Lsyncd (LIVE SYNCING DAEMON)

Host 2 Host Replication Made Easy

Lsyncd **(Live Syncing Daemon) is a light-weight data replication tool which is free, easy to install and convenient to configure. It is an open source technology based, ultimate 'host to host replication/ mirroring' tool with very minimal configuration, cost effective and can assure the RPO in minutes or even seconds.**

K GANAPATHI
Technical Director
kganapathi@nic.in

MADAN PRABHU D
Scientist -B
madan.prabhu@nic.in

Edited by
R. GAYATRI

Business Continuity Plan (BCP) and Disaster Recovery Plan (DRP) are the essential elements of every ICT application. BCP/ DRP requires a redundant arrangement for the IT hardware infrastructure, software and most importantly for the data getting generated through the application.

The method, tools, storage infrastructure required for maintaining the redundancy of data are to be decided based on numerous factors like frequency at which the data is getting changed, the type and size of the data, availability of facility at the redundant location (floor space, fund, similar infrastructure) availability of skilled manpower etc.

DATA REPLICATION LEVEL

As far as BCP/ DRP with regard to data is concerned, following are the two extremes of maintaining the data backup/ replication:

- The least that can be done is replicating the data using traditional backup tools. This is cost effective and in this method, storage source and destination can be heterogeneous. But the difference between point of disruption and point of recovery of data may be large in hours/ minutes – ie. the Recovery Point Objective (RPO) is higher.

- The best that can be done is maintaining a mirror site at the remote, which requires identical storage and other infrastructure both in production and remote. In this setup, the data is getting replicated at block/ byte level between storages (Storage to Storage replication). This demands more cost, more space, more skilled man power, but the RPO will be in seconds/ milliseconds.

Due to the above factors, not every IT project/ department can afford to maintain mirrored sites. Thus, ***Lsyncd*** is a data replication tool which is cost effective and also can assure the RPO in minutes or even seconds, which is suitable for Host to Host replication. ***Lsyncd*** is mainly for host to host replication of flat files stored out of the Database system.

HOST TO HOST REPLICATION WITH 'LSYNCD'

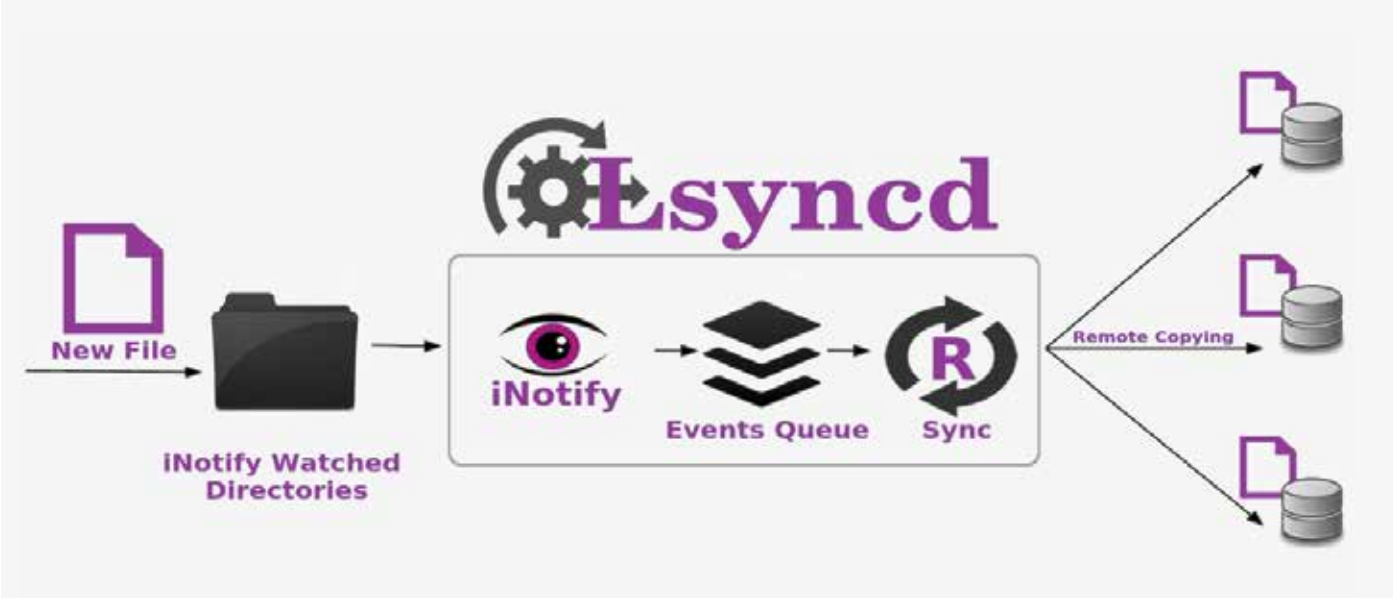
PLAYING FIELD OF LSYNCD

ICT application can generate and store the data in a database or in the OS as flat files. In the first case, the synchronization / replication of primary data can be done by underlying DBMS's replication service. But, in the case of flat files, the synchronization will be taken care of either by the Operating System's inbuilt replications tools or by any other third party tools – here comes a simple and handy solution for host to host replication '***Lsyncd***':

RSYNC VS LSYNCD

Traditional Linux server administrators might be using Rsync, the native tool available in all the Linux OS, as the replication tool for long and raise a question or doubt on using ***Lsyncd***.

Since 1996, '***Rsync***' holds the throne of the synchronization realm without any defenders. The merely delta encoding algorithm checks for the difference of file content from source to destination and transmits the difference data/deltas through 'Rsync' daemon via TCP or '***Rsync***' via SSH. Therefore, using the cron based '***Rsync***' synchronization always drags you into the hell of troubles in replication. The real problem of ***Rsync*** starts with cron based synchronization of numerous files scattered across various folders. The built-up time of checksum generation and the file attributes compari-



Lsyncd architecture

son for finding which set of partial files that should be transmitted grows exponentially with the number of files & folders increases. For example, if we have more than 2 TB of flat files it may take around 1-2 hours of time only to creates the list of modified files and when real transmission starts the data gap from the source to the destination will be already in 2 hours. Day by day, this gaps extends and finally the synchronization gap will be in GBs.. In order to minimize this built-up time, we need to introduce some file system event monitoring to fire the Rsync commands whenever any file changes or newly created. Inotify is a file change notification system in Linux Kernel and when we pair it up with the Rsync, we can spawn Rsync only for file creation/ change events. This is where ***lsyncd*** really shines and runs as a service on the OS level with logging facilities.

LSYNCD

As described above, ***Lsyncd*** is nothing but a tool written in ***lua*** language using linux ***inotify*** and ***Rsync*** package. i.e., ***Lsyncd = inotify + Rsync***.

The Live Syncing Daemon watches local directory tree events through inotify or fs events interface. It queues up these events in a queue and every x seconds it will execute these events and copy/create the files in another directory either locally or remotely. The tool itself makes use of ***Rsync*** as well as ssh for remote transfers, this allows you to make sure that no data is lost due to the checksums that ***Rsync*** looks at, as well as when using ***SSH*** you ensure that all data is encrypted. The beauty of ***Rsync+SSH*** is an advanced action configuration that uses a ***SSH*** to act file and directory moves directly on the target instead of re-transmitting the move destination over the wire.

Fine-grained customization can be achieved through the config file. Custom action configs can even be written from scratch in cascading layers ranging from shell scripts to code written in the ***Lua*** language for automating any activities over the replicated files. For example, if it is required to archive the uploaded files to another format; it can be automated through custom-action configs. In this way simple, powerful and flexible configurations can be achieved. The ***Lsyncd*** architecture is as given in the graphic representation above.

CASE STUDY

A sample instance was taken for case study, which contained a total of over 45.34 lakh files uploaded and still growing. With Rysnc as replication tool, the Point of recovery was in the range of 24 to 48 hours data gap. With ***Lsyncd*** as the replication tool, the files were getting replicated to the remote / DR node almost instantaneously as and when they are created in the primary node, resulting the RPO of 3 seconds data gap.

No of Files:	45.34 Lakhs
Rsync RPO:	24 to 48 Hrs data gap
Lsyncd RPO:	3 Sec data gap

CHALLENGES

The major issue of ***Lsyncd*** arises in the event of primary/ remote host failure or termination of ***Lsyncd***.

If the primary host fails, then ***Lsyncd*** generates the '***iNotify Watch Directories***' again from the scratch as when the primary host is made up. In a case study instance, ***Lsyncd*** took 4 hours to build the watch list and kick start the replication.

If the remote host fails, then ***Lsyncd*** keeps on trying to ***Rsync*** to remote, event queue keeps building up and ***Lsyncd*** daemon stops after some time. So, this requires the restart of ***Lsyncd*** service as and when the remote host is made up i.e. again it has to re-generate the '***iNotify Watch Directories***' to ensure that the process builds up the file directory.

For further information, please contact:

K. GANAPATHI
Technical Director
NIC State Unit, Chennai, TAMIL NADU

Email: kganapathi@nic.in
Phone: 044-28253958