



Celeste: how it really works.

Glenn Scott

Senior Researcher

Sun Microsystems, Inc.



**2007
Sun Labs
Open House**



Celeste

- Storage System
 - > Distributed
 - Nodes perform the storage and retrieval.
 - > Peer-to-peer
 - No master.
 - > Dynamic
 - Ad hoc arrangement and configuration of nodes.
 - > Untrusted infrastructure
 - Anticipates failure and malfeasance.

Celeste: how it really works

- Celeste Is Built Using “Beehive”
- Distributed Object Platform for Services
 - > Resilient
 - Operates despite arbitrary failures and maliciousness.
 - > Scalable
 - Resources increase or decrease linearly as the system expands or shrinks.
 - > Securable
 - Nothing is trusted. All operations are observable and verifiable during and after-the-fact.

Distributed Hash Table (DHT) Routing

- Beehive is built out of a virtual network of nodes
 - > Overlay network on top of an underlying IP inter-network.
- Used to route messages between nodes
 - > Nodes have globally unique identifiers (node-id)
 - B898C7B8A9188C87CD2F39ACE68607DC
 - Identifiers are verifiable, one node cannot masquerade as another.
 - > Messages are sent to nodes by specifying the node-id.
 - Maximum number of hops is small compared to network size
 - $O(\log N)$
 - > Any destination node-id is valid regardless of existence.
 - Messages arrive at the node with “closest” node-id
 - This is called the “root node”

Distributed Object Location and Retrieval

Distributed Objects

- Objects are static bags-o-bits stored on nodes.
 - > All objects consist of meta-data and raw, uninterpreted data.
- Object identifiers are like node identifiers (object-id).
 - > Object identifiers can be verified by the object's content and meta-data.

Distributed Object Location and Retrieval

Distributed Object Location

- Add message types.
- Messages to announce object availability and unavailability.
 - > Publish-Object – announce availability of object.
 - > Unpublish-Object – announce unavailability of object.
- Messages are routed to root node of object-id.
 - > Root node adds or deletes the record of object object-id's availability.

Distributed Object Location and Retrieval

Distributed Object Location and Retrieval

- Message to retrieve an object.
 - > Route-To-Object
- Requester transmits message to object-id.
- Message is received by the root node of object-id.
 - > Root node acts as a proxy.
 - Sends message to node with object to fetch the object and return it.
 - > Root node informs requester of the location.
 - Requester sends message to node with object to fetch it.

Object Caching

- The same object may be stored on several nodes.
 - > The root node of object-id records all locations.
- Nodes in the transit path of a publish message may also record object availability as an optimisation.
 - > Subsequent “retrieve” messages may not have to travel all the way to the root of object-id.

- **Ben Y. Zhao, Ling Huang, Jeremy Stribling, Sean C. Rhea, Anthony D. Joseph, and John Kubiawicz. Tapestry: A resilient global-scale overlay for service deployment. IEEE Journal on Selected Areas in Communications, 2003.**
- **C. Greg Plaxton, R. Rajaram, and A. Richa. Accessing nearby copies of replicated objects in a distributed environment. In Proceedings of the Ninth Annual ACM Symposium on Parallel Algorithms and Architectures, pages 311--320, June 1997.**
- **I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. Technical Report TR-819, MIT, March 2001.**

Object Decay

- Objects have a time-to-live (TTL) encoded in their meta-data.
 - > Time-to-live decrements periodically until zero and the object is forcibly deleted.
- Different nodes hosting an object may have different TTL for the object.
 - > Cached.
 - > Archived.

Object Behaviour

- Obviate the simple “R” in DOLR.
- Permit more sophisticated queries of objects.
 - > An object has a “type” or class reflecting the queries an object must respond to.
- Objects have behaviour.
 - > Queries may induce computation in the object.
 - > Object queries may involve other objects indirectly.
 - Objects can be interdependent.

Object Deletion

- Antithesis of the DOLR credo.
 - > The DOLR works hard to never forget.
- Implement delete behaviour in objects.
 - > Induce deletion by creating an anti-object for each object to delete.
 - > Anti-objects meet their object counterparts at their common object-id root, and convert them to anti-objects.
- Deleted objects leave behind their anti-object “husk.”
 - > Anti-objects continue to infect their objects over time.
 - > Anti-objects decay to nothing.
 - System reclaims space and resources automatically.

- **Germano Caronni, Raphael Rom, Glenn Scott. Maintaining Object Ordering in a Shared P2P Storage System. IEEE 3rd International Security in Storage Workshop, San Francisco, CA, December 2005.**
- **Gal Badishi, Germano Caronni, Idit Keidar, and Raphael Rom, Glenn Scott. Deleting Files in the Celeste Peer-to-Peer Storage System. In Proceedings of the 25th IEEE Symposium on Reliable Distributed Systems (SRDS'06).**

Mutable Objects

- Objects have update behaviour
 - > Objects have set and get semantics.
- Implications:
 - > Object-ids can no longer be derived from just the content.
 - > Cached copies of an object may be out-of-date.
 - > No way to guarantee that a retrieved object is up-to-date.

Mutable Objects

- Mutable objects are no longer just bags-o-bits.
 - > An object's current state is represented by a set of intermediary objects.
 - This means the object is now virtual.
 - > Updating an object requires the update of a minimum number of intermediary objects.
 - > Querying an object requires the query and agreement of a minimum number of intermediary objects.

- **James Cowling, Daniel Myers, and Barbara Liskov. HQ Replication: A Hybrid Quorum Protocol for Byzantine Fault Tolerance. 7th USENIX Symposium on Operation Systems Design and Implementation, November 2006.**
- **Michael Abd-El-Malek, Gregory R. Ganger, Garth R. Goodson, Michael K Reiter, Jay J. Wylie. Fault-Scalable Byzantine Fault-Tolerant Services. SOSP, October 23-26, 2005**



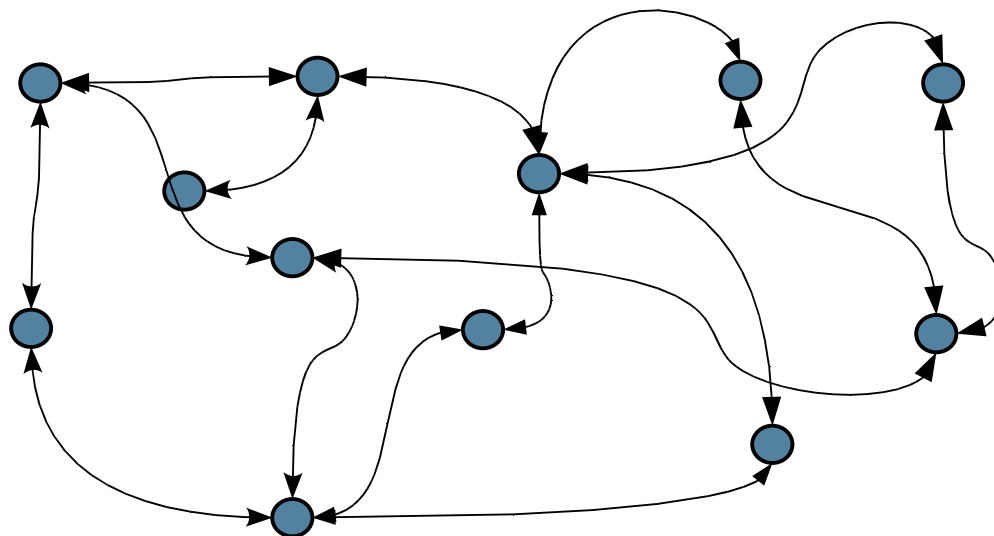
Glenn Scott

Glenn.Scott@sun.com



**2007
Sun Labs
Open House**





ABCD			
ABCD			ABCA
	ABCD		
CBAB		ABCD	
DABA			ABCD

ABCA			
ABCA			ABCA
	ABCA		
CBAB		ABCA	
DABA			ABCD

CBAB			
ABCD		CBAB	
	CBAB		CBAB
CBAB			
DABA			

DABA			
ABCD	DABA		DABA
		DABA	
CBAB			
DABA			

CABB			
ABCD	CABB		ABCD
		CABB	
CABB	CBAB		
DCBA			

