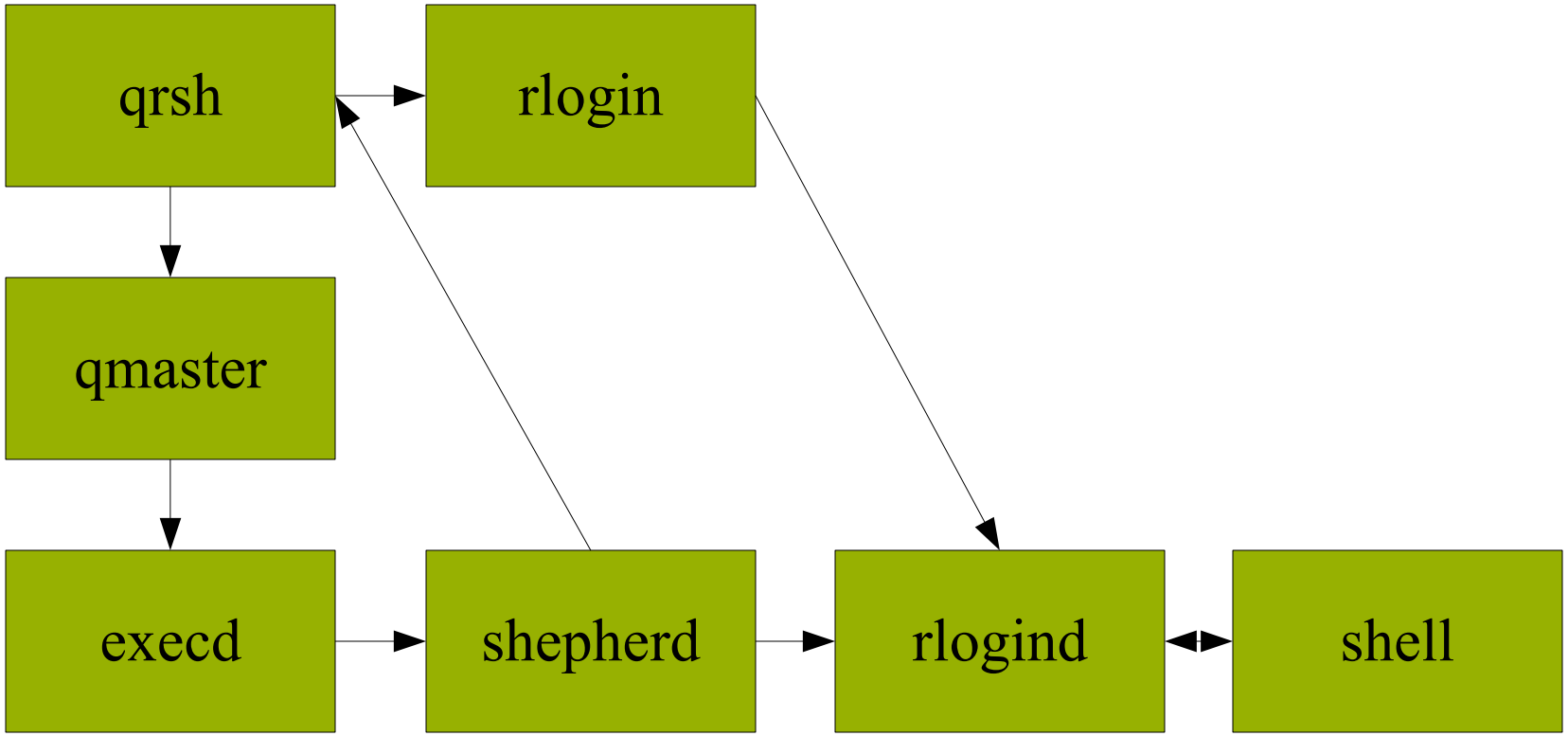
Advance Reservation Use Cases

- User Reservations
  - Grant access to exclusive resources
  - Can be used by a group of users
  - Resource availability is guaranteed
- System Reservations
  - Reserve nodes for system maintenance
  - Scheduler ensures nodes are free
  - Can be used by maintenance jobs

Advance Reservation Examples

- Reserve now a slot in queue all.q on host1 or host2
  - % qsub -q all.q -l 'host=host1|host2' -d 1:0:0
    Your advance reservation 1 has been granted
    % qsub -ar 1 work.sh
    Your job 1 (“work.sh”) has been submitted
- Reserve 16 slots on sol-amd64 nodes at a specific time
  - % qrsub -pe mpi 16 -l arch=sol-amd64 -a 06061200 -d 3600
    Your advance reservation 2 has been granted
    % qsub -ar 1 -pe mpi 4 blast.sh
    Your job 2 (“blast.sh”) has been submitted

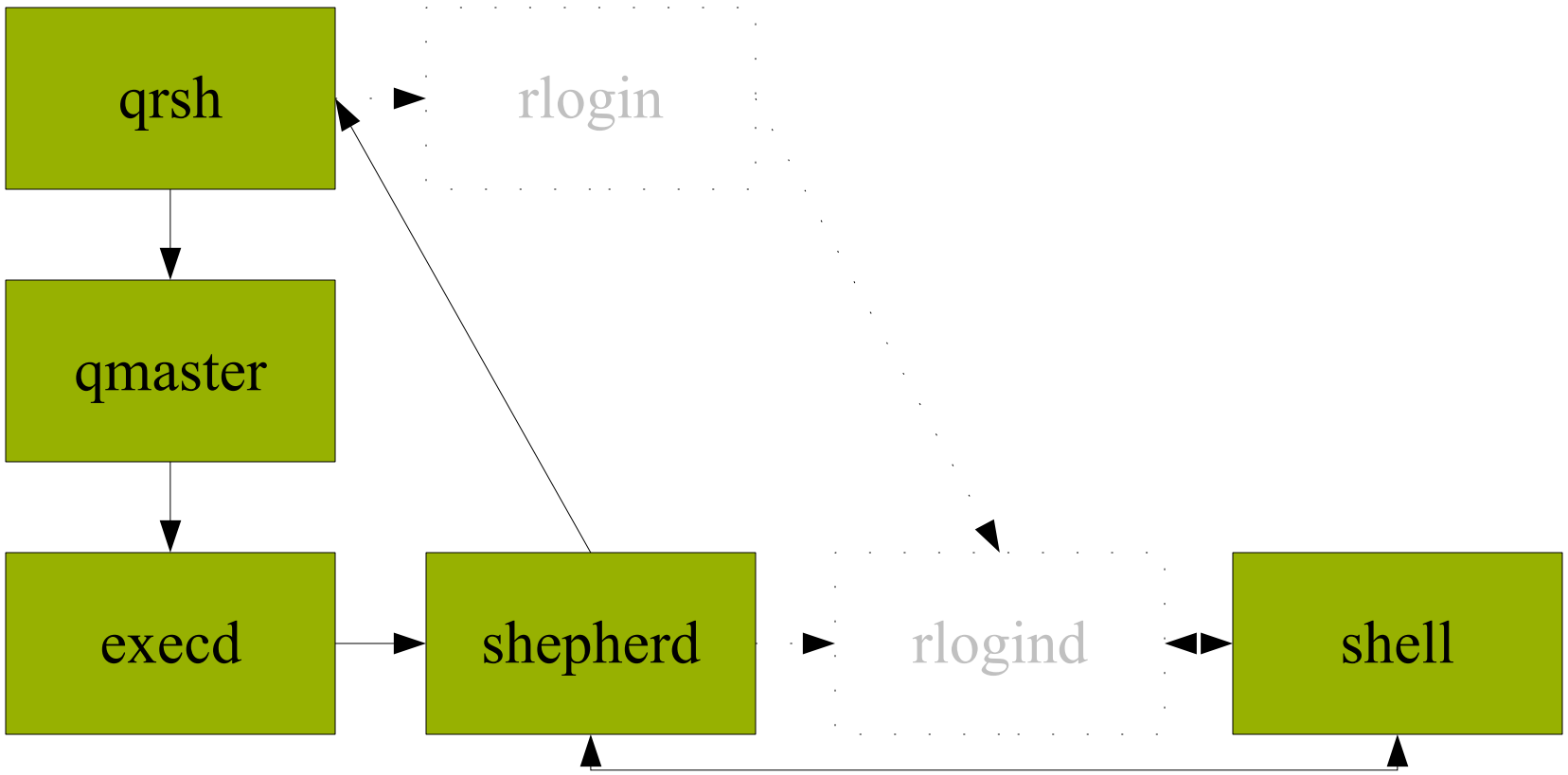
Old Interactive Job Support



Old Interactive Job Support

- Implementation based on Codine 5.1
  - Slow and complicated connection establishment
  - No pty for shell
  - Complicated signal forwarding
  - Based on external commands
    - rlogin, telnet
      - Limited number of reserved ports
      - No secure connection
    - ssh
      - No monitoring and accounting
  - Lots of other issue

New Interactive Job Support



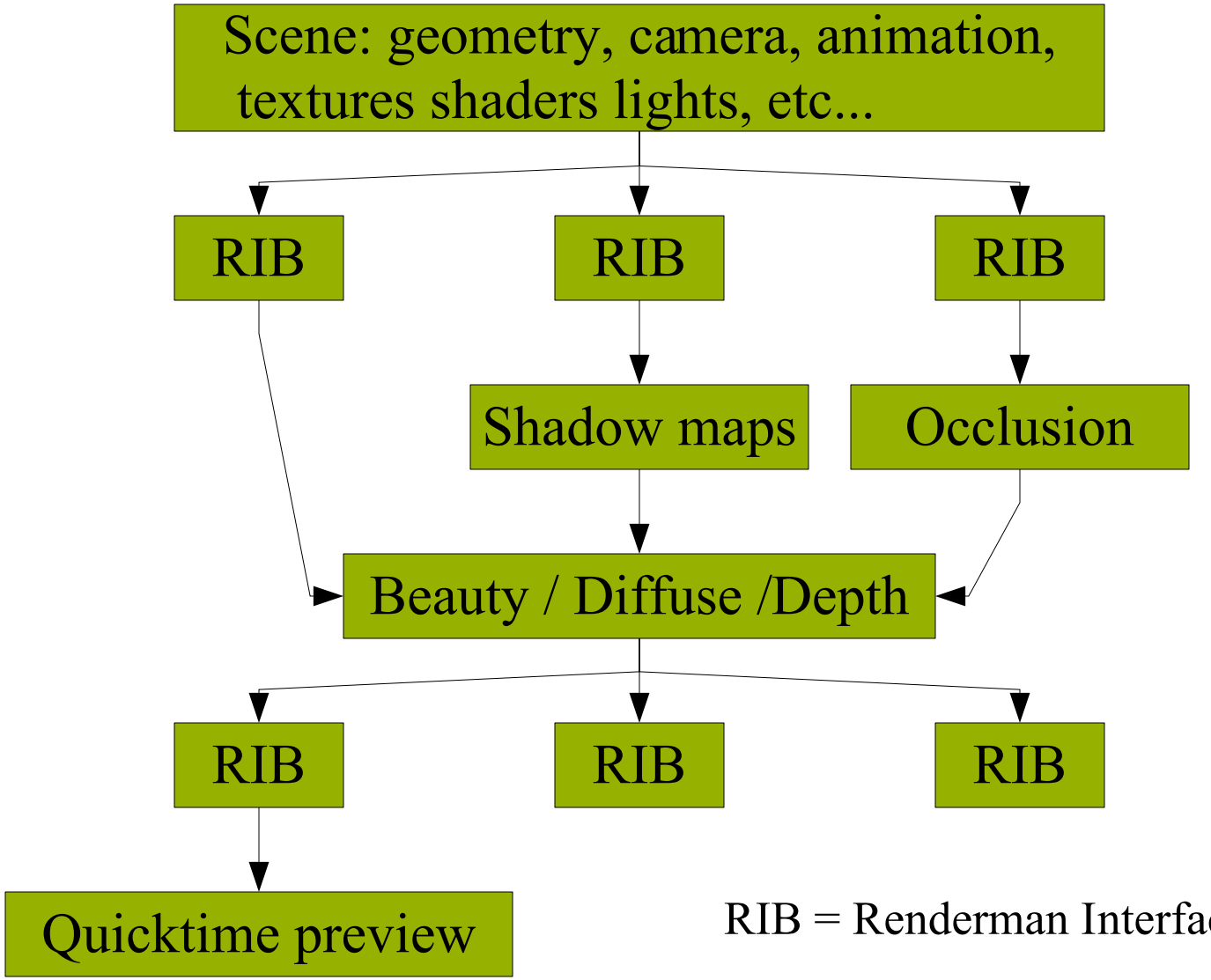
New Interactive Job Support

- Complete rewrite from scratch
  - No longer based on external commands
  - Client communicates directly with shepherd
    - Faster and simpler connection establishment
    - Port number limits don't apply
    - Full monitoring and accounting data
    - Secured connection in CSP mode
  - Signals are correctly forwarded
  - Shell get's a pty

Array Job Dependency

- Contribution from Rising Sun Pictures
  - Australian visual effects company
  - Focus entirely on producing visual effects for feature films
  - Movies rendered using Grid Engine
    - Harry Potter - The Goblet of Fire and The Order of the Phoenix
    - Superman Returns
    - Elephant Tales
    - Charlotte's web
    - ...
  - http://www.rsp.com.au

Rising Sun Pictures Workload



RIB = Renderman Interface Binary

Array Job Dependency

- Array jobs are ideal for render farms
  - Compute tasks (frames) are independent
  - Very low submission time
  - Very low qmaster footprint
- But dependency description was to inflexible
  - Submit an array job A
  - Submit an array job B which is dependent on job A

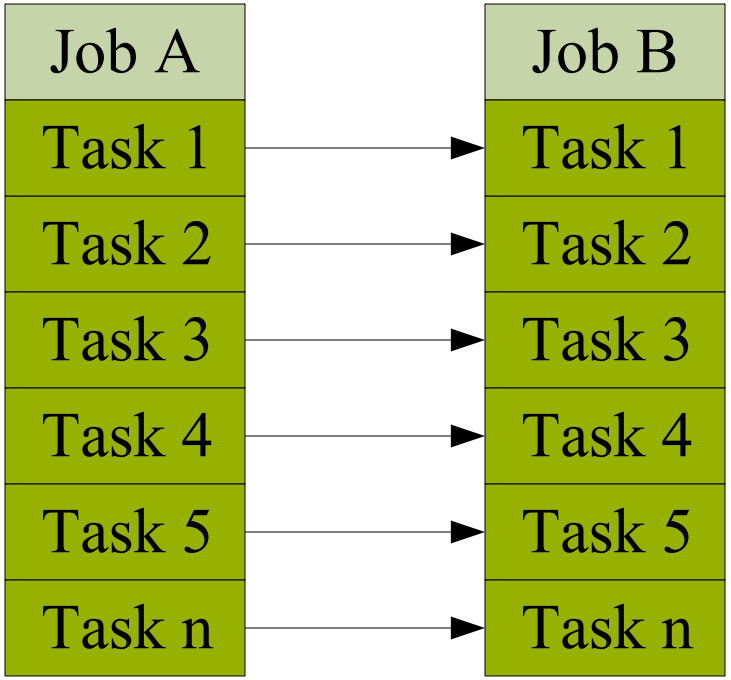
Job B won't start until the whole array job A has finished.

Array Job Dependency Workarounds

- Ignore the problem
  - Low throughput
- Render one dependency tree in a single job
  - No visibility of the progress
  - Doesn't work if sub tasks requires different nodes
  - Reduced throughput when using chunking
- Break array job into set of array jobs
  - Complex
  - High submission times
  - Multiple jobs relating to a single high level task

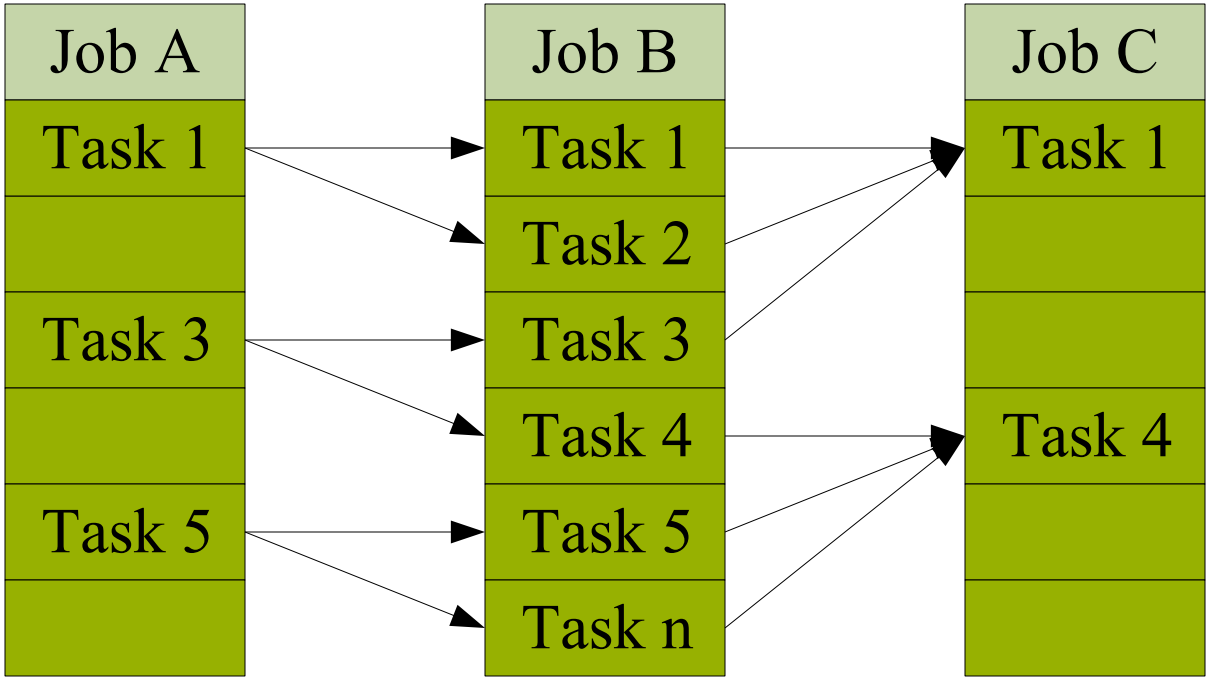
New Array Task Dependency

- Added real Array Task Dependency



New Array Task Frame Chunking

- Added the possibility to mix and match chunk sizes



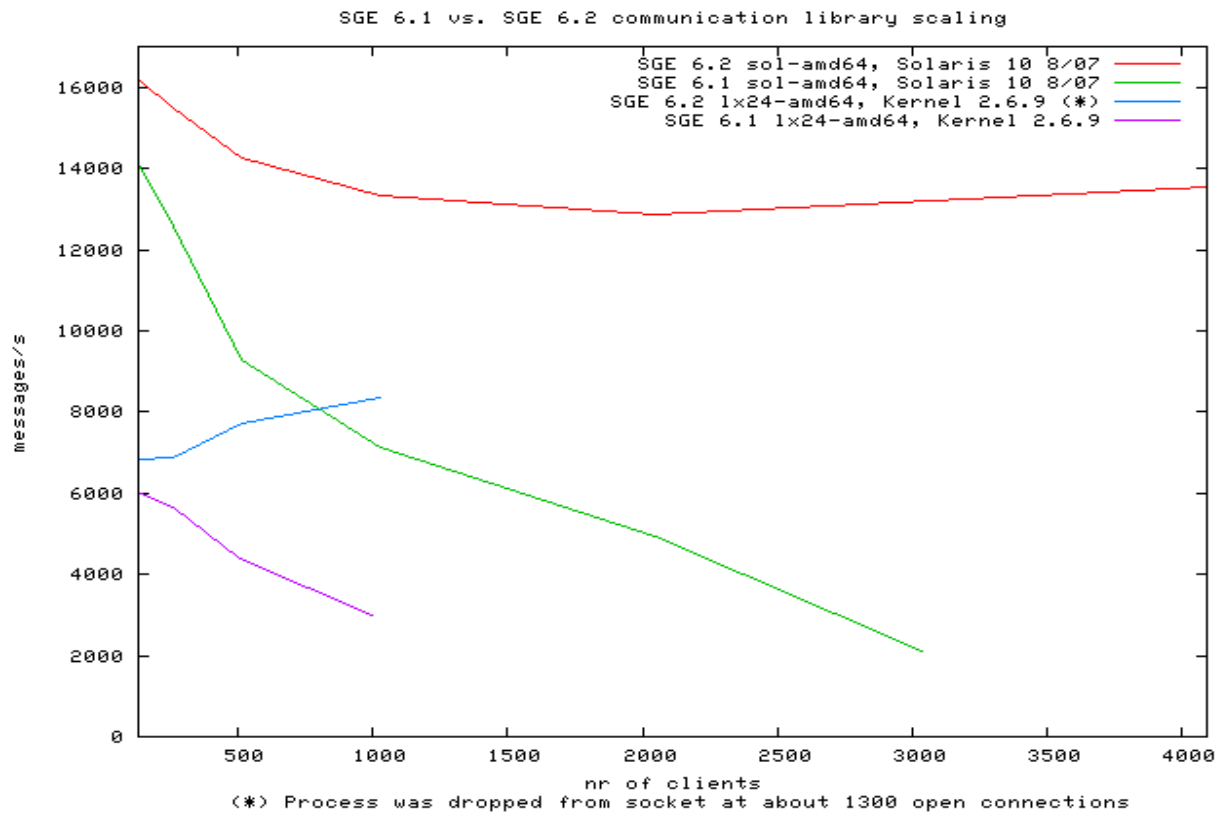
Scalability improvements

- Clusters become bigger and bigger
  - e.g. TACC using 3929 compute nodes
- Always a need for improving:
  - job throughput
  - resource utilization
  - Memory consumption
  - CPU usage

→ Targeted by various improvements

Communication Layer

- Replaced linear by hashed searches
- Manually optimized code for CPU and memory usage



Qmaster-Execd Protocol

- Reduced Data send to Execds as much as possible
  - Highly reduced data for tight integrated jobs
    - Scaled at TACC to ~4k x 16 core MPI jobs
  - Slightly reduced data for sequential jobs
- Only changed load values are now send to qmaster
  - Reduced idle load report size nearly by factor 4
  - Reduces qmaster CPU load
- Analyzed and fixed several code hotspots

Multithreading

- Introduced listener-worker model
  - Basement for further MT optimizations
- Moved scheduler from own process to a qmaster thread
  - Faster communication between scheduler and qmaster
  - Lower memory footprint
- Optimized Locking in EventMaster Thread
  - More precise event delivery time
  - Better scaling with more event clients

Scheduler

- United two-stage parallel matching code
  - Before: first *tag* then *select* queue instances
  - Now: joined to facilitate earlier quick exit
- Completed cluster queue matching
- ResourceQuota improvements (merged to 6.1u3)
  - United two-stage sequential matching code
  - Refine analysis of saturated RQ limit rules
  - Boost RQ limit evaluation through result caching

Additional Improvements

- Fixed a lot of non-linear scaling operations
  - qstat and qhost output
  - Job deletion
  - Queue modification
    - e.g 1.5k queue instances, 1000 running jobs from 8s to 2s
  - Qmaster startup
    - e.g. 1.5k queue instances, 1000 running jobs from 72s to 6s
- Reduced unnecessary copy operations
- Reduced execd CPU usage
- Lots of other unimpressive stuff

SMF Overview

Features

- Unified administrative model
- Service dependencies
- Self-healing services
- Parallel service startup
- Automatic snapshots of the repository per service instance
- Delegate tasks to non-root users with RBAC

SMF Overview

Terminology

- service - object that can be managed and observed
- instance - child of the service object
- FMRI - svc:/service_name:instance
- restarter - service responsible for restarting services
- milestone - predefined set of capabilities for a set of services

SMF Overview

Terminology cont.

- contract - keeps restarter informed about managed services
- manifest - description and initial configuration for a service (XML file)
- repository - configuration database for all services
- snapshot - for later rollback or inspection

SMF Overview

Commands

- svcs - report status of SMF services
- svcadm - enable/disable/restart SMF services
- svccfg - import/export/modify SMF manifest/repository
- svcprop - display properties for given service instance
- inetadm - svcadm for inetd based services
- inetconv - convert/import service from inetd.conf

SMF Overview

Service States

- uninitialized - after initial service import
- offline - service not running, configuration read
- online - service is running (is enabled)
- disabled - service is not running
- degraded - service is running with some failures (dependencies)
- maintenance - service is unavailable due to an error

Sun Grid Engine 6.2 SMF services

- Overview of supported services
- Behavior of SGE services
- Examples

Sun Grid Engine 6.2 SMF services

Overview of Supported Services

- svc:/application/sge/bdb
- svc:/application/sge/qmaster
- svc:/application/sge/shadowd
- svc:/application/sge/execd
- svc:/application/sge/dbwriter

Sun Grid Engine 6.2 SMF services

Overview of Supported Services cont.

- $SGE_ROOT/util/sgeSMF:
  - sge_smf.sh - script for import/deleting SGE services to/from the repository
  - sge_smf_support.sh - helper script for sge_smf.sh
  - bdb_template.xml
  - qmaster_template.xml
  - shadowd_template.xml
  - execd_template.xml
- At $SGE_ROOT/dbwriter/util/sgeSMF:
  - dbwriter_template.xml

Sun Grid Engine 6.2 SMF services

Behavior of SGE Services

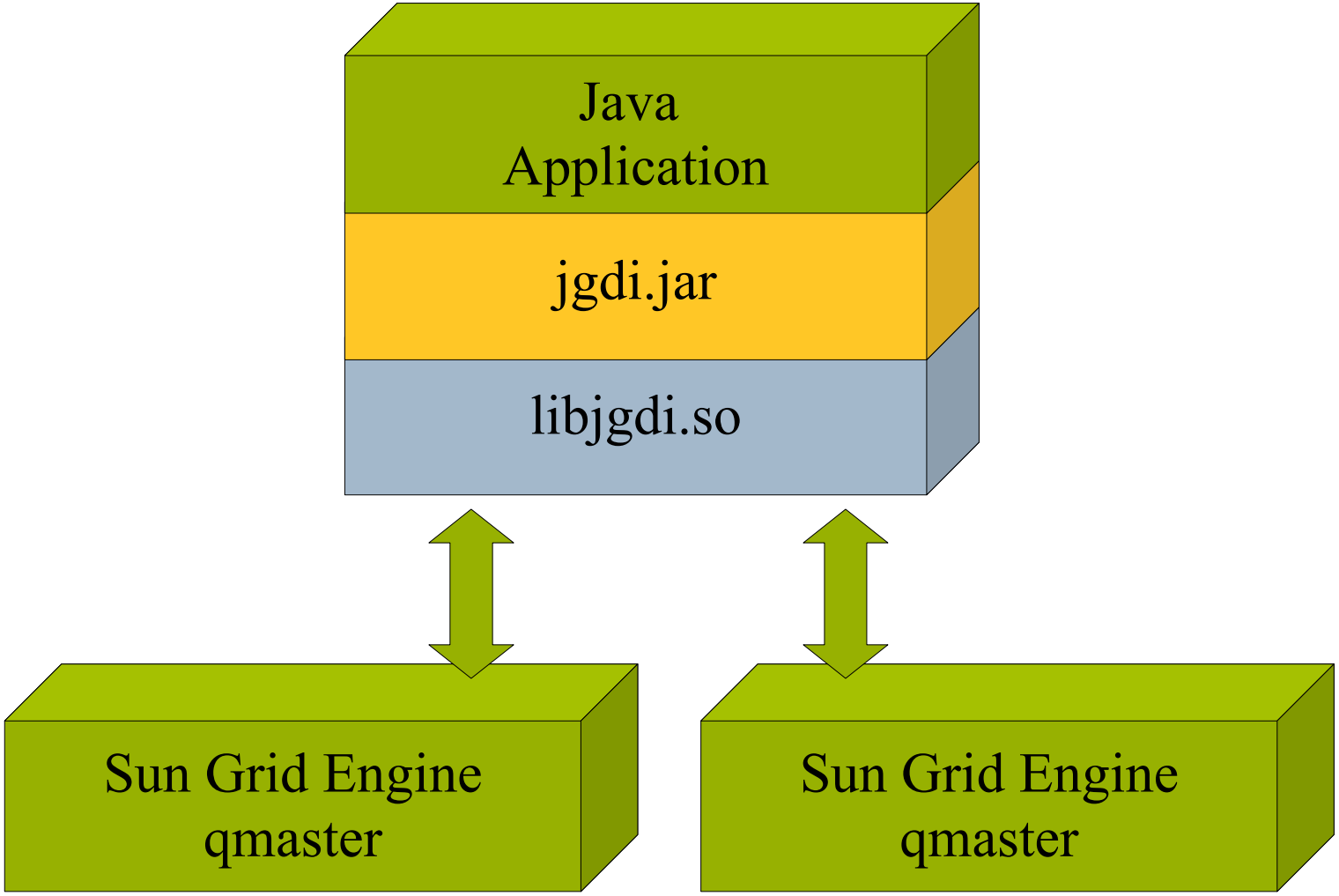
- Automatically started on reboot
- Restarted on error
- Cannot be killed with SIGKILL
- New interface svcadm

- application/sge/bdb
- application/sge/qmaster
- application/sge/shadowd
- application/sge/execd
- application/sge/dbwriter

JGDI Overview

- **Internal evolving interface !**
- Configuration & administration of SGE from java
- No job submission or accounting capabilities (DRMAA, ARCo)

JGDI Overview



JGDI example

See com.sun.grid.jgdi.examples.FirstExample

```
public class FirstExample {
  public static void main(String [] args)  {
      try {
          String url = args[0]; // bootstrap:///sge@osgc:1234
          JGDI jgdi = JGDIFactory.newInstance(url);
          System.out.println("Successfully connected to " + url);
          List<ClusterQueue> cql = jgdi.getClusterQueueList();
          for (ClusterQueue cq : cql) {
            System.out.println("Found cluster queue "+cq.getName());
          }
      } catch (JGDIException e) {
          e.printStackTrace();
      } finally {
          jgdi.close();
      }
  }
}
```

JMX Support

- JMX mbean server running in qmaster JVM thread
- Exposes JGDI functionality over JMX
- SDM GEAdapter uses event/qstat monitoring via JMX

JMX example

```
public void connect() throws GrmException {
    ...
    jgdiProxy = JGDIFactory.newJMXInstance(master, port, credentials);
    jgdi = jgdiProxy.getProxy();

    Set<EventTypeEnum> subscription = new HashSet<EventTypeEnum>(4);

    subscription.add(EventTypeEnum.ExecHostList);
    subscription.add(EventTypeEnum.ExecHostAdd);
    subscription.add(EventTypeEnum.ExecHostDel);
    subscription.add(EventTypeEnum.ExecHostMod);

    jgdiProxy.addEventListener(eventForwarder);

    jgdi.setSubscription(subscription);
    ...
```

Service Tags Overview

Introduction

- Automatic gear discovery
- Electronic labeling

Service Tags Overview

Introduction

- Automatic gear discovery
- Electronic labeling

Service Tags Overview

Components

- Discoverer – UDP based service to find listeners
- Listener – TCP based, gathers service tags and related environmental information
- Registry – local repository

Service Tags Overview

Commands

- stclient

Service Tags Support In SGE 6.2

- Register during qmaster startup
- Checked every time qmaster starts
- Removed on qmaster uninstall

Used Materials

- Hiregoudar G., Manus R. - SMF workshop material (Sun Microsystems)
- http://www.sun.com/bigadmin/content/selfheal/smf-quickstart.jsp

New Features in Sun Grid Engine 6.2

- **Roland Dittel, Lubomir Petrik**
- Software Engineers
- Sun Microsystems