OPENFABRICS ALLIANCE

10TH ANNUAL INTERNATIONAL DEVELOPER WORKSHOP

Update on Scalable SA Project

Hal Rosenstock
Mellanox Technologies

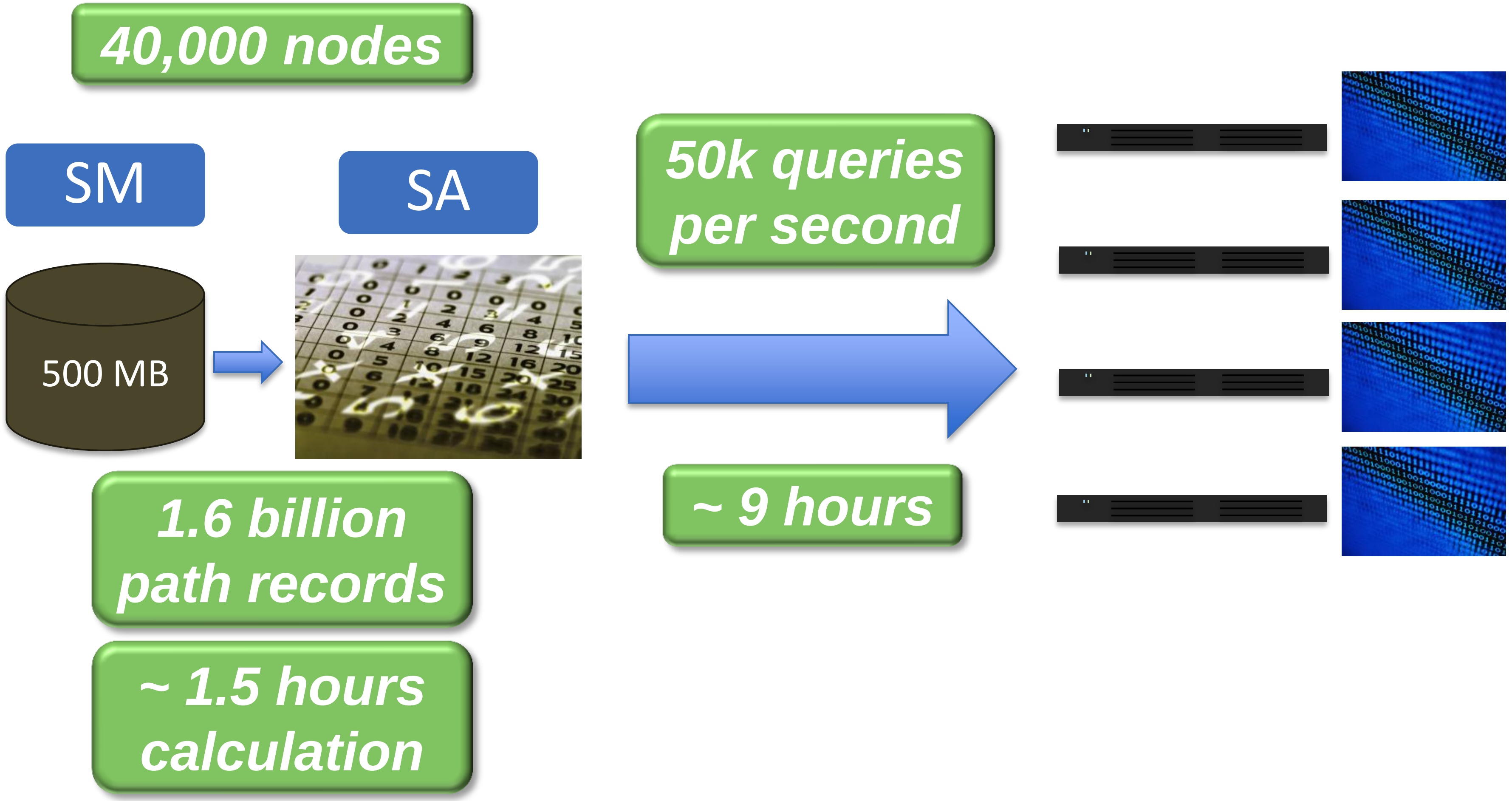
#OFADevWorkshop

The Problem And The Solution

n^2 SA load

- SA queried for every connection
- Communication between all nodes creates an n^2 load on the SA
  - In InfiniBand architecture (IBA), SA is a centralized entity
- Other n^2 scalability issues
  - – Name to address (DNS)
    - Mainly solved by a hosts file
  - – IP address translation
    - Relies on ARPs
- Solution: Scalable SA (SSA)
  - – Turns a centralized problem into a distributed one

Analysis



40,000 nodes

SM

SA

500 MB

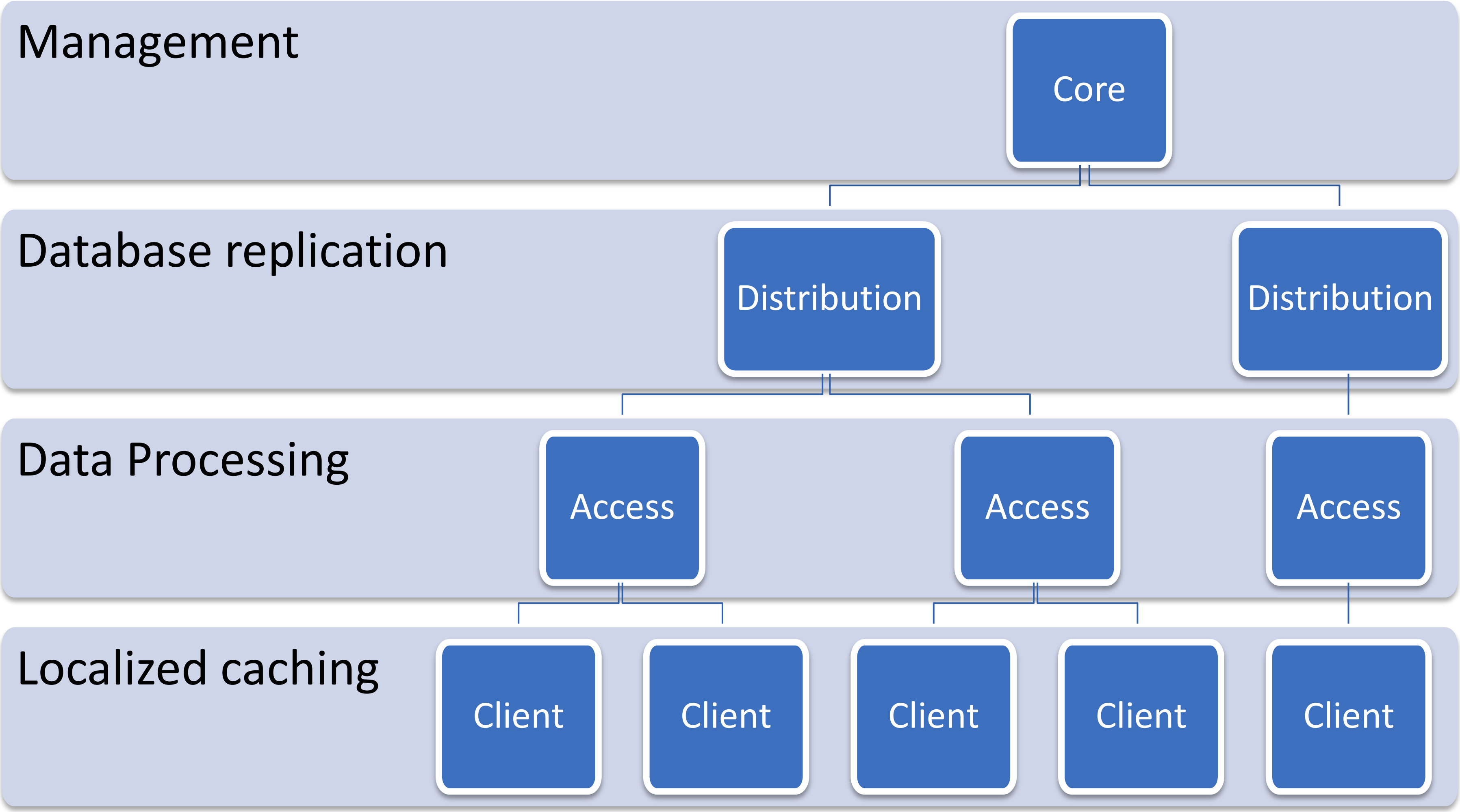
1.6 billion path records

~ 1.5 hours calculation

50k queries per second

~ 9 hours

SSA Architecture



Distribution Tree

- Built with rsockets AF_IB support
- Parent selected based on “nearness” based on hops as well as balancing based on fanouts

rsockets AF_IB rsend/rrecv performance

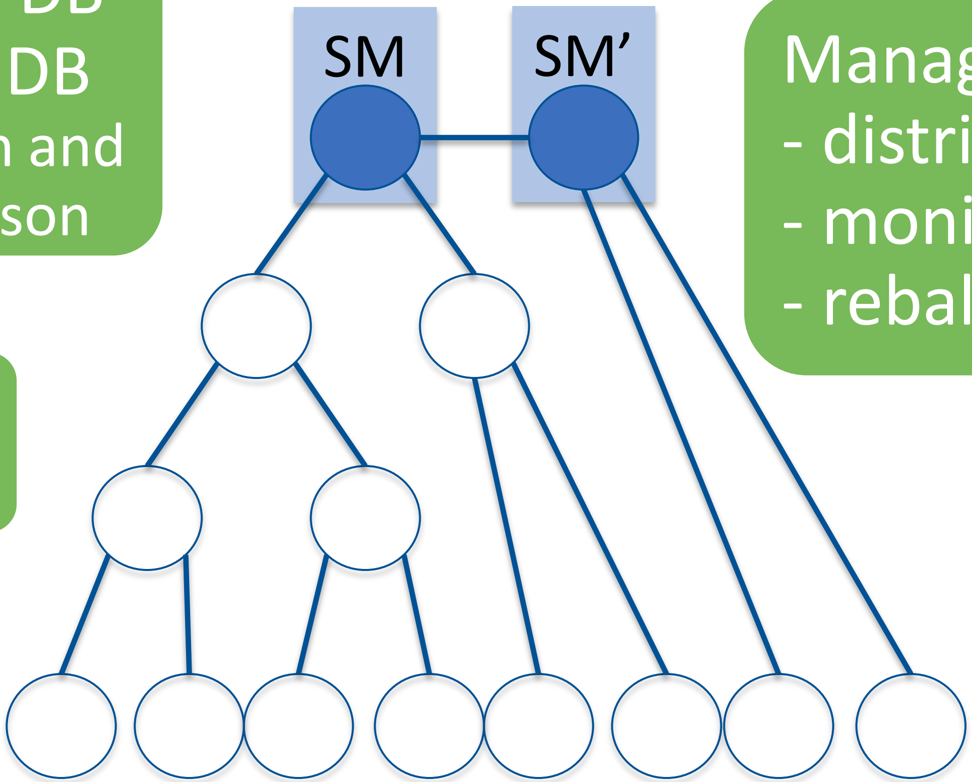
- On “luna” class machines as sender and receiver with 4x QDR links and 1 intervening switch
  - – 8 core Intel(R) Xeon(R) CPU E5405 @ 2.00GHz
- Default rsocket tuning parameters
- No CPU utilization measurements yet
- SMDB: ~0.5 GB (for 40K nodes)

Data Transfer Size in Bytes	Elapsed Time
0.5 GB	0.669 seconds
1.0 GB	1.342 seconds

Distribution Tree

- Number of management nodes needed is dependent on subnet size and node capability (CPU speed, memory)
  - Combined nodes
- Fanouts in distribution tree for 40K compute nodes
  - 10 distribution per core
  - 20 access per distribution
  - 200 consumer per access

Core Layer



Core found at SM LID

raw SM DB → SSA DB extraction and comparison

Manage SSA group
- distribution control
- monitoring
- rebalancing

Nodes join SSA tree

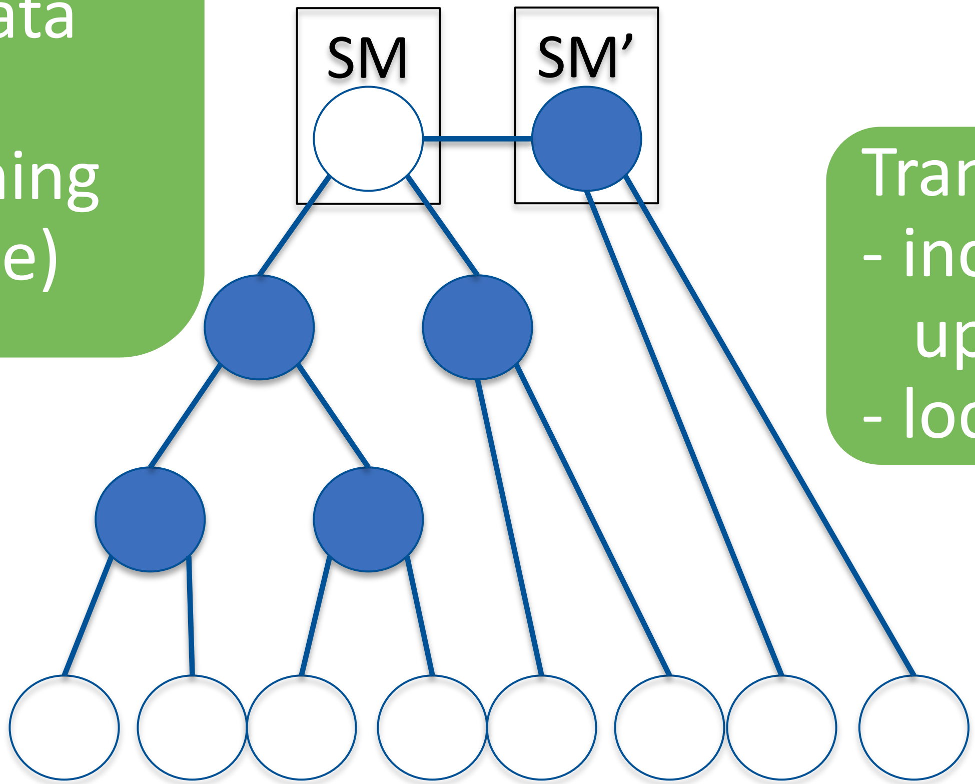
Core Performance

- Initial subnet up for ~20K nodes fabric
  - Extraction: 0.228 sec
  - Comparison: 0.599 sec
- SUBNET UP after no change in fabric
  - Extraction: 0.152 sec
  - Comparison: 0.100 sec
- SUBNET UP after single switch unlink and relink
  - Extraction: 0.190 sec
  - Comparison: 0.865 sec
- Measurements above on Intel(R) Xeon(R) CPU E5335 @ 2.00GHz 8 cores & 16G RAM

Distribution Layer

Distributes SSA DB
- relational data model
- data versioning (epoch value)

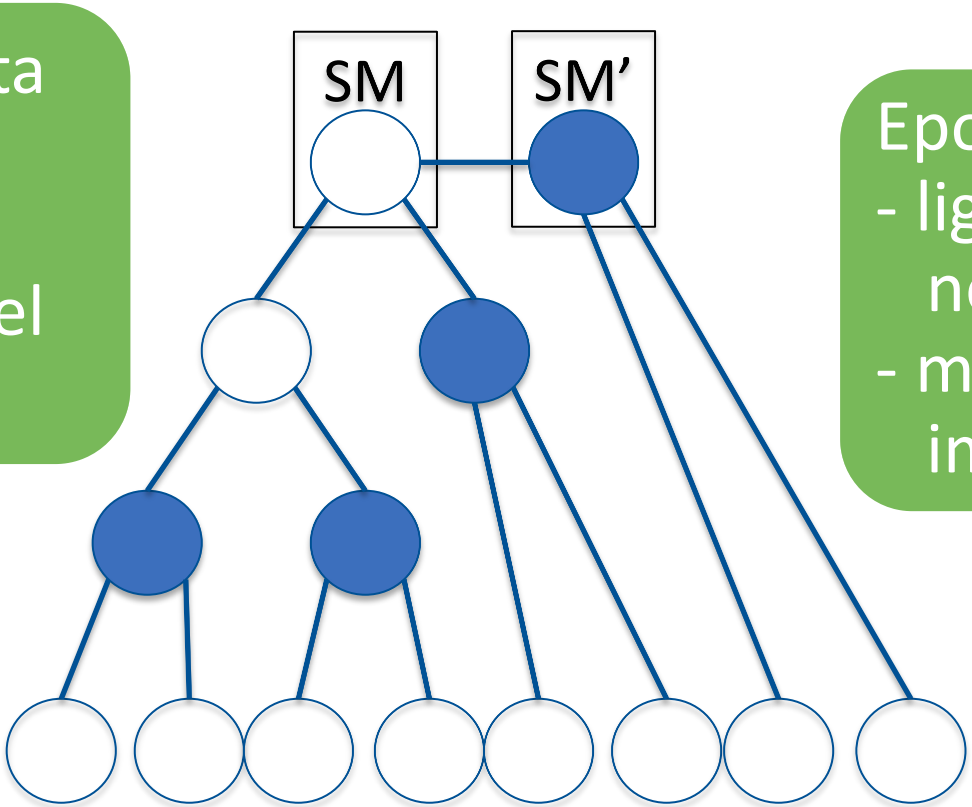
Data agnostic



Transaction log
- incremental updates
- lockless

Access Layer

Data aware



Formats data
- select SA queries
- higher-level queries

Epoch value
- lightweight notification
- minimal job impact

Access Layer Notes

- Calculates SMDB into PRDB on per consumer basis
  - Multicore/CPU computation
- Only updates epoch if PRDB for that consumer has changed

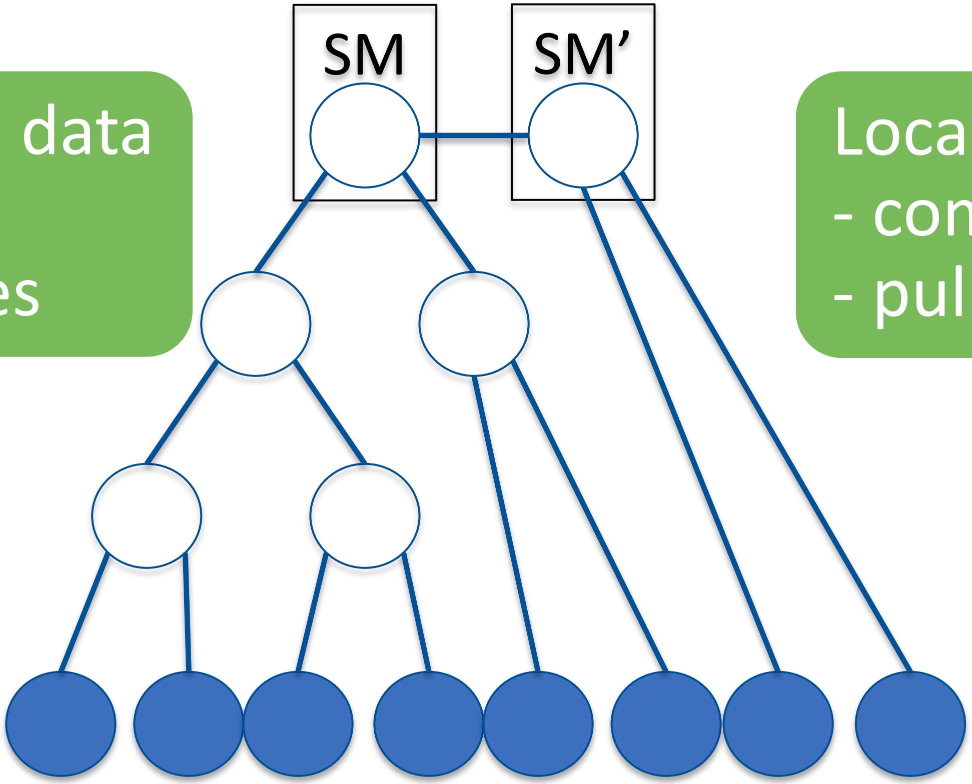
Access Layer Measurements/Future Improvement(s)

- Half world (HW) PR calculations for 10K node simulated subnet
- Using GUID buckets/core approach, parallelizing HW PR calculation works ~16 times faster on 16 core CPU
  - Single threaded takes 8 min 30 sec for all nodes
  - Multi threaded (thread per core) takes 33 seconds
  - Parallelization will be less than linear with CPU cores
- Future Improvement(s)
  - One HW path record per leaf switch used for all the hosts that are attached to the same leaf switch

Compute Nodes (Consumer/ACM)

Integrated with IB ACM
- via librdmacm

Publish local data
- hostname
- IP addresses



Localized cache
- compares epoch
- pull updates

ACM Notes

- ACM pulls PRDB at daemon startup and when application is resolving routes/paths
  - Minimize OS jitter during running job
- ACM is moving to plugin architecture
  - ACM version 1 (multicast backend)
  - SSA backend
- Other ACM improvements being pursued
  - More efficient cache structure
  - Single underlying PathRecord cache ?

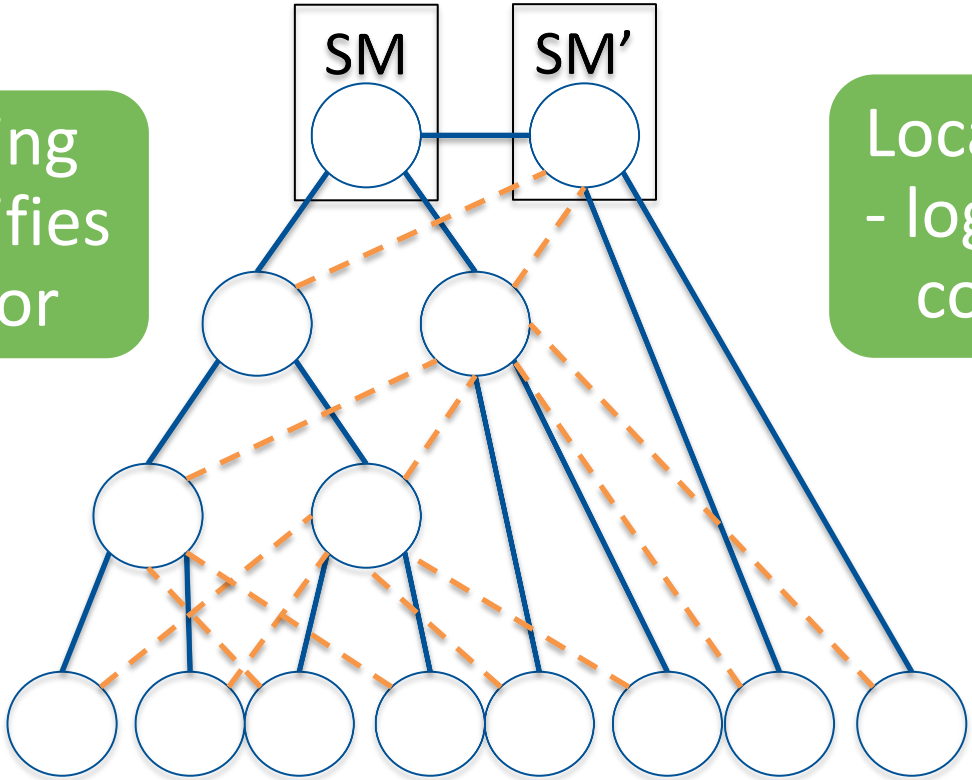
Combined Node/Layer Support

- Core and access
- Distribution and access

Reliability

Primary and backup parents

Error reporting - parent notifies core of error



Local databases - log files for consistency

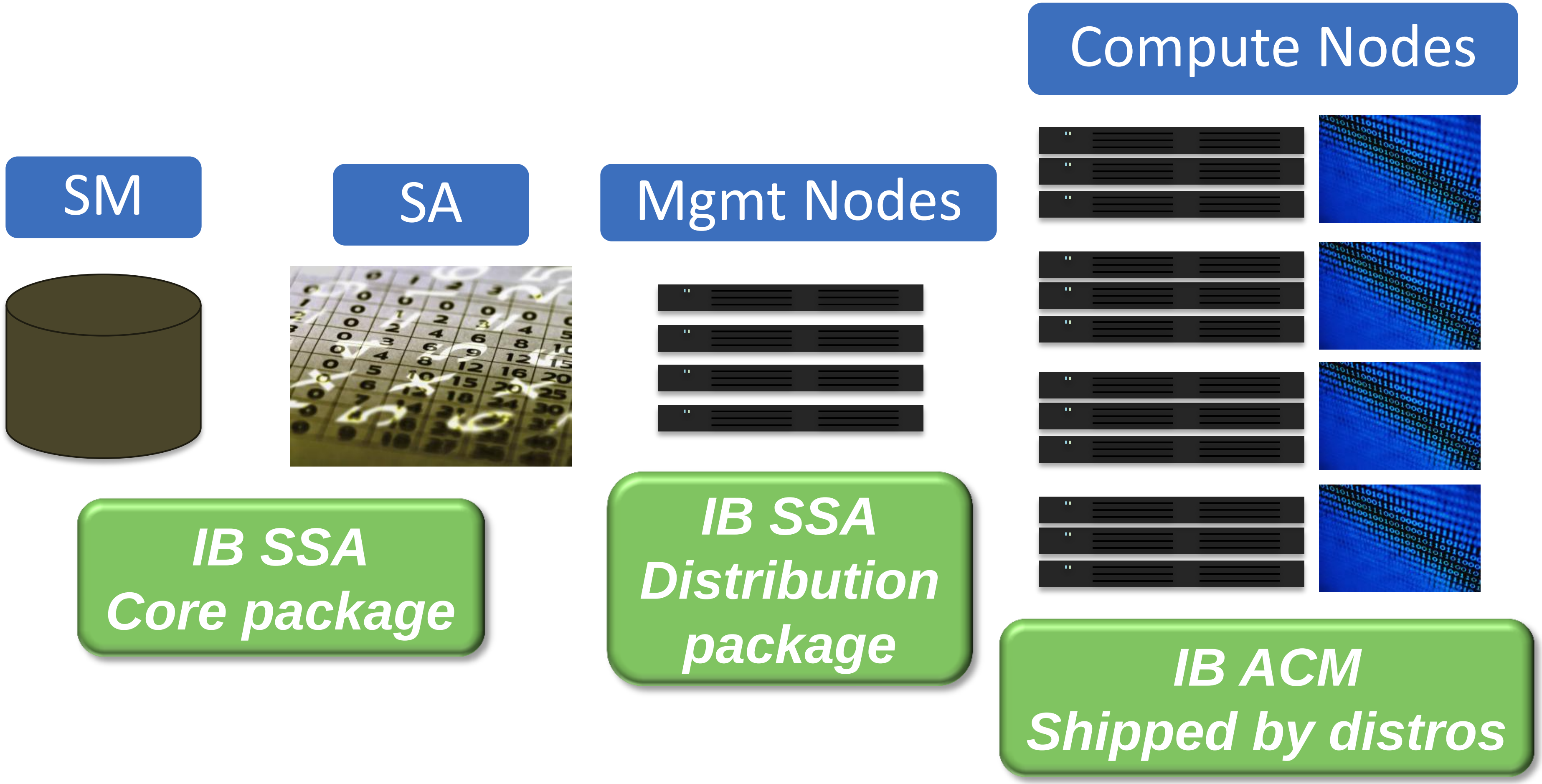
System Requirements

- AF_IB capable kernel
  - 3.11 and beyond
- librdmacm with AF_IB and keepalive support
  - Beyond 1.0.18 release
- libibverbs
- libibumad
  - Beyond 1.3.9 release
- OpenSM
  - 3.3.17 release or beyond

OpenMPI

- RDMA CM AF_IB connector contributed to master branch recently
  - Thanks to Vasily Filipov @ Mellanox ☺
  - Need to work out release details
    - Not in 1.7 or 1.6 releases

Deployment



SM

SA

Mgmt Nodes

Compute Nodes

IB SSA Core package

IB SSA Distribution package

IB ACM Shipped by distros

Project Team

- Hal Rosenstock (Mellanox) - Maintainer
- Sean Hefty (Intel)
- Ira Weiny (Intel)
- Susan Coulter (LANL)
- Ilya Nelkenbaum (Mellanox)
- Sasha Kotchubievsky (Mellanox)
- Lenny Verkhovsky (Mellanox)
- Eitan Zahavi (Mellanox)
- Vladimir Koushnir (Mellanox)

Development

- Mostly by Mellanox
  - Review by rest of project team
- Verification/regression effort as well

Initial Release

- Path Record Support
- Limitations (Not Part of Initial Release)
  - QoS routing and policy
  - Virtualization (alias GUIDs)
- Preview – June
- Release - December

Future Development Phases

1. IP address and name resolution
    1. Collect <IP address/name, port> up SSA tree
    2. Redistribute mappings
    3. Resolve path records directly from IP address/names
2. Event collection and reporting
    1. Performance monitoring

Summary

- A scalable, distributed SA
- Works with existing apps with minor modification
- Fault tolerant

- Please contact us if interested in deploying this!

Thank You

OPENFABRICS ALLIANCE

10TH ANNUAL INTERNATIONAL DEVELOPER WORKSHOP