Rightscaling: Varying Data Safety Techniques With Scale

MSST 2019

Lance Evans
Office of the CTO
lance@cray.com

RESILIENCY AT SCALE

WHAT IS THE MEANING OF THIS ??

RELIABILITY

Pertains to data accuracy over time

AVAILABILITY

Pertains to data accessibility over time

SERVICEABILITY

Pertains to preventing access interruption

CONVENTIONAL HPC WISDOM

- The largest systems require higher RAS characteristics
- Higher performance demands require more components, thus greater resiliency
- More capacity increases likelihood of failure or corruption
- Higher duty cycle increases corruption risk

TYPICALLY TRUE BUT NOT ALWAYS

- Largest systems must be prepared for app-level failure, leverageable for storage
- Higher performance is achieved without resiliency
- Higher capacity and throughput are achievable today with smaller systems
- Environment (e.g. heat) and time are larger factors than wear (within specs)

TODAY'S HPC SYSTEMS

- Typical Top Ten HPC System
  - Capacity tier approx. 500PB, O(25,000) HDDs in a system, O(5TB/sec)
  - Performance tier approx. 50-100PB, O(10,000) SSDs, write/read O(25/40)TB/s
  - Local IOPS tier approx. O(10,000-40,000) SSDs, write/read O(60/100)TB/s
- Component Level
  - Quality components have MTBF in the O(2M hour) range
  - At O(100,000) components in a top 10 storage system, MTBF is O(20) hours
  - Something breaks every day and can take from minutes to hours to find and repair
- System Level complexities
  - RAS in storage system design conflicts with performance demands
  - Job-scope orchestration of storage resources, with migration between tiers
  - Software changes happen rapidly
  - Enormous challenge to test / prove at scale

RECENT DEVELOPMENTS

- Traditional checkpoint performance requirement experiencing quadratic growth
  - Compute JMTTI is O(10-20 hours), while storage is expected to be reliable
  - RAM+HBM growth to O(10PB) demands checkpoint bandwidth of O(50TB/s)
- Hardware
  - Excess HDD capacity required for JMTTI > 200 (forward progress > 90%) checkpoint
  - Flash price reductions enable checkpoints without excessive system size
  - Fabric link speeds moving toward 40GB/s
- New computing techniques emerging
  - Machine learning at scale
  - AI techniques to accelerate traditional solver methods
  - High-productivity languages and programming models

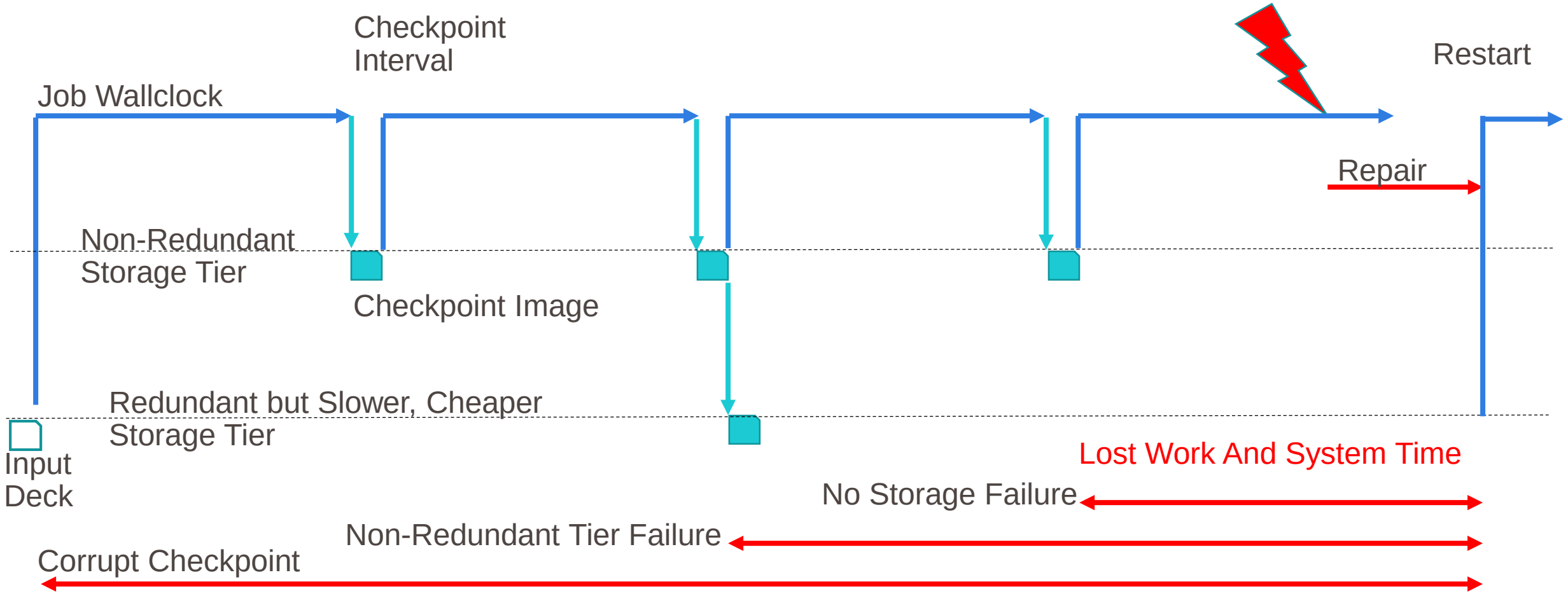
CUSTOMER REQUIREMENTS VARY

“We can’t tolerate the performance hit that adding reliability would introduce. We’ve done the math and can make more progress despite the outages.”

“We can’t tolerate the unreliability that the extra performance gives us. We can’t run on storage that will fail jobs and lose our data.”

Classic Checkpoint / Restart Resiliency Model

Mature application software designed for failure recovery
Generally used by large-scale ”capability” workloads & systems



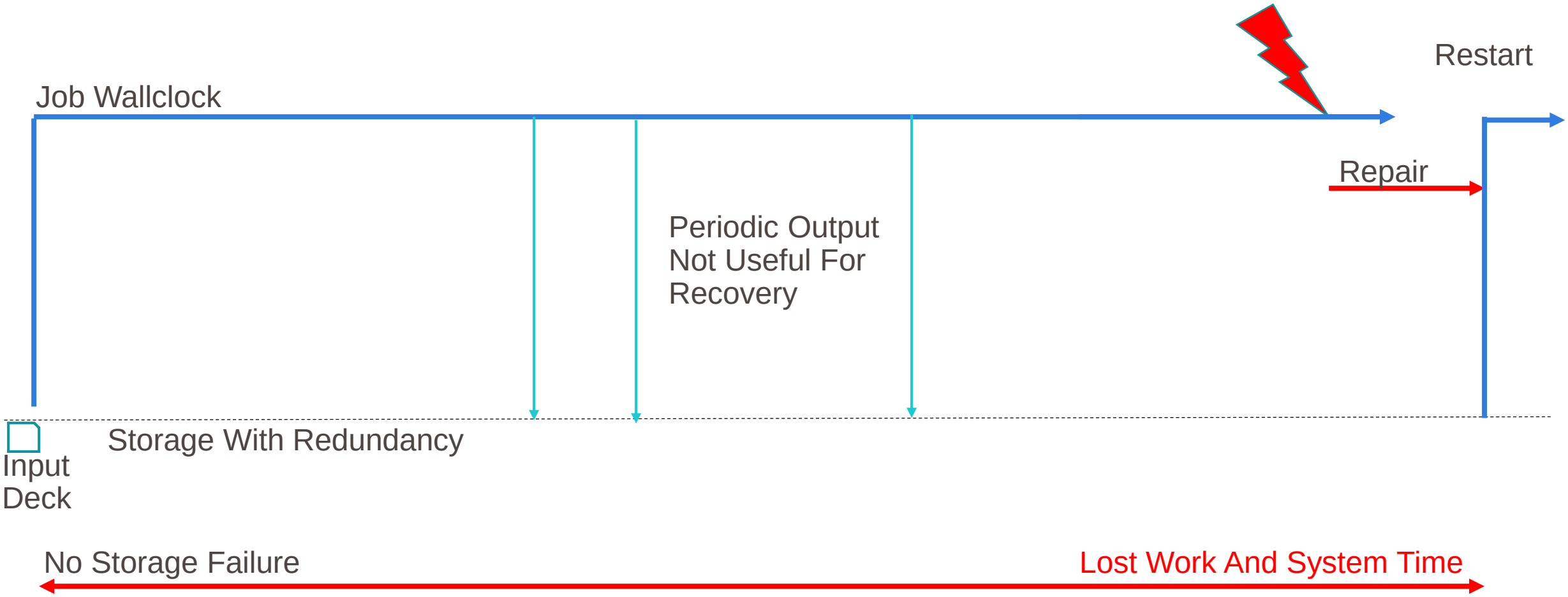
Classic Checkpoint / Restart Resiliency Model

- In a well-designed system
  - Average time to job interruption is reasonably long
  - Checkpoint interval is reasonably short
  - Intermediate data is useful for analysis and steering
  - Likelihood of non-redundant tier failing at a time of need, is low
  - Likelihood of silent corruption in non-redundant tier is low; can be mitigated
  - Machine needs to be taken down periodically anyway
- Conclusions
  - Design for fastest forward progress (processing speed, checkpoint speed)
  - Don’t design for redundancy in the expensive tier; put it in the backing tier
  - Plan for and automate recovery at the application and job control level

Commercial / Production Resiliency Model

Software designed assuming system is reliable

Generally used by smaller scale "capacity" workloads & systems

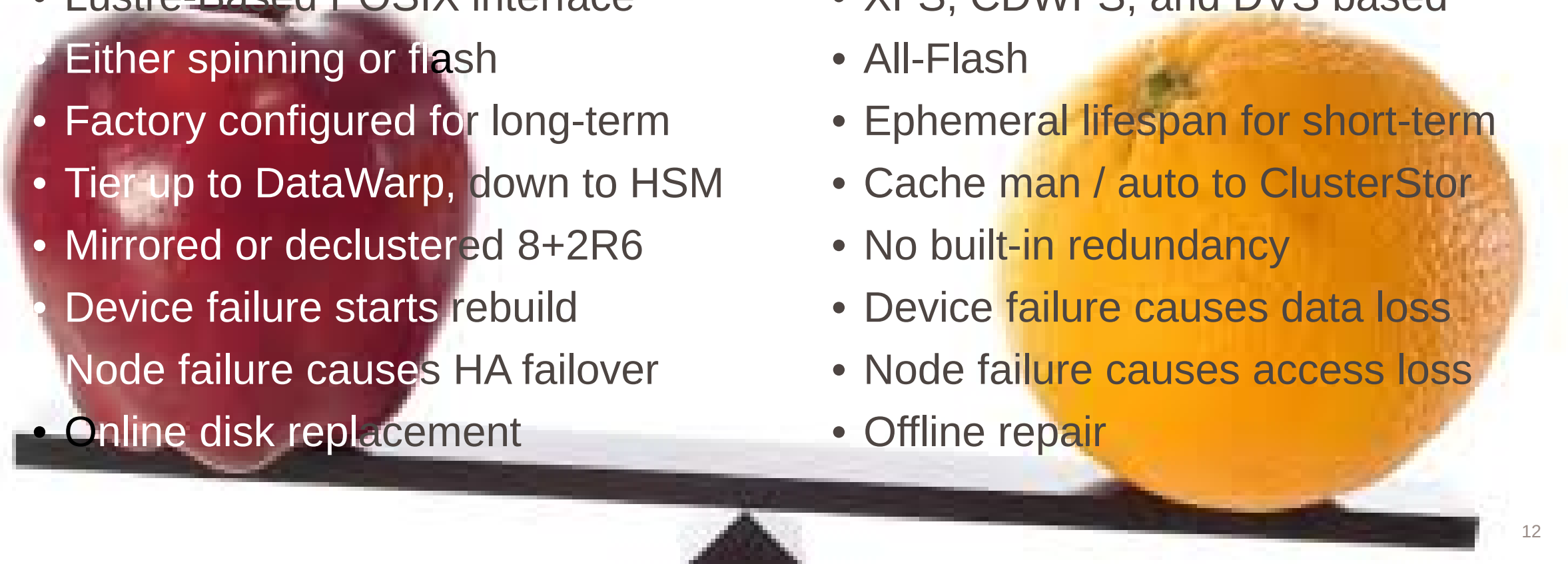


Commercial / Production Resiliency Model

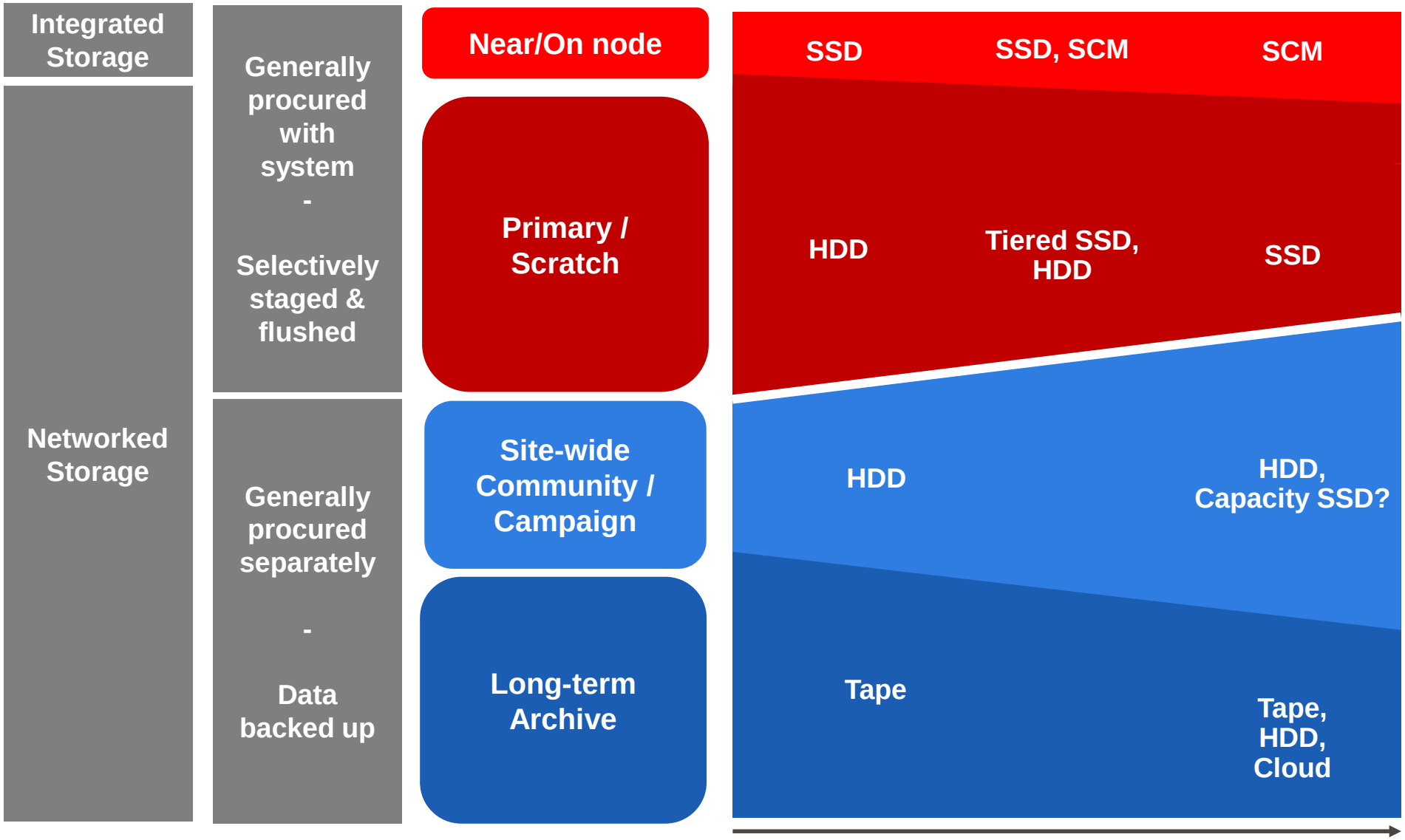
- In a well-designed system
  - Primary storage always remains online
  - Mature storage software is designed for failure recovery
  - Recovery is infrequent (typically due to job size)
  - High productivity achieved with rapid software development
- Conclusions
  - Accept slightly lower system performance to achieve resiliency
  - Ease of use by system consumers is paramount
  - Predictability of workflow is worth extra cost

CRAY'S TWO MAIN FILE PRODUCTS

- ClusterStor
  - Primary file system
  - Full POSIX compliance
  - Lustre-Based POSIX interface
  - Either spinning or flash
  - Factory configured for long-term
  - Tier up to DataWarp, down to HSM
  - Mirrored or declustered 8+2R6
  - Device failure starts rebuild
  - Node failure causes HA failover
  - Online disk replacement
- DataWarp
  - Burst Buffer
  - NFS file semantics
  - XFS, CDWFS, and DVS based
  - All-Flash
  - Ephemeral lifespan for short-term
  - Cache man / auto to ClusterStor
  - No built-in redundancy
  - Device failure causes data loss
  - Node failure causes access loss
  - Offline repair



DATA PATH DIVERSITY



Data Lifetime	Throughput (Rate)
Hours to days	Double digit to single digit TB/s, (Trillions)
Days to months	Single digit TB/s to triple digit GB/s (Billions)
Months to years	Triple digit to double digit GB/s (Millions)
Years to forever	Double digit GB/s (Thousands)

TAKEAWAYS

- Larger “capability” systems benefit from less primary resiliency (burst buffer)
- Smaller and “capacity” systems benefit from commercial-grade features, reliability
- Fewer data paths is better, but two are recommended
- A backing store with very high reliability is beneficial in all cases
- Design each system uniquely in consideration of specific use cases

OPEN QUESTIONS

Can resiliency methods be more effective moved into client / app space?

Can redundant components be eliminated while improving resiliency?