Linux Based Disaster Recovery Solutions

James Bottomley

SteelEye Technology

3 April 2006

What Is Disaster Recovery?

- Can mean a multitude of things.
- From the simple weekly backup and the transport of tapes to an off-site storage place, to
- A fully automated instantaneous geographic takeover from a live standby site when a disaster strikes the primary site.
- Disaster Recovery is any system which allows recovery of operation in the face of a systemic, site wide failure.
- This distinguishes it from High Availability which is the recovery from individual points of failure within a site.

Disaster Tolerance Criteria

- There are two key criteria for determining the nature of a Disaster Recovery solution:
  1. How much data are you willing to lose? and
  2. How quickly do you need to become operational.
- If the answers to both these are measured in days, then probably a simple off-site tape rotation backup is sufficient for all your needs.
- If you need more stringent limits, then you probably require the continuous live backups afforded by replication.

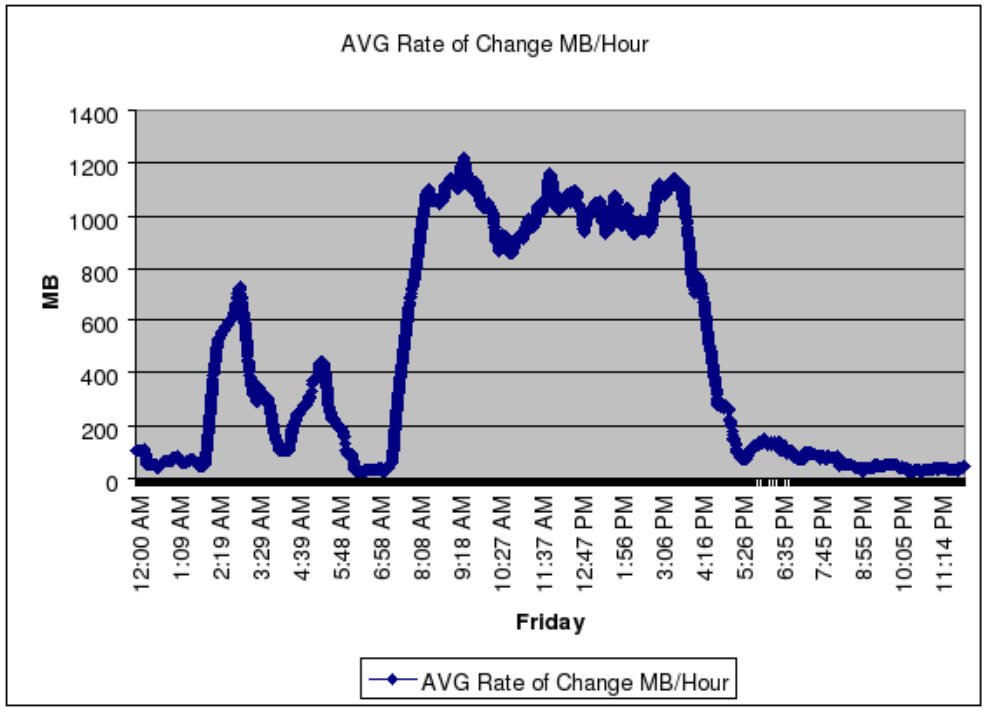
Service Level Agreements

- If your organization performs services for others, it often has an agreement in place about the availability of those services (and the amount of data protection offered).
- Being on the hook for a SLA can drastically impact your need for Disaster Recovery and Continuous Backup.
- To meet performance metrics under a SLA it is critically important to understand how you measure things like "service availability" and "acceptable data loss"

How Big a Network Pipe Do You Need?

- One of the essential prerequisites for establishing your network requirements is knowing your data volume.
- There are two useful figures which characterise this:
  - The Average Bandwidth: This is simply the total data throughput averaged over a long time period (like a week).
  - The Sustained Peak: This takes the same time period, but calculates the data rate on an hourly basis. The sustained peak is the largest of these.
- The network bandwidth should lie between these two figures.

Customer Case Study: Email Server



- Customer was an Educational Institution
- Wanted Disaster Recovery over a T1 line (1.54Mb/s or 693 MB/hour)

Customer Case Study: The reality

- Most *good* replication products can utilise 90% of the line, so the data rate is 623MB/s.
- Therefore, from 8:30am to 5:00pm The rate of data turnover exceeds the possible capacity of the T1 line.
- Real time replication under these circumstances is *impossible*.
  - So either the customer has to turn over less data (compression)
  - or they have to pay for a wider pipe (T3).
  - or they have to settle for non-real time (periodic) replication.

Customer Case Study: Lessons Learned

- When providing disaster recovery, managing customer expectations can be at least as important as perfecting the technology.
- People implementing disaster recovery rarely understand their data turnover.
- The situation can be controlled much more effectively if you run data turnover analysis assessments *before* the customer buys the network link.
- Disaster recovery planning has to be first and foremost, not just an afterthought.

Intent Logging

- An intent log is simply a record of blocks that differ between the primary and replica (the name comes from Intention to Write log).
- since data is not recorded in the intent log, the intention to write *and* the actual data must be committed on the primary before the write can be acknowledged (two I/O operations).
- The intent log can be a simple bitmap (one bit per block), so it's very small and a fixed size (can never overflow).
- When the mirror is broken and restored, only the changed data needs be transmitted, however many times it has changed in the interim.

Intent Logging II

- However, during a replay from the intent log, data is sent to the replica out of order, so the replica is *unusable* until the replay is complete.
- Usually during normal application operation "hot spots" develop. If the log clearing daemon is careful about ageing sectors, most hot spots wind up having their bits already set in the log. Thus no need to do a log write.
- For a prolonged outage, the amount of data transferred for a resynchronisation is usually much less for an intent log than for a transaction log.
- Intent logged systems require a *separate* cache for data acknowledged but not yet committed to the replica.

Linux Replication Solutions

- Solutions exist both in Open Source and Closed Source.
- Both Open Source solutions are intent log based.
- Veritas rumoured to have SRL replication technology available but not released on Linux which is transaction log based.
- Replication technology is invasive to the operating system, so principal disadvantage with closed source solutions is getting timely kernels that match the distributor;
- secondary problem is that full kernel replacement often invalidates distribution vendor's support agreements.

MD/NBD

- Work for 2.6 is being sponsored by SteelEye Technology, Inc.
- Based on existing in-kernel md (Redundant Array) and nbd (network block device).
- proposed solution involves adding a non-volatile intent log and asynchronous capability to the existing in-kernel md driver.
- patches to implement this capability are being reviewed on the kernel mailing lists for inclusion into 2.6.
- Forms the basis of our LifeKeeper Disaster Recovery Solution.

DRBD

- Project of Philippe Rensner, currently being worked on by SUSE.
- Adds a completely new driver to the kernel
- system is a simple mirror with a volatile intent log and asynchronous capability.
- Project is not currently on inclusion track for 2.6

Conclusions

- Disaster Recovery using Replication is a viable solution today.
- It can be implemented today using completely open sourced components.
- Subtleties in the implementation often leads to service oriented offerings to assist implementing organisations.