

How to configure maximum performance storage space for Debian GNU / Linux on IBM DS 8300 Data Storage Systems

Author: André Felipe Machado<andremachado@techforce.com.br>

The IBM DS 8300 Data Storage Systems are multi millions dollars flexible high availability and performance SAN machines.

But you may left much of such performance and availability behind if you do not configure them correctly for Debian GNU / Linux.

See how to ask for performance data storage space on them. Or what you need to configure on them.

Read about an actual configuration [running with Debian GNU / Linux hosts at SERPRO](#)

Essential concepts about IBM Data Storage performance

The IBM DS architecture was born for mainframe use. So you need to understand some different concepts / naming from the Debian GNU / Linux regular ones.

The IBM DS 8300 hardware and software (it is a multi cpu Power AIX machine) is **optimized for highly parallel I/O requests.**

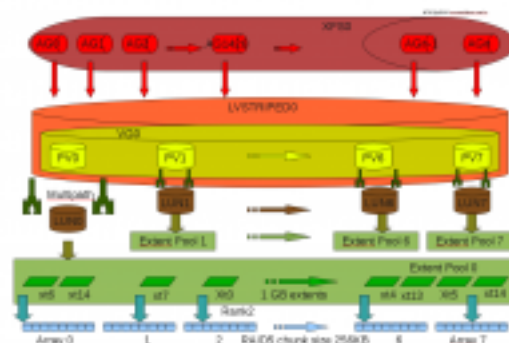
The IBM DS 8300 hardware could be "logically partitioned" (LPAR) in order to guarantee quality service levels (space, latency, transfer speed) and LPARs could be grouped into Storage Facility Images (SFI) for more availability. It is like a QoS, reserving disks, interfaces, cpus, ram for some hosts.

The storage abstraction layers concept

The IBM DS 8300 has many **abstraction layers**, progressively **grouping** the **inferior layers** into **higher abstraction layers**, *transparently* (for the Debian GNU/Linux hosts) managing availability and performance of each layer to the higher one.

Disk Drive Modules (DDM: FC or SATA) -> Array Sites (identical DDM) -> Arrays (array sites forming "RAID" 5 or 6) -> Ranks -> 1 GB Extents -> Extent Pool -> LUN -> Virtual Logical Sub System (LSS) -> Volume Groups -> SVC (a storage management sw).

The high performance storage, multipath, lvm and filesystem tuning overview



Performance Connectivity

Each high transfer and rotation speed Fiber Channel DDM should be connected by FC interfaces instead of slower SATA DDM to compose Array Sites.

Each rank could be connected to up to 8 Fiber Channel Device Adapters (DA).

Each host computer could be connected to the SAN device through up to 8 Fiber Channel Host Bus Adapters (HBA).

If, among other ideal settings, 8 DA are connected to a server using 8 HBA, a maximum theoretical sustained write speed of 1800 MB/s could be achieved, *given that the host multipath software is able to aggregate bandwidth instead of the regular failover behaviour.*

Partial predefined configurations example

Some companies, may prefer to have some predefined configurations, leaving less ones to the sysadmin configure when requesting storage space.

~~Below is an example of a configuration. Please, notice that is **almost** the highest performance configuration, given the connection constraints.~~

- 8 DDM compose 1 array site (already maximum).
- 1 array site forming 1 Array (RAID 5) (already maximum).
- 1 Array compose 1 Rank (already maximum).
- 1 Rank has many 1 GB extents.
- 8 ranks form 1 Extent Pool (already maximum).
- 1 Extent Pool compose 1 or more LUN.
- Computer hosts connect at LUN abstraction level.
- 1 Rank is connected through one Fiber Channel Device Adapter.
- Each computer host may have up to 2 FC HBAs.
- There is only 2 LPARs. One for all GNU / Linux machines and other for the mainframes.
- Sequential Prefetching Adaptative Replacement Cache (SARC)
- Adaptative Multistream Prefetching (AMP)

Volume request actual example

[We used this:at our company](#)

- Using 15 kRPM FC DDMs, we need 8 LUNs, each from a different Extent Pool , with extent rotation over the ranks within the Extent Pool, with Storage Pool Striping across multiple ranks inside the Extent Pool.

Maximum performance configuration hints

Beyond the already cited already maximum configurations at the actual example, you may request the as many you could of the following configurations.

- Extent rotation over the ranks within the Extent Pool.
- Storage Pool Striping across multiple ranks inside the Extent Pool.
- 15 kRPM FC DDM to form the array sites.
- 8 FC DAs for each Rank connection, if allowed.
- 8 FC HBA for each "client" connection, if allowed.
- Each LUN from different Extent Pool, if allowed.
- Variable LPAR dimensioned for your "client" application needs, if allowed.
- Connection at the Volume Group abstraction layer (on the LSS), if allowed.
- Join 2 LPARs into one Storage Facility Image (SFI), if allowed.
- 4 LUNs at one SFI (at least LPAR) and 4 LUNs at other SFI (at least LPAR), if allowed.

These configurations hints are related to the "client" host:

- Request
8 LUNs to form your
host based
LVM striped volume. The correct stripe width calculation is for a future article.
- Configure your host
multipath for
load balancing over the HBA and
NEVER mount LUNs directly.

Each configuration may have severe impact on costs, notably the LPAR, SFI related ones and the number of FC used, as they preclude other host and apps to use allocated resources.

As of GNU / Linux 2.6.26 kernel version, multipath is still not able to aggregate bandwidth on HBA, only implementing round robin load balancing. So you still would not greatly benefit from having more than 2 HBAs.

Actual performance numbers for comparison

Using the already cited actual configurations, with Extent rotation over the ranks within the Extent Pool, with Storage Pool Striping across multiple ranks inside the Extent Pool, with 15 kRPM FC DDMs to form the array sites, and configuring Debian GNU / Linux 5.x Lenny hosts multipath for round robin load balancing and failover, configuring 8 LUNs to form a LVM striped logical volume,

using ext3 tuned for big files, we got around 883 MB/s read and 361 MB/s write for 10 GB files.

But for heavy load hosts, the key is PARALLEL I/O, and then SAN really shines when configured and filesystem chosen and tuned.

Keep in mind that when tuned for SMALL files, the write performance for 10 GB files suffers a significant drop. We got around 175 MB/s write speed for ext3 tuned for SMALL files. At these conditions, common speed tests are almost meaningless. Better benchmarking options are ~~both the ext3 and XFS~~ ~~that~~ ~~are~~ ~~not~~ ~~the~~ ~~best~~ ~~for~~ ~~parallel~~ ~~I/O~~. ~~The~~ ~~reason~~ ~~is~~ ~~that~~ ~~XFS~~ ~~is~~ ~~a~~ ~~much~~ ~~better~~ ~~option~~ ~~for~~ ~~parallel~~ ~~I/O~~, but undocumented "secret" ext3 tuning tips (read the source Luke) could **almost** even the numbers. Into a future article we will address these Debian *heavy* parallel load LVM and filesystems "secret" tuning tips.

Resources:

[IBM System Storage DS8000: Architecture and Implementation.](#)

[IBM System Storage DS8000 Information Center.](#)

[DS8000 Performance Monitoring and Tuning.](#)