

Ganeti

A cluster virtualization manager.

Guido Trotter <ultrotter@google.com>

- Google, Ganeti, Debian

© 2010-2013 Google

Use under GPLv2+ or CC-by-SA

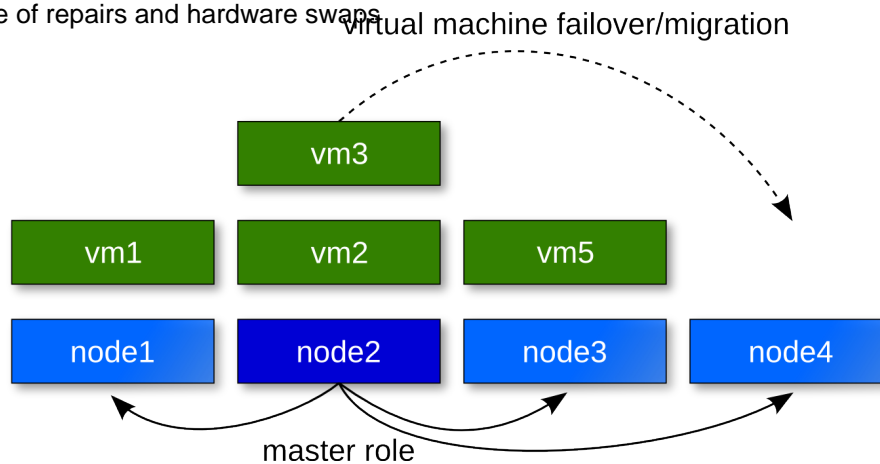
Some images borrowed/modified from Lance Albertson and Justin Pop

Outline

- Introduction to Ganeti
- Latest features
- Using Ganeti in practice
- How Ganeti is deployed at Google

What can it do?

- Manage clusters of physical machines
- Deploy Xen/KVM/lxc virtual machines on them
 - Live migration
 - Resiliency to failure (data redundancy over DRBD, or RBD)
 - Cluster balancing
 - Ease of repairs and hardware swaps



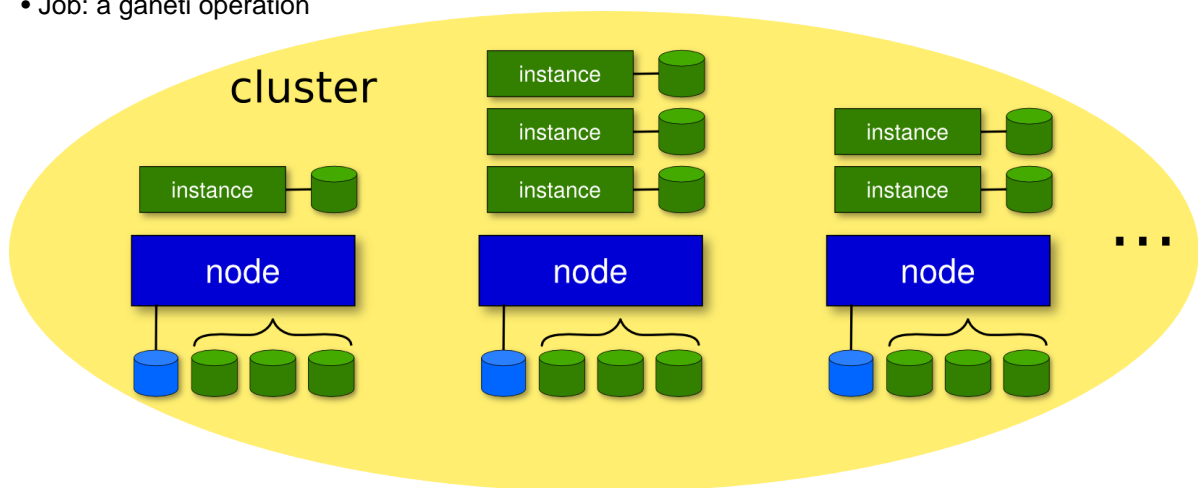
Ideas

- Making the virtualization entry level as low as possible
 - Easy to install/manage
 - No specialized hardware needed (eg. SANs)
 - Lightweight (no "expensive" dependencies)
- Scale to enterprise ecosystems

- Manage simultaneously from 1 to ~200 host machines
- Access to advanced features (drbd, live migration)
- Be a good open source citizen
 - Design and code discussions are open
 - External contributions are welcome
 - Cooperate with other Ganeti users

Terminology

- Node: a virtualization host
- Nodegroup: an omogeneous set of nodes
- Instance: a virtualization guest
- Cluster: a set of nodes, managed as a collective
- Job: a ganeti operation



Technologies

- Linux and standard utils (iproute2, bridge-utils, ssh)
- KVM/Xen/LXC
- DRBD, LVM, SAN, files or RBD
- Python (plus a few modules)
- socat
- Haskell



Node roles (management level)

- Master Node
 - runs ganeti-masterd, rapi, noded, confd and mond
- Master candidates
 - have a full copy of the config, can become master
 - run ganeti-confd, noded and mond
- Regular nodes
 - cannot become master
 - get only part of the config
 - run noded, and mond
- Offline nodes, are in repair

Node roles (instance hosting level)

- VM capable nodes
 - can run virtual machines
- Drained nodes
 - are being evacuated
- Offlined nodes, are in repair

Newer features

- Master IP turnup customization
- Out of Band management
- full SPICE support (KVM)
- Node health/power/epo commands (OOB)

New features in 2.6

The very stable version (since Jul 2012):

- RBD support (ceph)
- initial memory ballooning (KVM, Xen)
- cpu pinning
- OVF export/import support
- support for customizing drbd parameters
- policies for better resource modeling

New features in 2.7

Release candidate:

- Network management
- External storage
- Exclusive storage (for LVM)

- Hroller
- Linux HA agents (experimental)
- Openvswitch

New features in 2.8

At beta stage:

- Monitoring agent (mond) (drbd and instance status)
- Exclusive CPUs
- Discrete instance policies
- Autorepair tool (harep)
- Job reason trail
- Configuration downgrading

What to expect

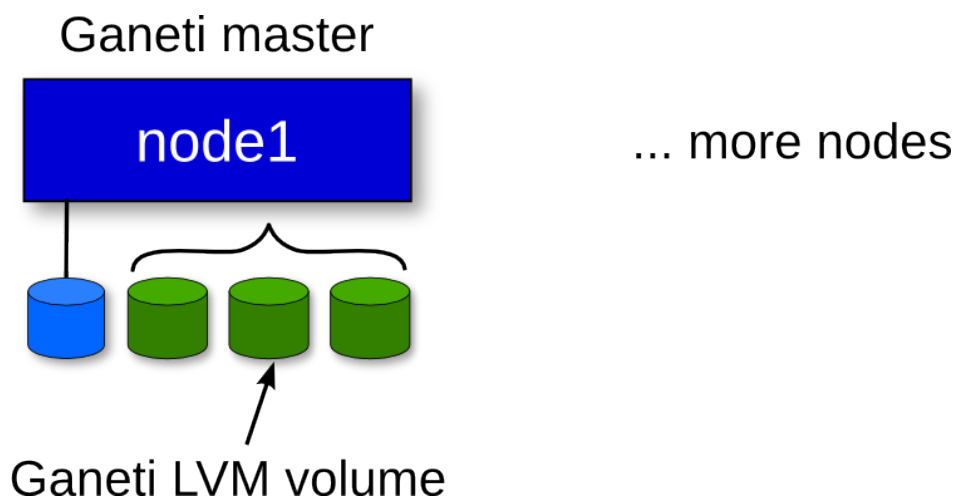
Just ideas, not promises:

- Better openvswitch integration
- Better ceph integration
- Gluster backend
- Non-transparent hugepages support
- Rolling reboot
- Better automation, self-healing, availability
- KVM block device migration
- Better OS installation

Initializing your cluster

The node needs to be set up following our installation guide.

```
gnt-cluster init [-s ip] ... \
  --enabled-hypervisors=kvm cluster
```



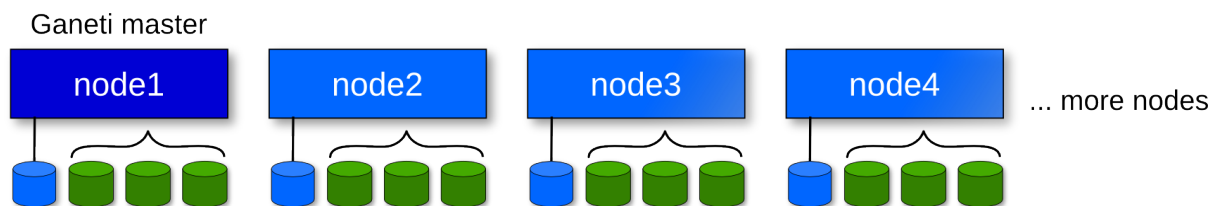
gnt-cluster

Cluster wide operations:

```
gnt-cluster info
gnt-cluster modify [-B/H/N ...]
gnt-cluster verify
gnt-cluster master-failover
gnt-cluster command/copyfile ...
```

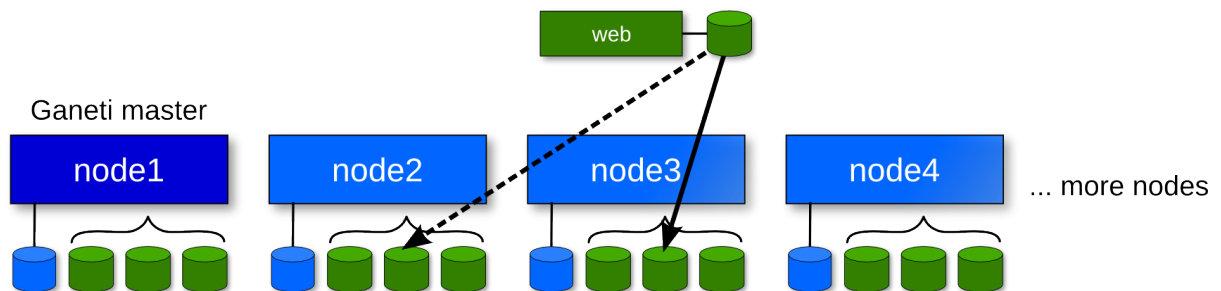
Adding nodes

```
gnt-node add [-s ip] node2
gnt-node add [-s ip] node3
```



Adding instances

```
# install instance-{debootstrap, image}
gnt-os list
gnt-instance add -t drbd \
  {-n node3:node2 | -I hail } \
  -o debootstrap+default web
ping i0
ssh i0 # easy with OS hooks
```



gnt-node

Per node operations:

```
gnt-node remove node4
gnt-node modify \
  [ --master-candidate yes|no ] \
  [ --drained yes|no ] \
  [ --offline yes|no ] node2
gnt-node evacuate/failover/migrate
```

```
gnt-node powercycle
```

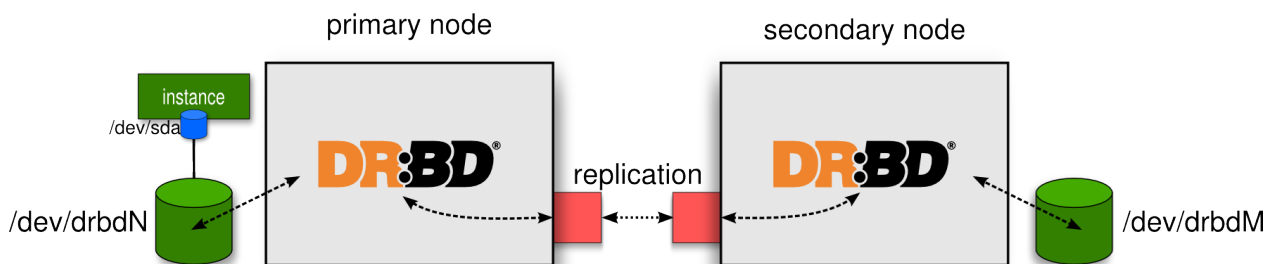
gnt-instance

Instance operations:

```
gnt-instance start/stop i0
gnt-instance modify ... i0
gnt-instance info i0
gnt-instance migrate i0
gnt-instance console i0
```

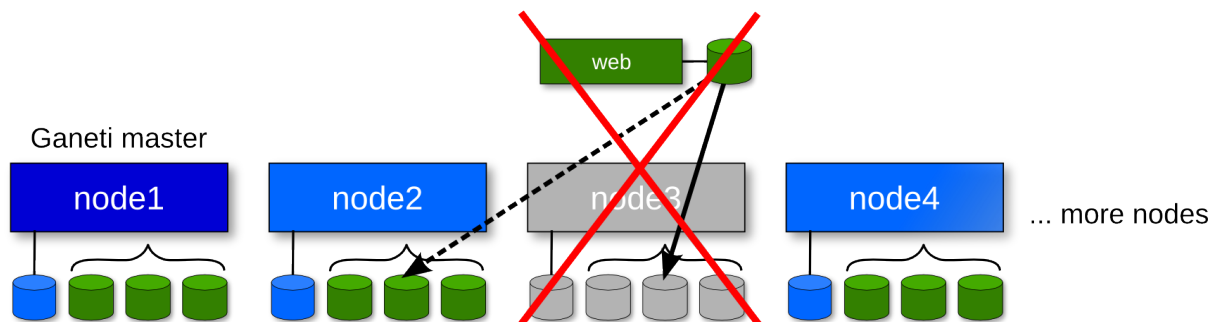
-t drbd

DRBD provides redundancy to instance data, and makes it possible to perform live migration without having shared storage between the nodes.



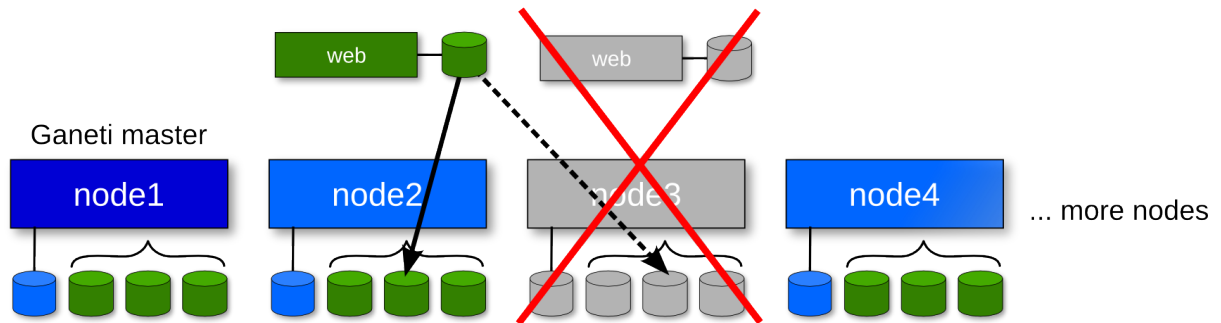
Recovering from failure

```
# set the node offline
gnt-node modify -O yes node3
```



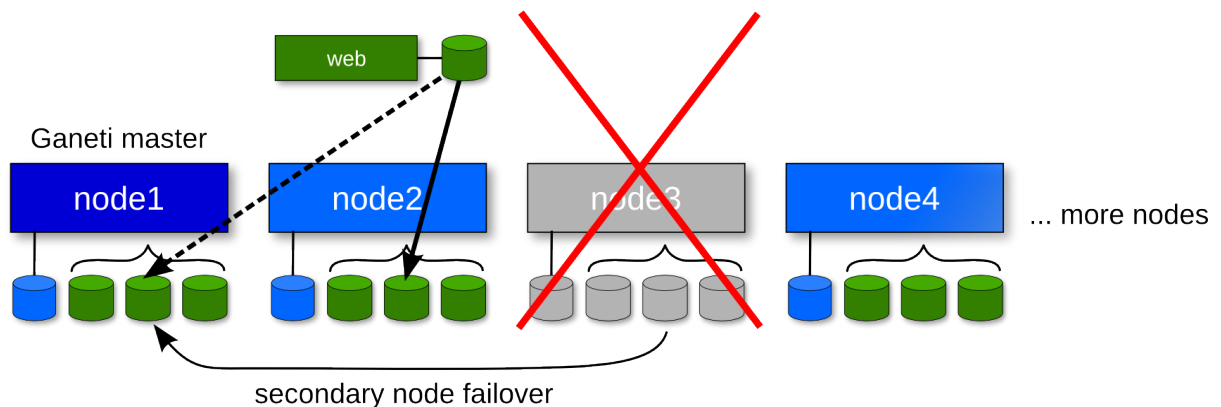
Recovering from failure

```
# failover instances to their secondaries
gnt-node failover --ignore-consistency node3
# or, for each instance:
gnt-instance failover \
  --ignore-consistency web
```



Recovering from failure

```
# restore redundancy
gnt-node evacuate -I hail node3
# or, for each instance:
gnt-instance replace-disks \
  {-n node1 | -I hail } web
```



gnt-backup

Manage instance exports/backups:

```
gnt-backup export -n node1 web
gnt-backup import -t plain \
  {-n node3 | -I hail } --src-node node1 \
  --src-dir /tmp/myexport web
gnt-backup list
gnt-backup remove
```

htools: cluster resource management

Written in Haskell.

- Where do I put a new instance?
- Where do I move an existing one?
 - hail: the H allocator
- How much space do I have?
 - hspace: the H space calculator
- How do I fix an N+1 error?
 - hbal: the cluster balancer

Controlling Ganeti

- Command line (*)
- [Ganeti Web manager](#)
 - Developed by osuosl.org and grnet.gr
- RAPI (Rest-full http interface) (*)
- On-cluster "luxi" interface (*)
 - luxi is currently json over unix socket
 - there is code for python and haskell

(*) Programmable interfaces

Job Queue

- Ganeti operations generate jobs in the master (with the exception of queries)
- Jobs execute concurrently
- You can cancel non-started jobs, inspect the queue status, and inspect jobs

```
gnt-job list
gnt-job info
gnt-job watch
gnt-job cancel
```

gnt-group

Managing node groups:

```
gnt-group add
gnt-group assign-nodes
gnt-group evacuate
gnt-group list
gnt-group modify
gnt-group remove
gnt-group rename
gnt-instance change-group
```

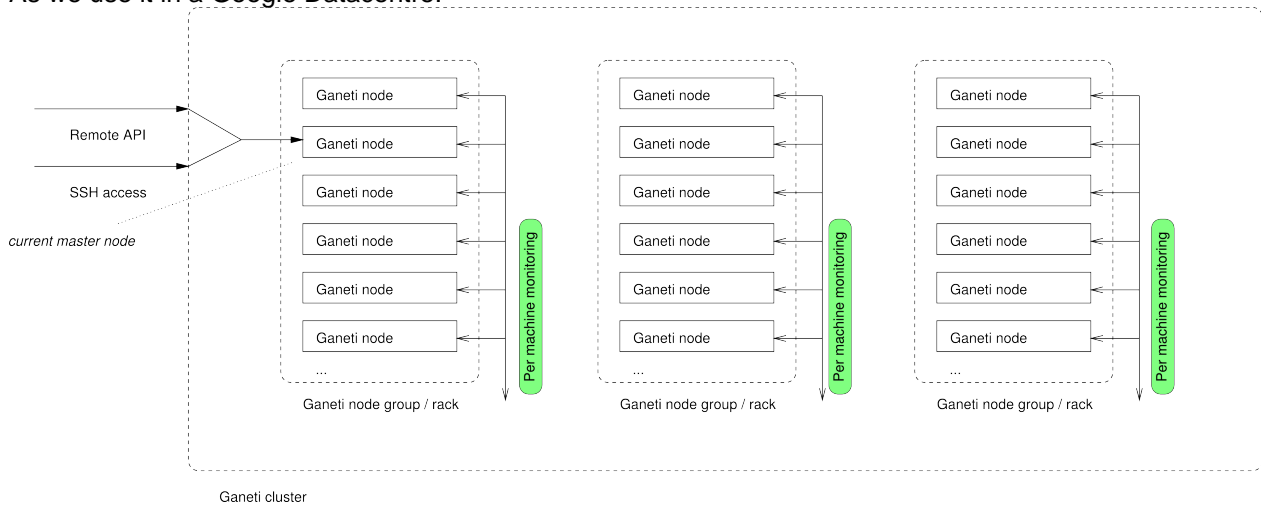

Running Ganeti in production

What should you add?

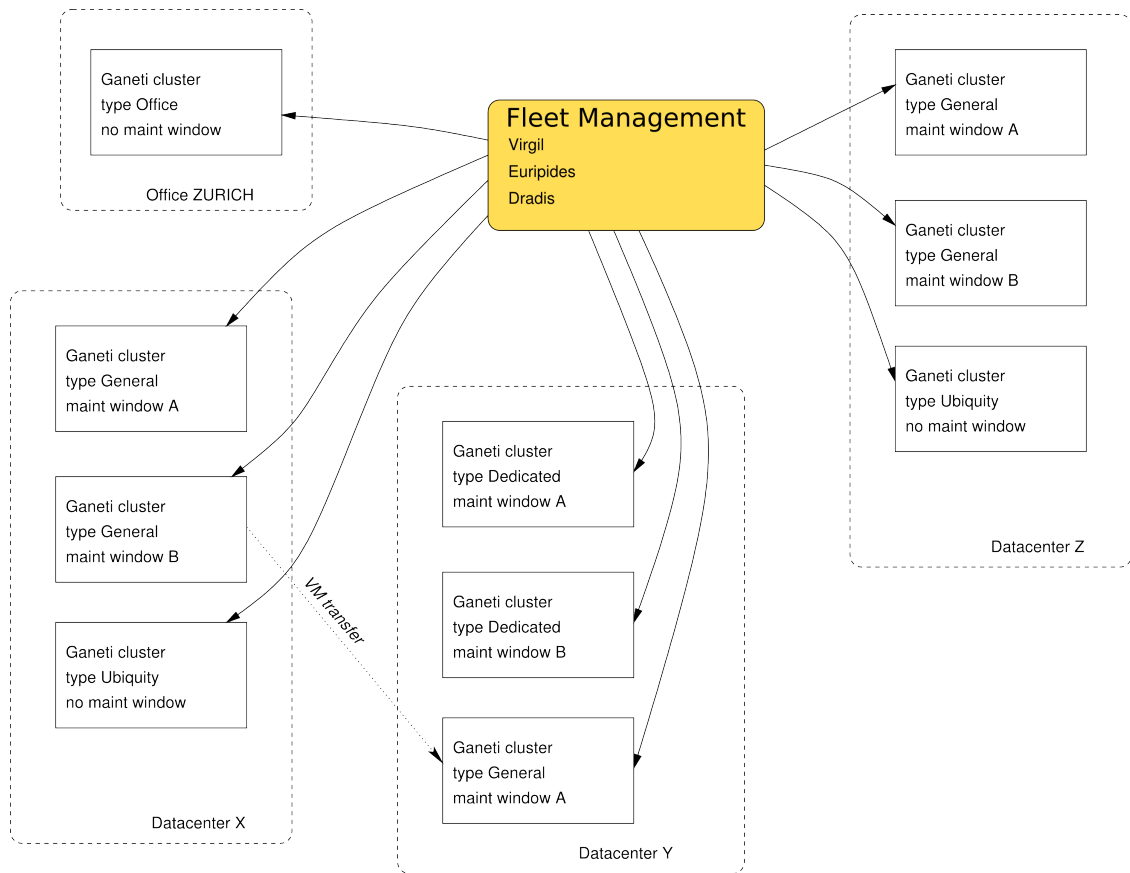
- Monitoring/Automation
 - Check host disks, memory, load
 - Trigger events (evacuate, send to repairs, readd node, rebalance)
 - Automated host installation/setup (config management)
- Self service use
 - Instance creation and resize
 - Instance console access

Production cluster

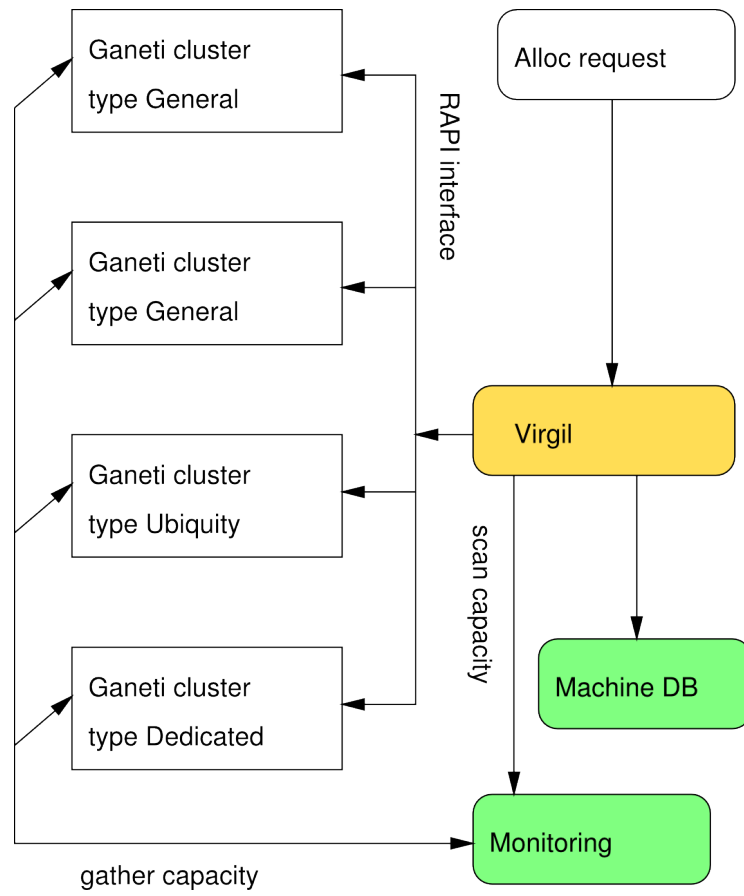
As we use it in a Google Datacentre:



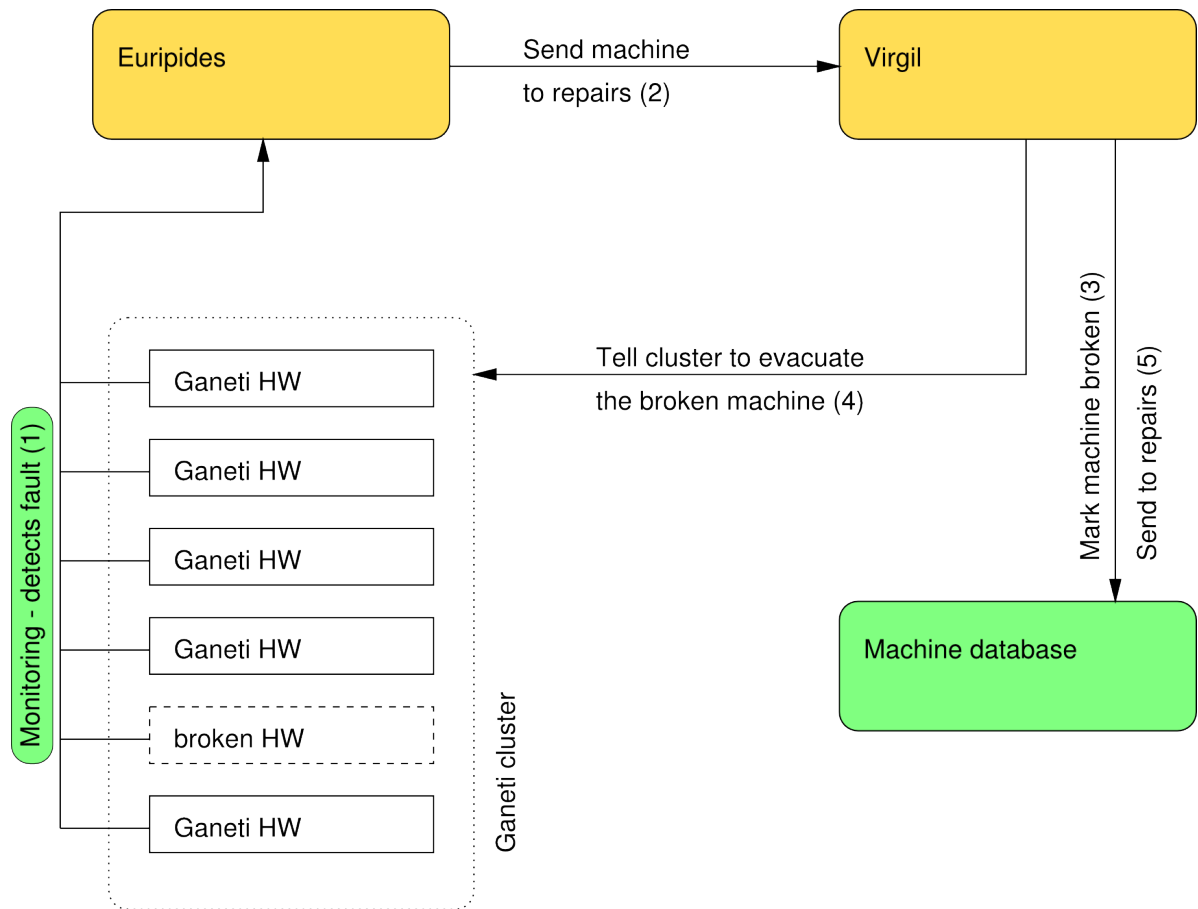
Fleet at Google



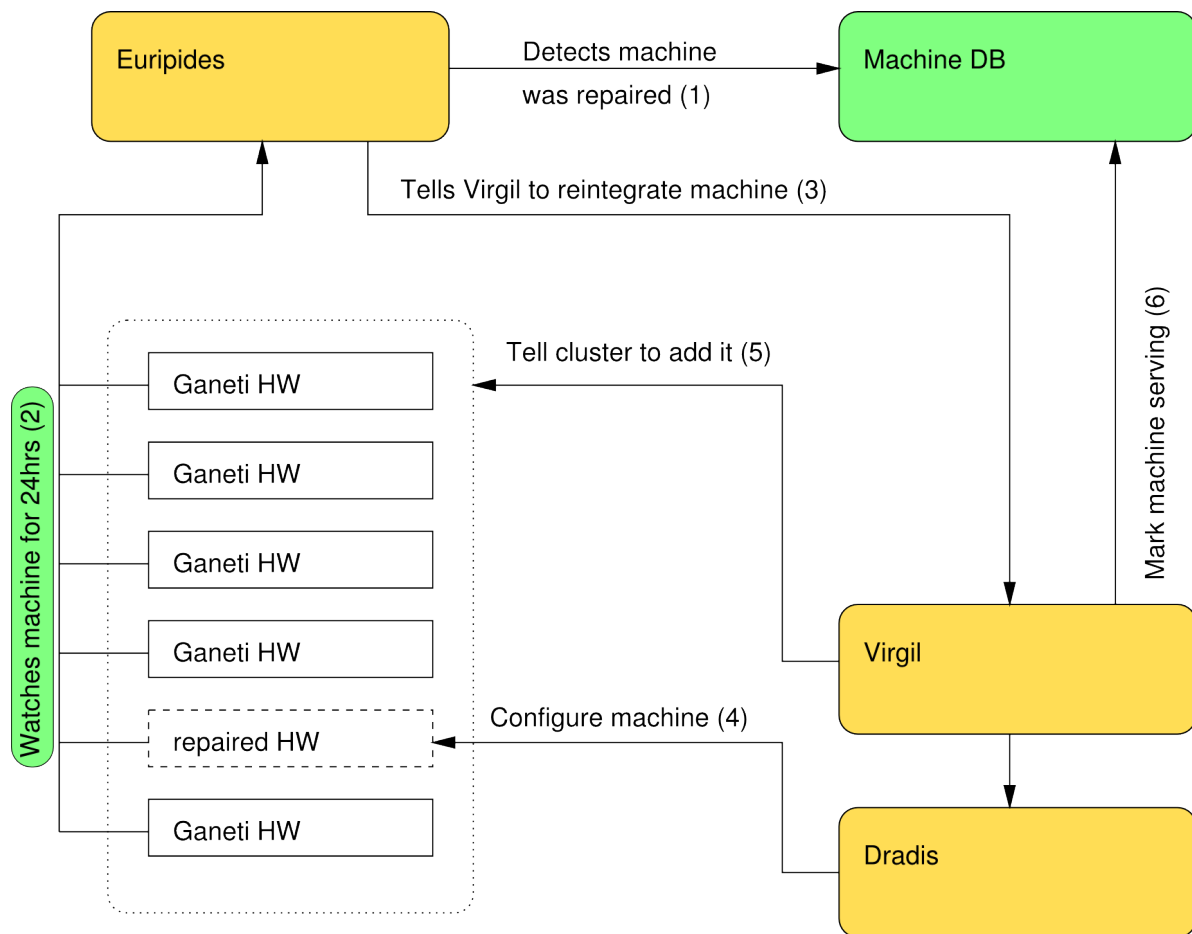
Instance provisioning at Google



Auto node repair at Google



Auto node readd at Google



People running Ganeti

- Google (Corporate Computing Infrastructure)
- grnet.gr (Greek Research & Technology Network)
- osuosl.org (Oregon State University Open Source Lab)
- fsffrance.org (according to docs on their website and trac)
- ...

Conclusion

- Check us out at <http://code.google.com/p/ganeti>.
- Or just search for "Ganeti".
- Try it. Love it. Improve it. Contribute back (CLA required).

Questions? Feedback? Ideas? Flames?

© 2010-2011 Google

Use under GPLv2+ or CC-by-SA

Some images borrowed/modified from Lance Albertson and Justin Pop

