Exploring Availability and Usage Guarantees in Resource Allocation Through Leases

Sara Sprenkle
Master’s Defense
January 21, 2004

Committee:
Jeff Chase            Carla Ellis            Dan Sorin
Motivation

- Cheap computing resources
- Standardized networking, communication protocols
- Large computing clusters
  - Provide large-scale services
  - Perform computations
  - Consumers: heterogeneous clients

Problem: underutilized resources

Solution: export resources to outside users
Big Picture

- static, long-lived domains
- dynamic resources
Motivation

- **Example Applications**
  - PlanetLab - shared global network testbed
    - run network apps from multiple sites
  - Web services
    - resources to handle demand spikes, maintain service quality
  - Cluster, Grid apps
    - large, computation jobs
    - network services
Problem

- Manager grants resources to consumers
- Consumer, manager want guarantees about use of resources
  - Consumer: exclusive ownership of resources for a period of time
  - service-quality guarantee
  - Manager
  - cost-effective decisions about resources
Leases

- Contract between granter (landlord) and consumer (tenant)
- Guarantees use of set of resources for a term
- Resource set
  - Abstract or concrete definition of resources
- Term - start time, expiration time
- Assume homogenous nodes
  - Extend to partial resources, heterogeneous nodes
Challenges in Lease Management

- Added complexity
  - Consumer: predict need for resources
  - Manager: how long to grant resources

- Manager assigns lease to resource set
  - Limits manager’s ability to adhere to allocation policy
    - how quickly reallocates resources
  - Affects utilization of resources
    - how quickly recovers from failure

- Solutions: implicit and contract-based leases
Hypothesis

The availability and usage guarantees provided by leases outweigh the management costs.
Outline

- Motivation
- Lease Management Challenges
- Resource Provisioning Environment
- Implicit Leases
  - Evaluate in COD
- Contract-based Leases
  - Model
  - Simple strategy
- Future Work
- Conclusions
Resource Provisioning Environment

Owner: owns the resources
Manager: manages resources on behalf of owners, allocates resources to consumers
Consumer: requests resources from manager, receives a lease on a slice of the resources
Implicit Leases

- Granted by manager
  - Not explicitly seen by consumer
  - If available again, manager renews the lease without notifying the consumer.

- Expire when consumer relinquishes or when manager revokes lease

- Short-term contracts: \((t_0, t_0 + \Delta)\)
  - start_time = \(t_0\)
  - expiration_time = \(t_0 + \Delta\)

- Benefits
  - Simple manager functionality
  - Reduced state for allocation
Cluster on Demand

- Resource provisioning environment
- Cluster manager
  - Allocates resources among competing *virtual clusters or slices*
  - Configures resources for vcluster
- Virtual Cluster Manager (VCM)
  - Application-specific service manager
  - Monitors service’s resource demands
  - Negotiates for resources
Cluster on Demand Prototype

- Sun’s GridEngine batch pools
  - Task scheduler
- SGE-VCM
  - Monitors queued jobs
  - Periodically resizes the vcluster
    - Policy: 1 resource for X pending jobs, relinquish node after idle for Y time
Experiments

- Real batch traces
  - Architecture, BioGeometry, and Systems groups

- Live Testbed
  - Devil Cluster (IBM, NSF)
    - 71-node COD prototype
  - Trace-driven--sped up trace to execute in 12 hours
  - Ran synthetic applications
  - Each group is its own vcluster
Live Test Results

Number of Nodes

Day1  Day2  Day3  Day4  Day5  Day6  Day7  Day8

Time

Number of Jobs

Day1  Day2  Day3  Day4  Day5  Day6  Day7  Day8

Time
Emulation Framework

- COD Manager, VCM unmodified from live tests
- Replaced SGE pool with Emulated SGE, simulator
- Simulator manages pending jobs, job processing, resources
Experiment: Allocation Policies

- **Priority-based**
  - Allocate resources in priority order
  - Priority-order: BioGeometry, Architecture, Systems
  - Minimum number of nodes: 2

- **Minimum Resource Guarantee**
  - Prevent starvation → configurable resource guarantee
  - Allocate 90% of 80 nodes among vclusters
  - Min( TotalRequestResources, GuaranteedMin)
Results: Strict Priority-Based

![Graph showing the number of nodes over time with different priority-based systems: Total, BioGeometry, Systems, and Architecture. The graph indicates that BioGeometry steals nodes from Architecture.]
Results: Min Guaranteed Resources

<table>
<thead>
<tr>
<th>Time (Days)</th>
<th>Number of Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
</tr>
</tbody>
</table>

Legend:
- Total
- BioGeometry
- Systems
- Architecture
Overlay of Results

<table>
<thead>
<tr>
<th>Number of Nodes</th>
<th>Time (Days)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Systems</td>
<td>Arch</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>requested</td>
<td></td>
</tr>
<tr>
<td>nodes</td>
<td></td>
</tr>
</tbody>
</table>

Chart showing the overlay of results with lines representing different categories such as Total, BioGeometry, Systems, and Architecture against the number of nodes over time (in days).
Implicit Lease Discussion

- Manager can exactly match allocation policy
  - Follow policy immediately
  - Policy thresholds prevent thrashing between slices
- Only short-term guarantees to consumer, manager
Outline

- Motivation
- Lease Management Challenges
- Resource Provisioning Environment
- Implicit Leases
  - Evaluate in COD
- Contract-based Leases
  - Model
  - Simple strategy
- Future Work
- Conclusions
Contract-based Leases

- Non-revocable contracts for resources
  - Consumer cannot relinquish either

- Benefit: guarantees for consumer, granter

- Cost: more complex manager
  - Limited allocation policies

- Used in SHARP [Fu03]

- How should the manager choose the lease term for an allotment of resources?
Choosing a Lease Term

- **Resource efficiency**
  - Shorter leases $\Rightarrow$ higher resource efficiency

- **Agility**
  - Shorter leases $\Rightarrow$ higher agility

- **Failure recovery**
  - Shorter leases $\Rightarrow$ faster failure recovery

- **Renewal overhead**
  - Longer leases $\Rightarrow$ less renewal overhead
Evaluation Metrics

- **Granter load: message overhead**
  - Practicality of policy

- **Allocation policy metric**
  - Compare with and without leases

- **Resource Efficiency**
  - Measure waste because of poor allocation policies
Visualizing Resource Efficiency

- **Allocated Resources**
- **Wasted Resources**
- **Actual Resource Use**

Axes:
- **Resources**
- **Time**
Manager’s Considerations

- Renewal request load
- Overallocated resources
- Inability to match allocation policy
- Available resources
- Types of resources
- Fairness to consumers
- …

- Manager’s policies are complex
Acquiring Leases

- Assume: manager allocates leases as the consumer requests
  - No denied requests (infinite resources)
  - No partially fulfilled requests
  - Consumer has power

- Give consumer incentive to use resources efficiently
  - Assign cost to leases

- How will the consumer acquire leases?
Lease Cost

- Overhead lease cost
  - Account for manager’s overhead
  - Assume fixed cost

- Resource cost
  - Amount, length of resources
  - Properties of resource cost function:
    - Spot price for immediate use of resource
      - immediate use == more valuable
    - Reduced price for buying in advance
    - Minimum resource price
Family of Resource Cost Functions

\[ c(t) = \beta^\alpha \frac{(sp - mp)}{(t + \beta)^\alpha} + mp \]

- \( sp \) = spot price
- \( mp \) = min res price
- \( \alpha \) and \( \beta \) = decay parameters

Graph showing the family of resource cost functions with different parameters for \( sp \) and \( mp \).
Lease Cost

- Fixed, overhead lease cost
- Cost of resources

\[ C(t) = \text{lease\_price} + \text{num\_resources} \int_s^e c(t) dt \]
Consumer’s Strategy

- Advanced reservations
- Long-term contracts
- Few leases
- Risk vs reward
  - Risk: buying too many resources
  - Reward: cheaper to buy resources in bulk, in advance
Consumer’s Strategy

- Cover demand curve with leases with minimum cost
- Smaller leases: less waste, higher cost

Resources

Allocated resources

Total cost of coverage = sum of costs of all leases

Lease (start, end, res) has a cost
Simple Strategy

- Fixed-sized leases
  - Fixed length, number of resources

- Consumer buys fixed-length leases to cover resource demand curve
  - If demand is greater than leased resources, buy more leases
  - Always buys at spot price
  - No lease renewals

- What is the fixed-size lease that covers demand with
  - minimum wasted resources?
  - minimum cost?
Example Using Simple Strategy

Jobs take 100 secs
Request rate: 1 job/sec
Requested for 100 sec

Lease time = 5
Lease resources = 10

Resource demand

Allocated resources

Time

Resources
Wasted Resources

minimum wasted resources when res=1, time=1

wasted resources

lease resources

lease time
Total Lease Coverage Cost

Cost Function Parameters:
- $sp = 0.05 \quad \alpha = 0.4$
- $mp = 0.01 \quad \beta = 1000$
- lease price = 5

$\min \text{cost} \approx 688$
- time = 19
- res = 25
Discussion

- Finding optimal fixed-size lease is not intuitive
  - Simplified problem!
- Minimum wasted resources, cost are opposing goals
- Cost function penalizes more leases
  - High lease overhead: equivalent to about 10 resources for 10 seconds
  - Less penalty for overestimating resources
  - Important to choose appropriate cost-function parameters
Outline

- Motivation
- Lease Management Challenges
- Resource Provisioning Environment
- Implicit Leases
  - Evaluate in COD
- Contract-based Leases
  - Model
    - Simple strategy
- Future Work
- Conclusions
Future Work

- Lease model for other metrics (not presented)
  - Current: fixed job lengths, lease sizes, constant request rate
  - Extend to more general model
  - More insights to appropriate lease strategies

- Known resource demand curve
  - More sophisticated lease assignment strategies
    - Longer-term leases
    - Advanced reservations
    - Example: long lease to cover most demand, shorter lease to handle bursts on the fly
Future Work

- Contract-based leases’ effect on policy, application performance
  - Implement and evaluate strategies in SHARP
- Allow consumers to terminate contract-based leases early
  - Constraint enforced by SHARP security policy
  - Decreases consumer’s risk in buying leases
  - Manager can utilize resources more efficiently
    - If consumer pays penalty for early termination, manager can recoup expenses
Conclusions

- Benefits and costs of leases
  - Provide usage guarantees at cost of management complexity
- Implicit leases
  - No restriction on executing allocation policy
  - Sharing benefit but only temporary resource guarantees
- Contract-based leases
  - Enforce guarantees
  - Guidelines for assigning leases
  - Model to explore lease-assignment strategies
  - Opportunity for more research
Acknowledgements

- Jeff Chase
- Dan Sorin
- Andy Danner
- Laura Grit
- Justin Moore
- The Systerhood
- Ron Doyle
- Xtreme Blue Cambridge 2003
- Glenn Sockhead
- Carla Ellis
- Mom, Dad, Jessie
- Patrick Reynolds
- David Irwin
- Yun Fu
- 1999 Incoming Class
- Diane Riggs
- Lori Pollock
- Darrell Anderson
Architecture Close-up

Nodes allocated in response to demand, more quickly when contention is low.
Simple Lease Model

- 1 manager, 1 consumer

- Variables
  - Fixed-length jobs \( J \)
  - Fixed-length lease \( L \)
  - Constant request rate \( r \)
  - Denial rate \( d \)
  - Net acceptance rate \( a = r - d \)
Analysis of Model
Example Using Simple Strategy

<table>
<thead>
<tr>
<th>Time</th>
<th>Resources</th>
<th>Resource demand</th>
<th>Lease time</th>
<th>Lease resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>100</td>
<td>80</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>90</td>
<td>70</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>80</td>
<td>60</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>40</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>40</td>
<td>30</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>30</td>
<td>20</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

(Chart showing time vs. resources with key points indicated.)
SHARP Resource Negotiation Protocol
Related Work

- Resource Provisioning
  - Hierarchical SLAs
- Leases in distributed systems
  - Distributed file caches
    - Gray, Cheriton
    - Storage Tank - optimistic renewal
    - Squid - volume leases
  - Service management - Jini
  - Efficiency in a highly available system - Lampson