Xest: a file system for wide-area virtual machine deployments

Tim Moreton
tim.moreton@cl.cam.ac.uk

30 August 2005

Storage for VMs in the wide-area

- *XenoServers*: public ‘edge’ computing:
  - Control of service deployments (per VM root fs)
  - Network position (response time, content distribution)
- Challenges for the storage infrastructure:
  - VM migration, instantiation
  - High latency, constrained bandwidth
  - Scale: large FS images, 10k running VMs
  - Making huge deployments manageable for admins
  - Access control, billing, auditing
  - Multiplexing VMs on a machine [XenFS]

Making it tractable

- Nomadic disks, not a global distributed file system
  - Provide migrating ‘local’ storage, do sharing later
  - Well-established CoW techniques [P9,WAFL,Parallax,..]
- Exploit commonality whereever it exists
  - Large distro images but significant root FS sharing
- *Grouping:* organise FS by observed working sets
  - Coarser granularity of operations
  - Implicit prefetch: access an obj, fetch its group

Nomadic, isolated, virtual disks

- *Nomadic*: transparently available despite migration
- *Isolated*: only writeable at a single VM at once
- Shared storage is convenient, simplifies administration
- But collaborative use and short-term data exchange rare
- Concurrency control limits performance/availability
- Optimistic consistency semantics risks conflicts
- Instead: Disallow sharing, `fork` VDs for each VM
- Add `merge` primitive to reconverge VDs (offline, ASRs)

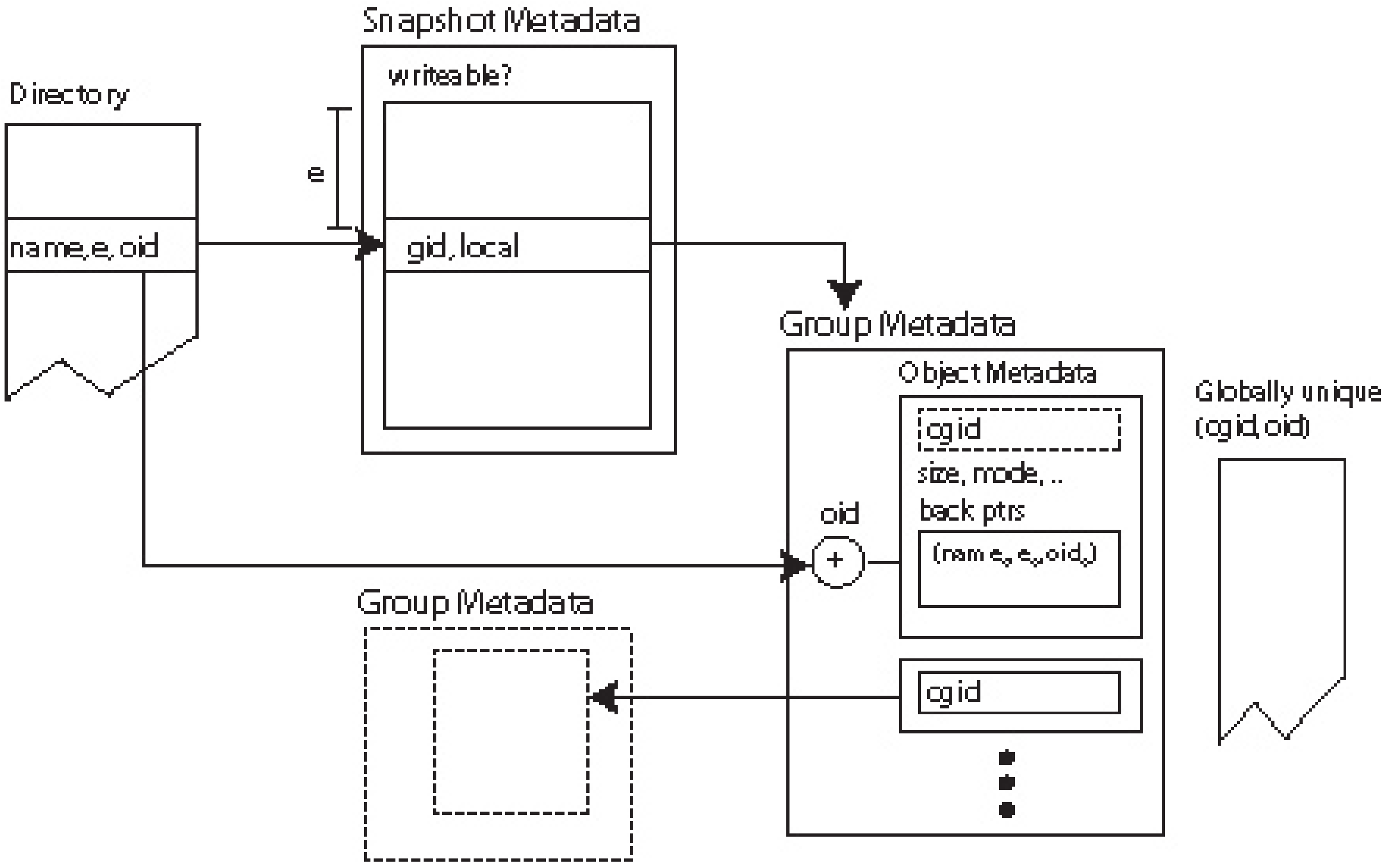
Eliciting, not engineering, commonality

- Implicit: shared content in CoW ancestors
- Explicit: VD merges, b/g process to detect shared content
- Departure from overlays model [WORLDS‘04]
  - Applying updates to templates problematic
  - Only exploits commonality between VD and its template
  - `merge` into template requires control of all overlays

Working set centric storage

- XenoServers are *necessarily* globally dispersed
- Latency, b/w bottleneck for deploying, migrating VMs
- But mitigating techniques hampered by latency and scale:
  - Prefetching: distant prefetch horizon due to high RTTs
  - Pervasive replication: scalability of addressing
- Grouping: organise FS by observed working sets
- Observe references, run clustering alg on closed VD
- Decouple wide-area interactions from local FS activity
- Reduces state overhead but limiting effects of aliasing
- Name, advertise, retrieve whole groups at once

Storage organization



Addressing

- Integrates with Bamboo DHT to manage membership
- Eventually consistent despite failure, partition
- A VD is a linear series of snapshots
- VD metadata in DHT by id and by H(name)
- Nodes caching groups maintain pointers in DHT

Regrouping

- Jarvis-Patrick: If $|N_{o_1} \cap N_{o_2}| \geq n$, merge($G(o_1)$,$G(o_2)$)
- Incremental: split by DFS marking before merge
- Multi-pass: merge between closely related objects first, and stop early if we reach acceptable group size
- Overlaps: Objects accessed from many distinct working sets (e.g. a `.so`) are fetched and cached with each group
- Statistics are of the form: *'Given an access to o_1, with what probability was o_2 also accessed within k references?'*
- Intuitively, groups are 'basic blocks' ('chains' of refs)

No clustering algorithm is perfect

- Demand fetches take priority over prefetches:
  - Group-based object location but sub-group fetches
  - Fetches on which FS is stalled preclude prefetches
- Hybrid Adaptive Caching (HAC) for the cache:
  - A group may contain hot and cold objects
  - Remove cold objs from groups .. TODO
  - Cost to evicting cold objs: ptr advertises whole group
  - Two ‘grades’ of ptr to maintain availability of hot objs

Deployment status

- Deployment nearly ready (10 days away)...
- Imitate wide-area XenoServer deployment over PlanetLab
- Trace server workloads and replay in migratory setting
- Evaluate miss rate, effect on stall time, addressing overhead, ...
- Results from a simulation study show a considerable reduction in stall time is possible, but are traded off against unnecessary fetches due to grouping inaccuracy