LightStor ™

An Architecture and Design Overview

Copyright

Contributors

G S Madhusudan

Matias Bjørling

Philippe Bonnet

IIT-Madras

Acknowledgements

[Optional section.]

Feedback

Please direct any comments or suggestions about this document to: madhu@lightstor.org

Publication date and software version

Published

Contents

Chapter 1
Introduction

Introduction to LightStor

LightStor aims to build a comprehensive Storage and Backup system with unlimited size and bandwidth scalability using a RapidIO fabric. While other routable fabrics like Infiniband or quasi-fabrics like PCIe can be used, LightStor benefits are best demonstrated when RapidIO is used.

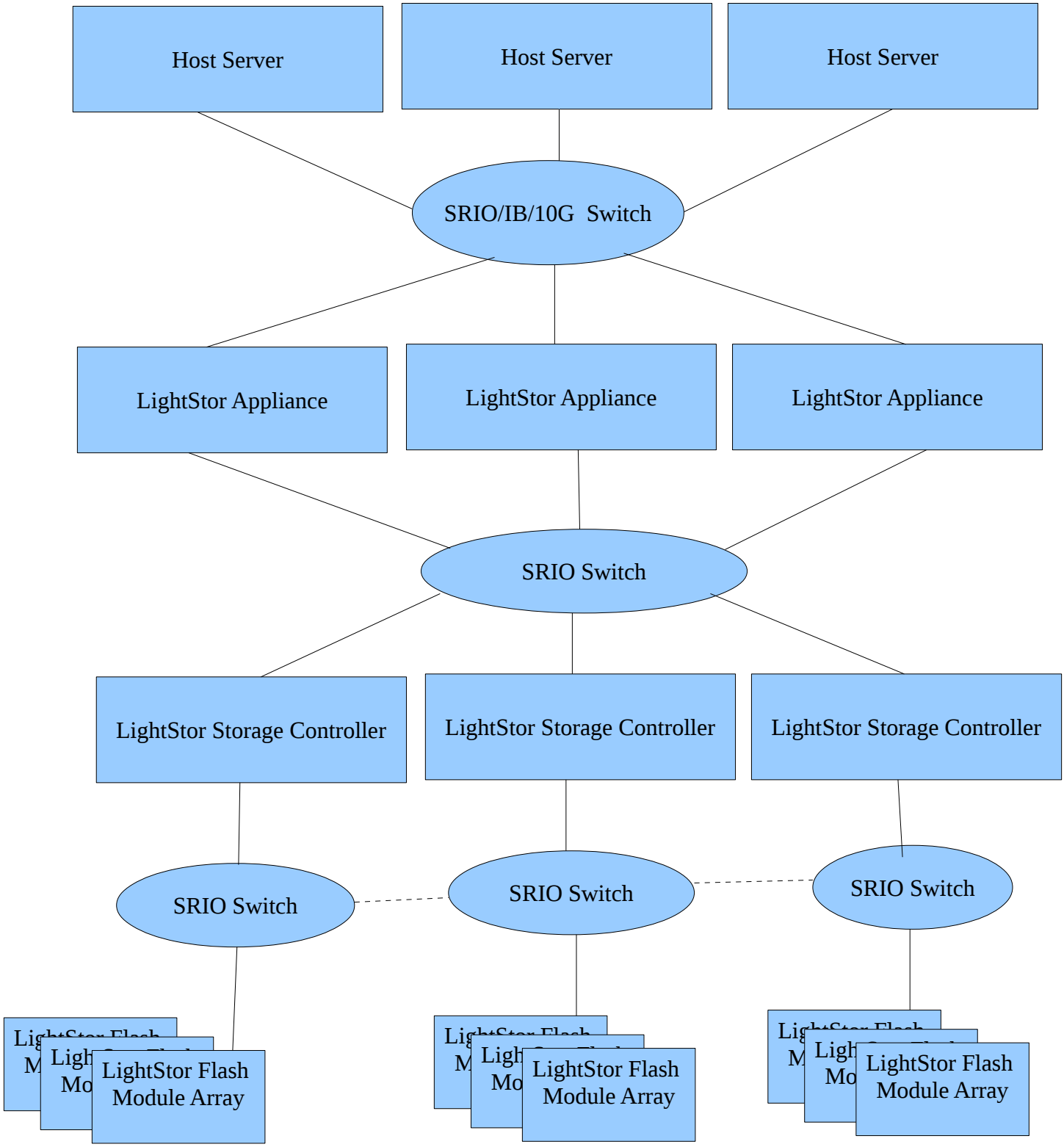
All components of LightStor the source for the controller HW written in the Bluespec Language, device side stack, host side stack, Appliance VM stack and the design files for the PCBs are in open source allowing anyone to build an end to end exabyte class storage system with Optical interconnects using completely open sourced components. The system is competitive in terms of performance and features with any proprietary system.

Phase 1 of the project will focus only on SSD/BVRAM based storage and support for HDDs and storage tiering will be added in Phase 2.

LightStor is built using three components and a SRIO fabric

- Storage Controller
  - Controller HW - Board design + HDL source (Bluespec)
  - Open standard NAND flash modules (Board design)
  - Lightweight Linux with container based sandbox for User defined Functions
  - LightNVM compatible device side driver
- NAND/NVM modules
  - open NAND/NVRAM module design
  - device controller (low level NVM functions + ONFI ←--> SRIO bridge)
- Appliance Engine
  - Commodity HW with SRIO HBA
  - LightNVM based host side stack
  - Container/VM based appliance support

A typical topology is show below


Host Server
Host Server
Host Server
SRIO/IB/10G Switch
LightStor Appliance
LightStor Appliance
LightStor Appliance
SRIO Switch
LightStor Storage Controller
LightStor Storage Controller
LightStor Storage Controller
SRIO Switch
SRIO Switch
SRIO Switch
LightStor Flash Module Array
LightStor Flash Module Array
LightStor Flash Module Array

Note:

- The dotted lines represent optional redundant paths from alternate storage controllers to Flash module arrays via SRIO switch (not shown). This is required if controller level redundancy is required
- Flash modules plug into the storage controller card

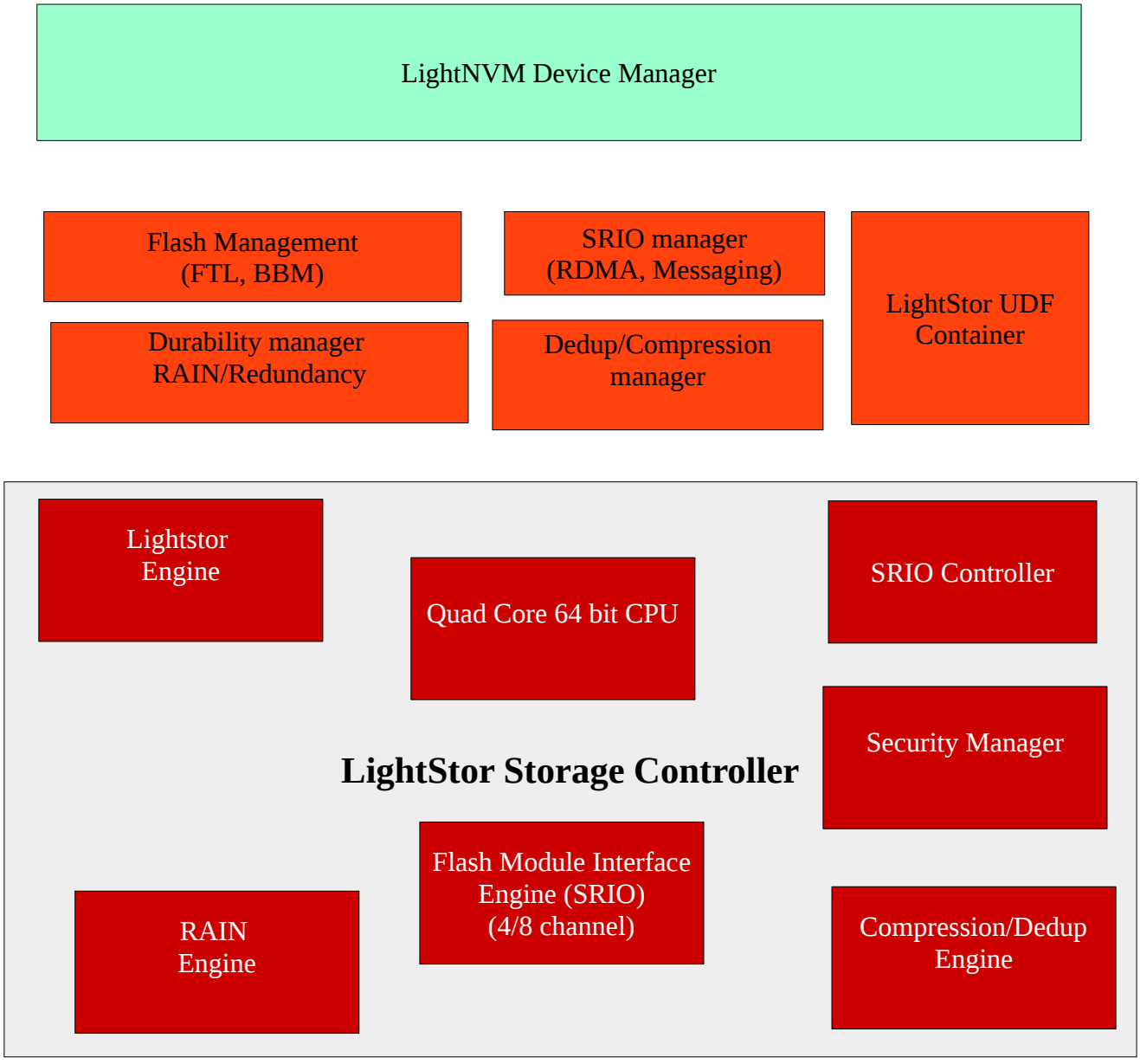
LightNVM API

The LightNVM API is the main storage API (conceptually equivalent to the NVMe API). But unlike typical SSD APIs, it moves certain functionality to the host. The functionality split between the host and the SSD controller is show below

	Host	Controller	Notes
Logical to Physical Addr. Translation	x		Mapping table are kept within host to easily make decisions about data placement. Data are buffered to internal IO buffers for IO performance.
Durability Management		x	Maintains a compressed mapping table, built on the requests submitted from the host. Used for durability and fast startup
Garbage Collection	x		
Wear leveling	x	x	Disk notifies about any wear-leveling required updates.
I/O Buffering		x	One buffer per channel
Bad Block Management	x	x	(i) host are notified about bad blocks or (ii) disk exposes a virtual layer, with bad blocks abstracted away.
Atomic Transactions	x		
Key-value I/O	x		Translation are either bypassed and used like ODF (Baido open-channel SSD), or some logic are implemented within the host to handle sub-block keys.

Storage Controller

The storage controller is the first ever open source SSD controller. The functional architecture of the storage controller system is shown below.



HW Configuration and PCB Format

Full length PCIe card or SHB card format are suitable standard formats. Size is dictated primarily by the flash module count. See next section. Typical configuation will be a 4.x U box with 10-20 cards plugging into a carrier board with SRIO switches. External SRIO parts will depend on bandwidth required. With Gen 3.x SRIO, a four lane port will give 40Gbit/sec. 1, 2 and 4 lane configurations are preferred (for a variety of reasons).

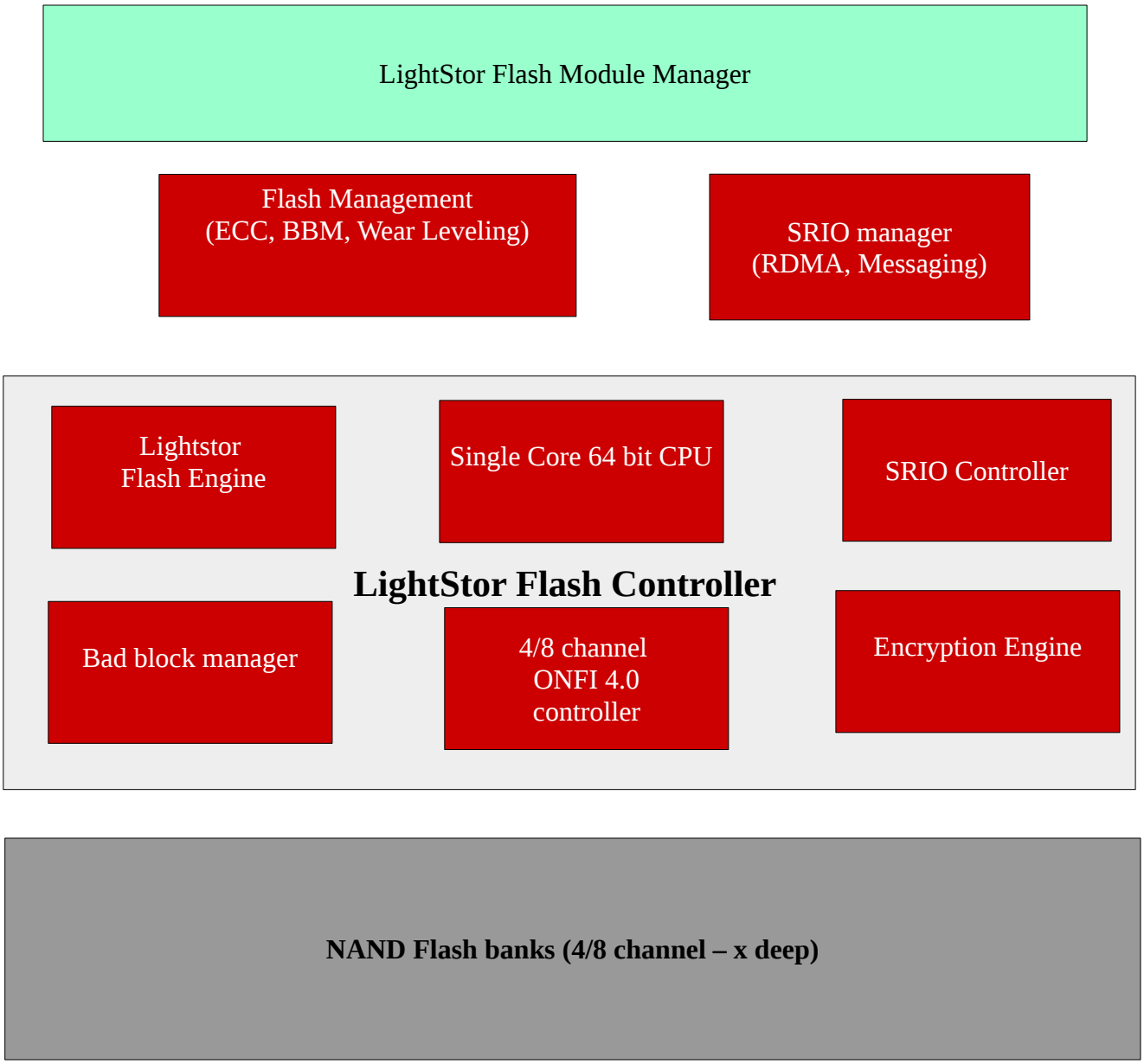
SoC Architecture

The SoC architecture of the controller is show below

<add SoC pic>

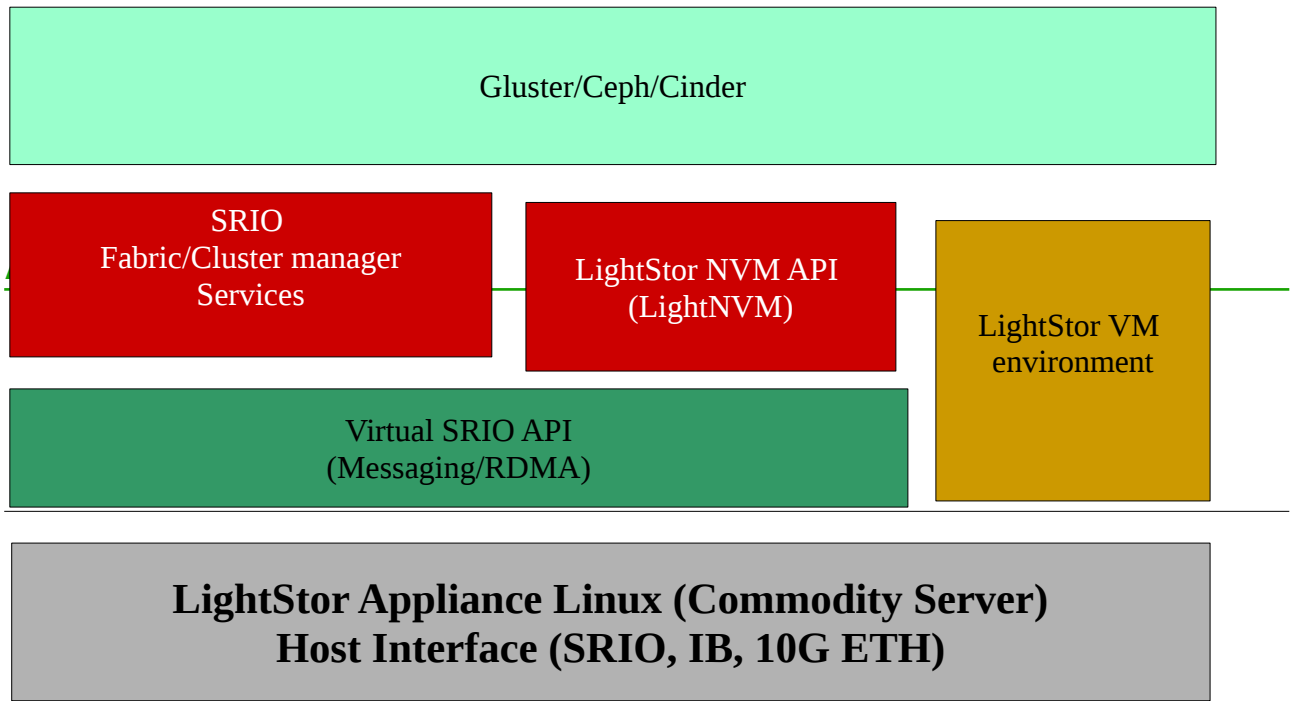
The CPU is used primarily for control path and to create/update lookup tables/maps and handling exceptions. The primary data-path is **SRIO ↔ LightStor Engine ↔ Flash Module Engine.** The RAIN engine and the Compression/Dedup engine are looped into the data path as needed. The Security manager handles all the high level security functions and programs the encryption engine in the flash module. The actual encryption is done in the flash module controller.

Flash Module



PCB Format

The M2 22mm wide format is one option. A 22x80 size will allow 12 modules in a PCIe full length/full height card with enough space for the storage controller and memory. Analysis needed here. But module length can be varied to strike a trade-off between more channels (hence more performance) vs higher capacity.


Gluster/Ceph/Cinder
SRIO
Fabric/Cluster manager
Services
LightStor NVM API
(LightNVM)
LightStor VM
environment
Virtual SRIO API
(Messaging/RDMA)
LightStor Appliance Linux (Commodity Server)
Host Interface (SRIO, IB, 10G ETH)

Index