

# Beaver: Practical Partial Snapshots for Distributed Cloud Services

## Summary

This team of students developed a piece of software called Beaver for taking partial snapshots of systems in a cloud computing setting. This is a classic problem but they add to the field an algorithm that takes into account external traffic that is not part of the system taking the snapshot. Beaver does not use large amounts of network bandwidth. Beaver is not always successful but when it is successful it is always correct. This seems like an important point from the paper: “A key idea in Beaver is that, even among the incoming traffic to the in-group, only a subset of that traffic is causally relevant.”

A high-level overview of the algorithm is this: Beaver sets up a “Monolithic gateway overlay” to monitor internal and external before the data is processed. Each machine has a virtual and a direct IP address. The system uses software load balancers to map between the two IPs. The Beaver protocol lives in the load balancers which is why Beaver does not add any network traffic to the system because all of the traffic would be routed through the load balancer.

They cleverly define a partial snapshot. In the classic snapshots, it assumes that the machines in the snapshots are the only machines in the network. This assumption is usually not true because the internet is very large. This work removes that assumption making the thing that used to be full snapshot partial snapshots.

The monolithic gateway overlay has only two jobs tagging incoming packets with snapshot tags and initiating snapshotting.

## Pros

- I appreciated they go into detail about how the snapshots system can be used. Integration testing, deadlock detection, garbage collection, and in-flight message detection. (I am surprised that they do not include restoring state in the use cases)

## Cons

- I thought that starting with an overview of how the algorithm works would make the paper more interesting and easier to follow.

## Further Development

It would be able to add more things to the load balancer. That seems to be the main design paradigm and a very good one. Perhaps we could collect some metadata or other metrics.

## **Other Comments**

How is the data of the snapshot stored? Where is it stored and how is that data used?