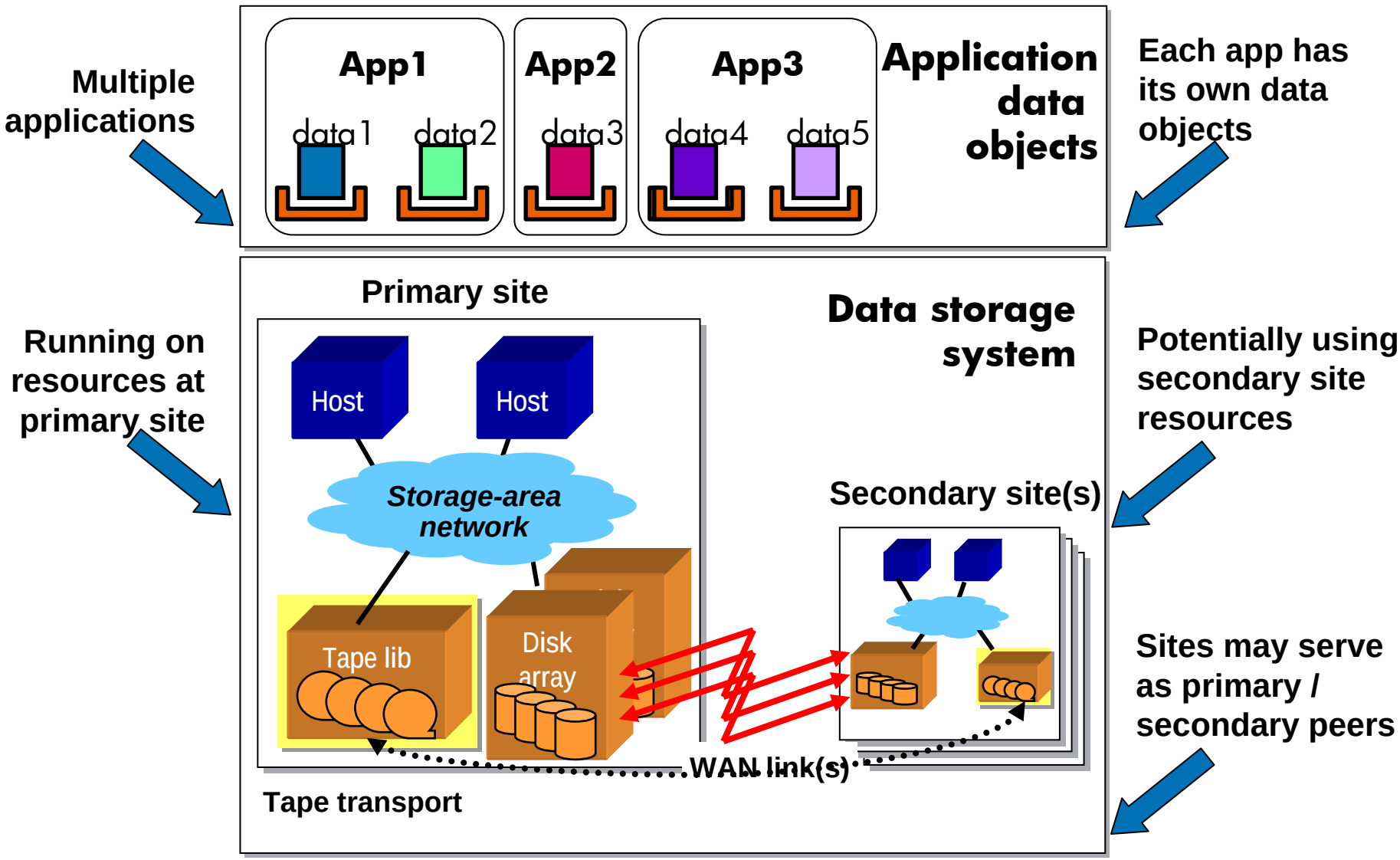
Challenges in modeling enterprise storage systems

Arif Merchant
Hewlett-Packard Laboratories
Contact: arif.merchant@hp.com

An enterprise storage system



Storage System models
Why should you care?

- Storage is a significant part of IT costs
  - Over 35% for enterprises with 5000+ employees (IDC 2002)
  - Large fraction of storage cost is management
  - Environment is complex: 100s of applications sharing petabytes of data
  - Stringent application requirements – failures can be catastrophic (Survey: $89K-6.4M/hr of downtime)
- Good storage models are crucial for managing storage
  - Systematic, accurate models needed for informed choices
- Uses of models
  - Capacity planning – green field & consolidation
  - Storage design
  - On-going management

Storage management automation

Where storage models fit in [Anderson2002,Alvarez2001]

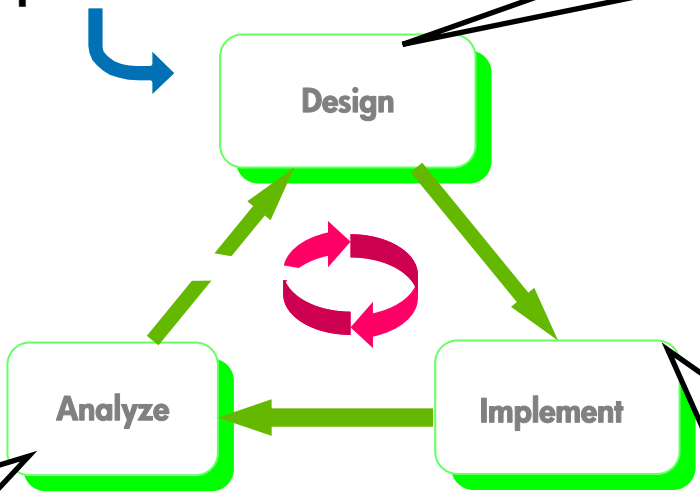
Understand needs

- Workload characterization
- System components

(Changing) business requirements

Determine solution

- Select devices + configurations
- Model to verify config. Is good
- Assign load to devices



Iterative feedback loop

Monitor usage

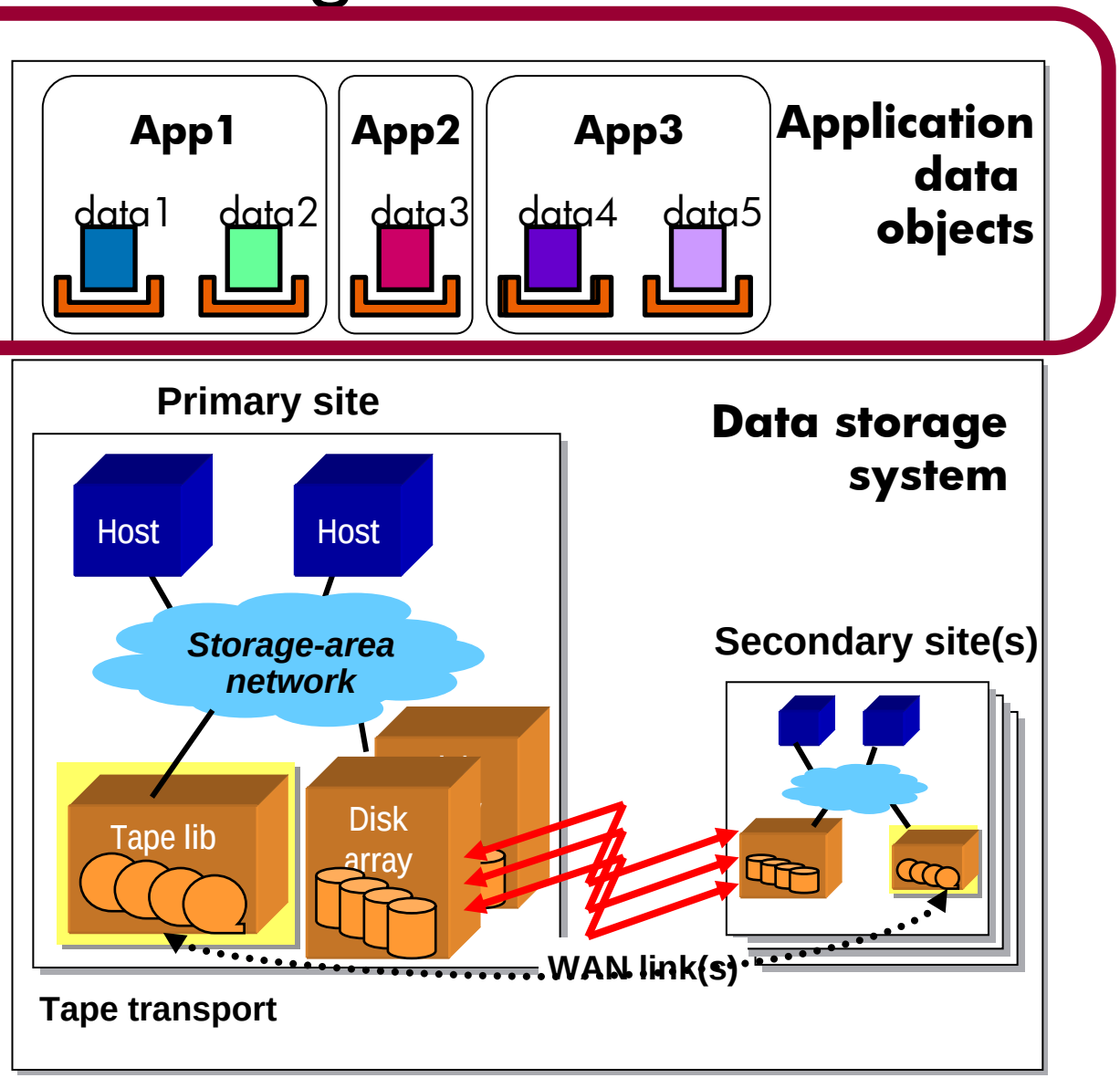
- Measure workload, re-characterize
- System response

Construct solution

- Configure devices, LVM, …
- Migrate data

Components of storage models

Workloads



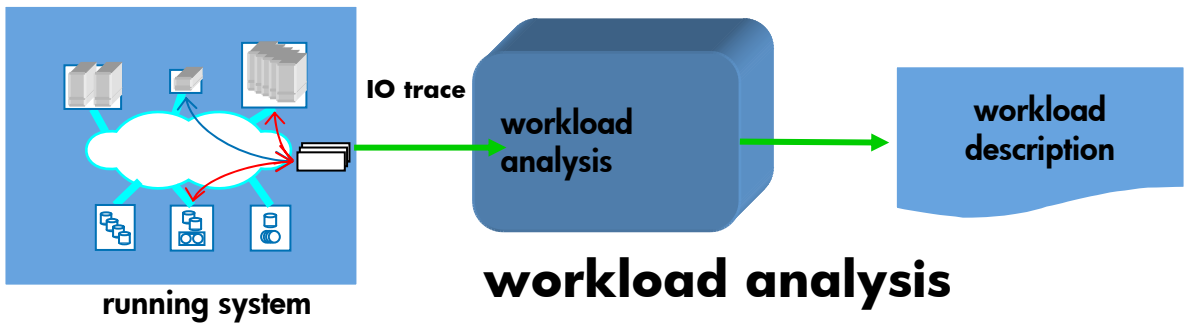
Workload characterization

- Need
  - Compact characterization
  - Adequate for predictive storage system models

- Minimal parameter set:
  - Size of data
  - IO rate, IO size distribution
  - Reads vs. writes
  - Sequentiality, spatial locality
  - Temporal locality (e.g., frequency of repeated access)
  - Concurrency (Simultaneously outstanding requests)

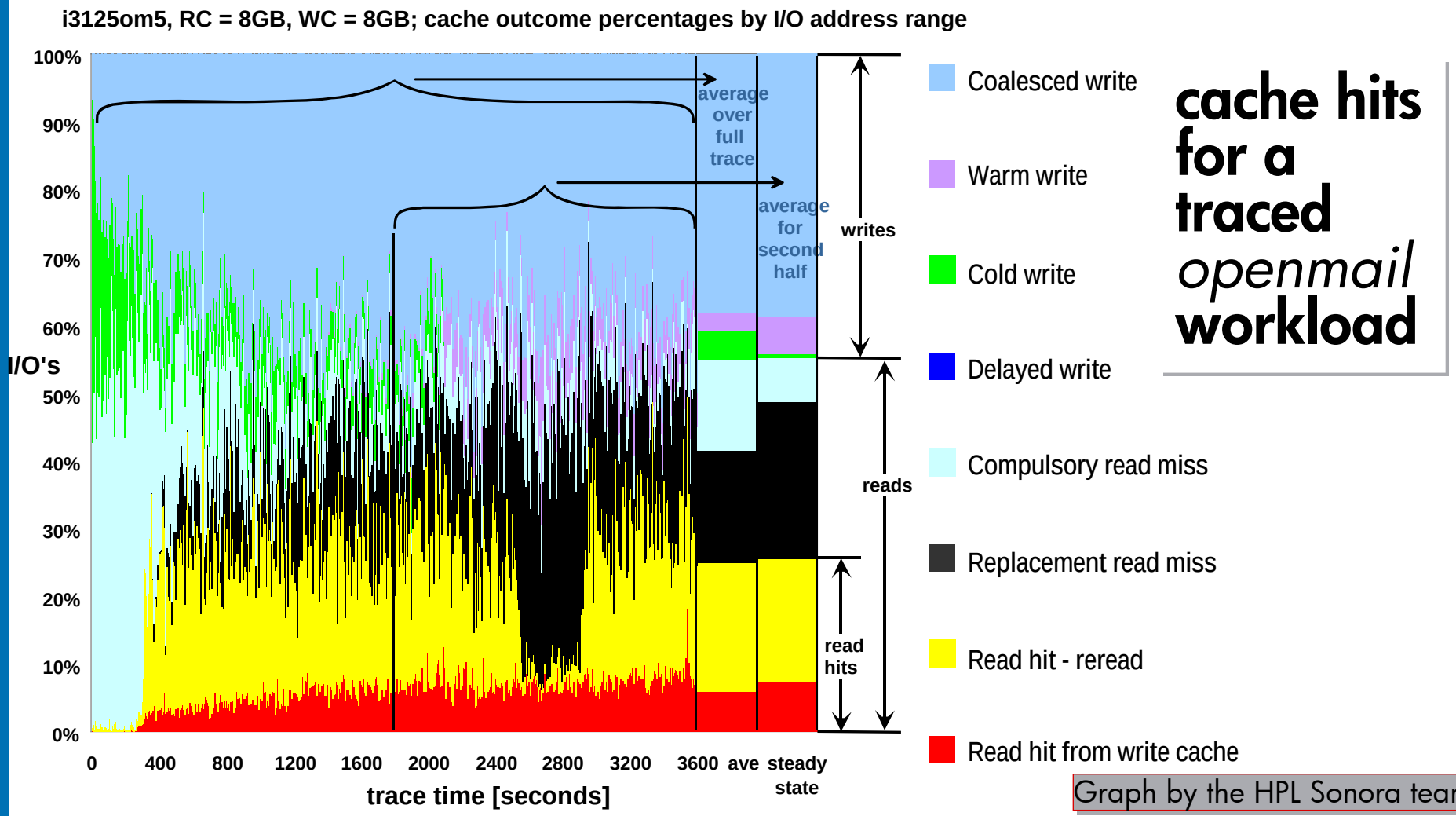
Measuring workloads

- How do we acquire workload parameters?
  - Users do not know workload details
  - Measurements on production system may add load
  - Storage devices have limited, vendor-dependent measurement points (although industry standards SMI-S are helpful!)
  - Workload behavior may depend on hardware & configuration
  - **Open problems:**
    - **Minimum perturbation methods for measuring workload parameters**
    - **Tradeoffs between inaccuracies of workload parameters and model accuracy**



Workload attributes – access variability
How do you compactly capture this complex behavior?

Sample results for a single run

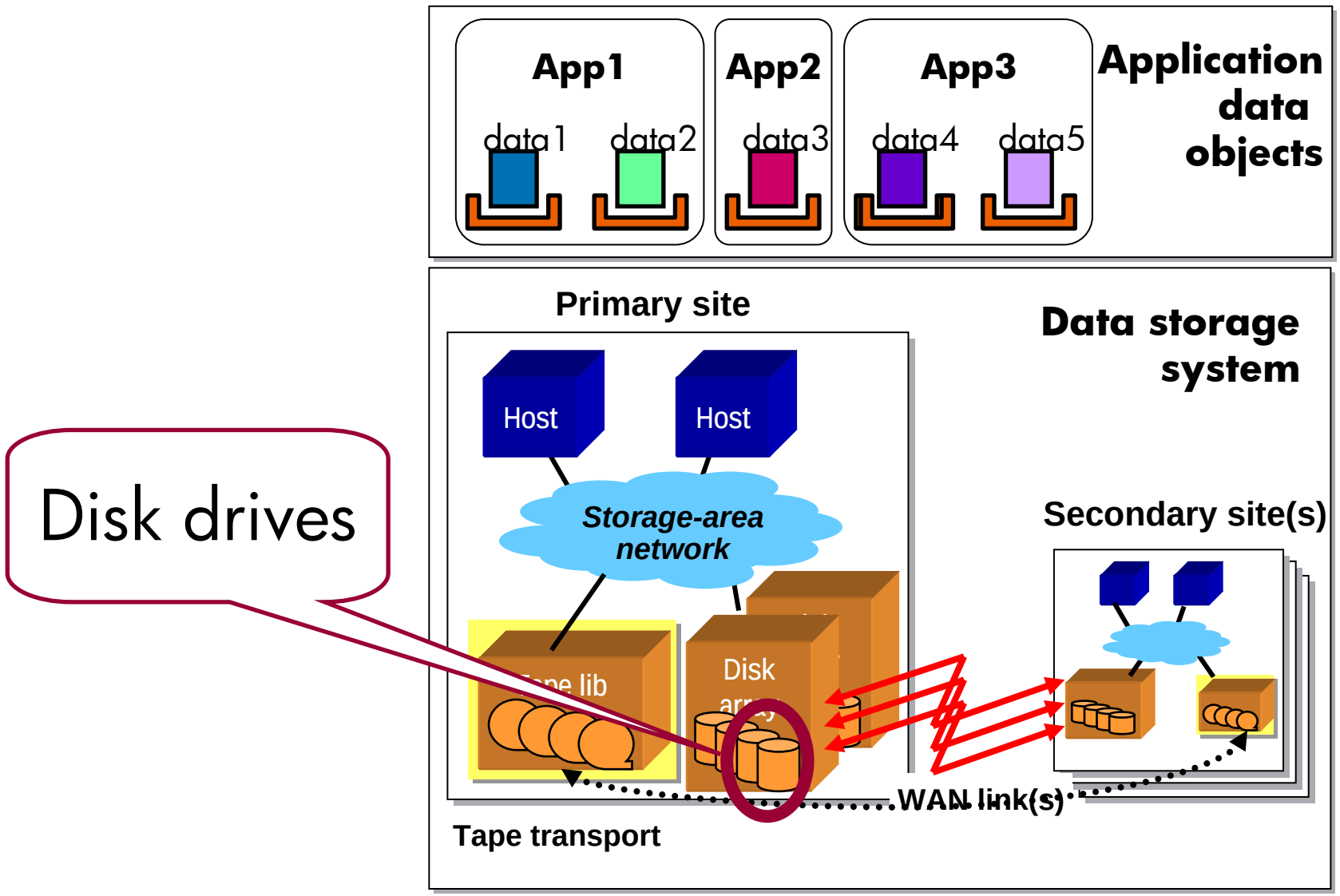


Graph by the HPL Sonora team

Workload specification challenges

- What is an adequate workload specification?
  - Complex request arrival patterns
    - Self-similar? Dependent on response?
    - Seasonality, trends
  - Complex access patterns
  - Locality (spatial and temporal)
  - Concurrency variation
  - Correlation/interference between workloads
- Synthesis of workloads accurately representing real workloads

Components of storage systems
The basic storage device



Disk drives simplified

track

sector

head

platter

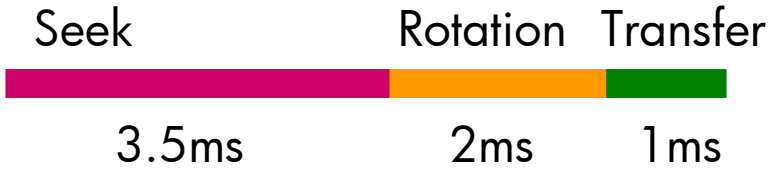
1 01 11101 00 11100 00 1

Block

Disk drives specs

High end (enterprise) disk

Seagate Cheetah 15K drive



64 KB random read service time

- Typical access times (seek + rotation):
  - 5.5ms
  - Not improving quickly (limited by speed of physical movement)
- Typical sustained transfer rate:
  - 58 to 96 MB/s or ~1ms for 64KB
  - Increasing quickly over time (increases with bit density)
- Annual failure rate: 0.62%

High disk access time

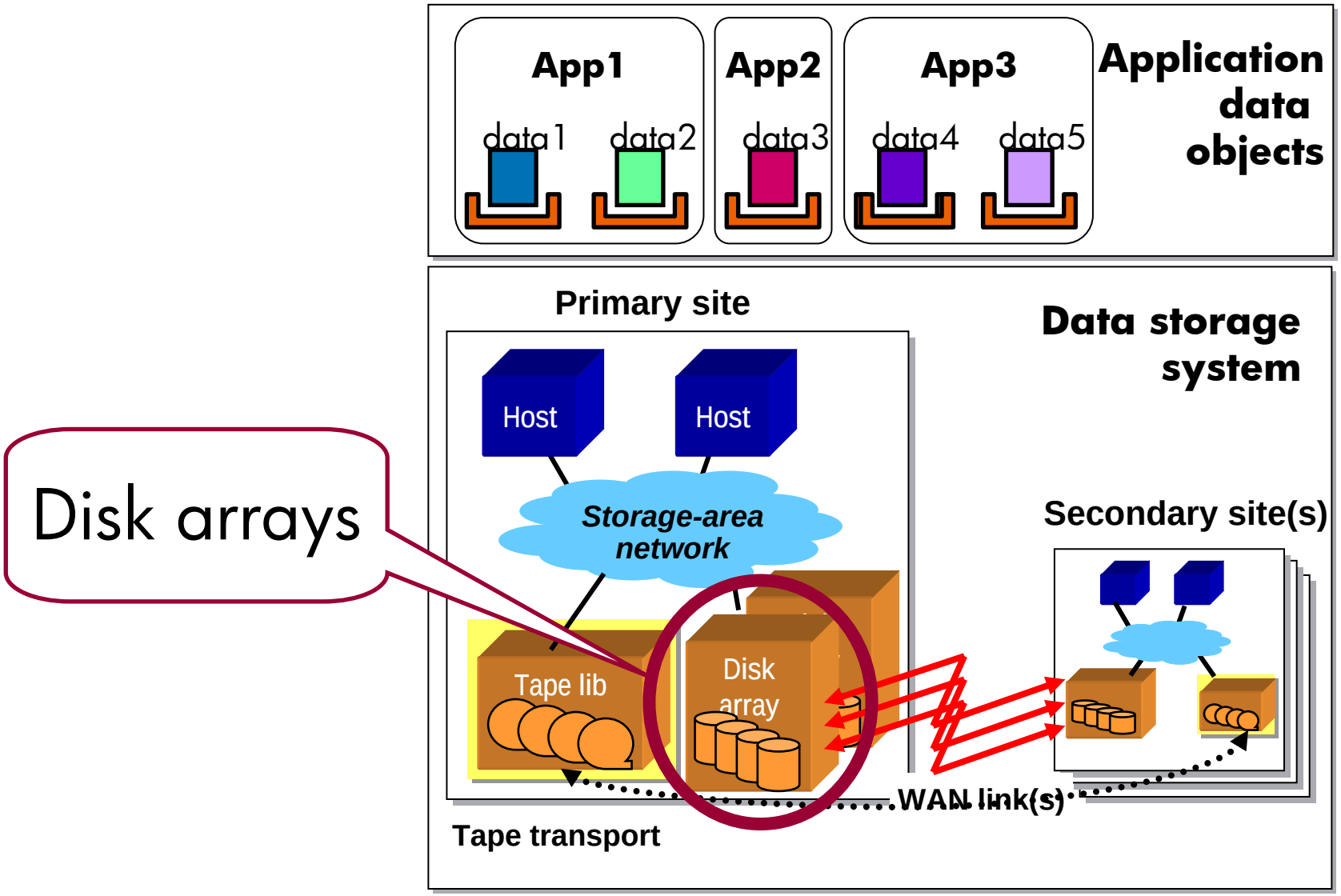
The root of many storage (modeling) problems

- Random IO throughput << Sequential
  - 1.4MB/sec (8KB random, 1 request at a time) versus 58-96MB/sec sequential
- Disk scheduling algorithms
  - CLOOK, elevator, etc. reorder pending IOs to reduce seek time
  - Random IO throughput is higher with multiple outstanding IOs
- Disk cache pre-fetching:
  - Mixing multiple sequential workloads reduces throughput
  - Detecting sequentiality and pre-fetching sequential data improves matters a little
- Many other optimizations …

Disk drive models

- Long history of disk drive models
  - [Ruemmler1994] is a classic paper, describes access characteristics and parameters
  - [Shriver1998] includes effects of caching and scheduling
  - DIXtrac [Schindler1999] extracts disk parameters automatically
  - DiskSim [Ganger1998] is an accurate disk system simulation environment

Components of storage models
At the next level

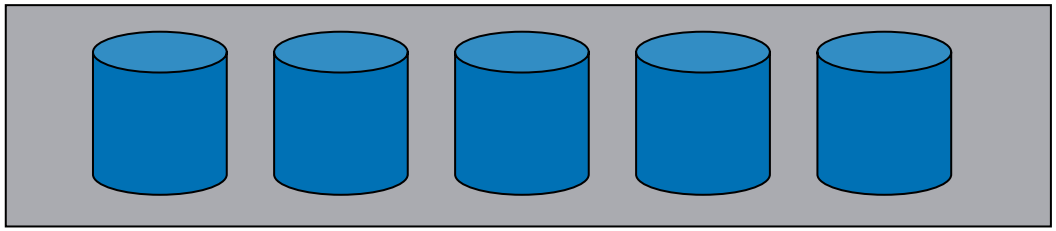


Disk arrays: Aggregation and Virtualization

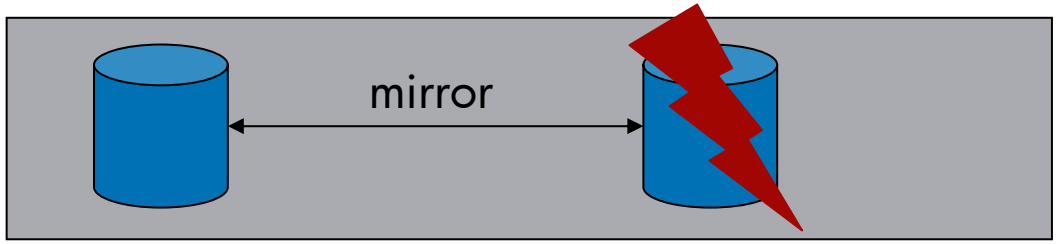
- In its simplest form, a disk array is:
  - Mechanical enclosure
  - Power and cooling
  - Connectivity

Disk Arrays: Aggregation

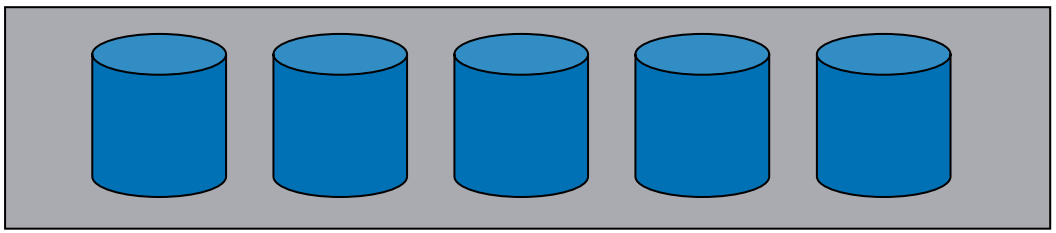
- Capacity

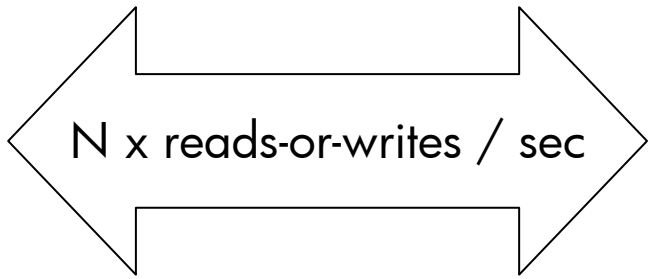


- Reliability



- Performance

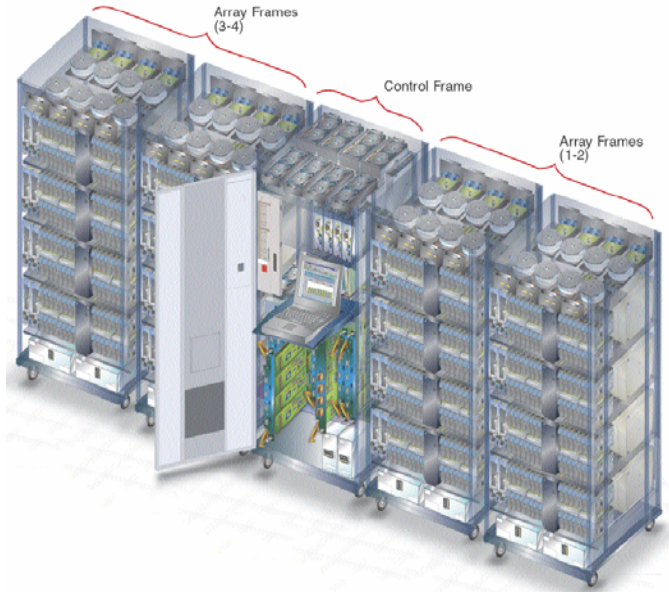




Disk Arrays: Virtualization

- Lets you provision and "slice and dice" your block storage into *volumes*
- Volumes are logically big disk drives
- Different volumes may use different redundancy schemes – mirroring, error correcting codes, etc.
- Accomplished through processing and software on the array, but it can also happen in other places (e.g., SAN switches, the host).

Real Arrays: Small, Medium, Large

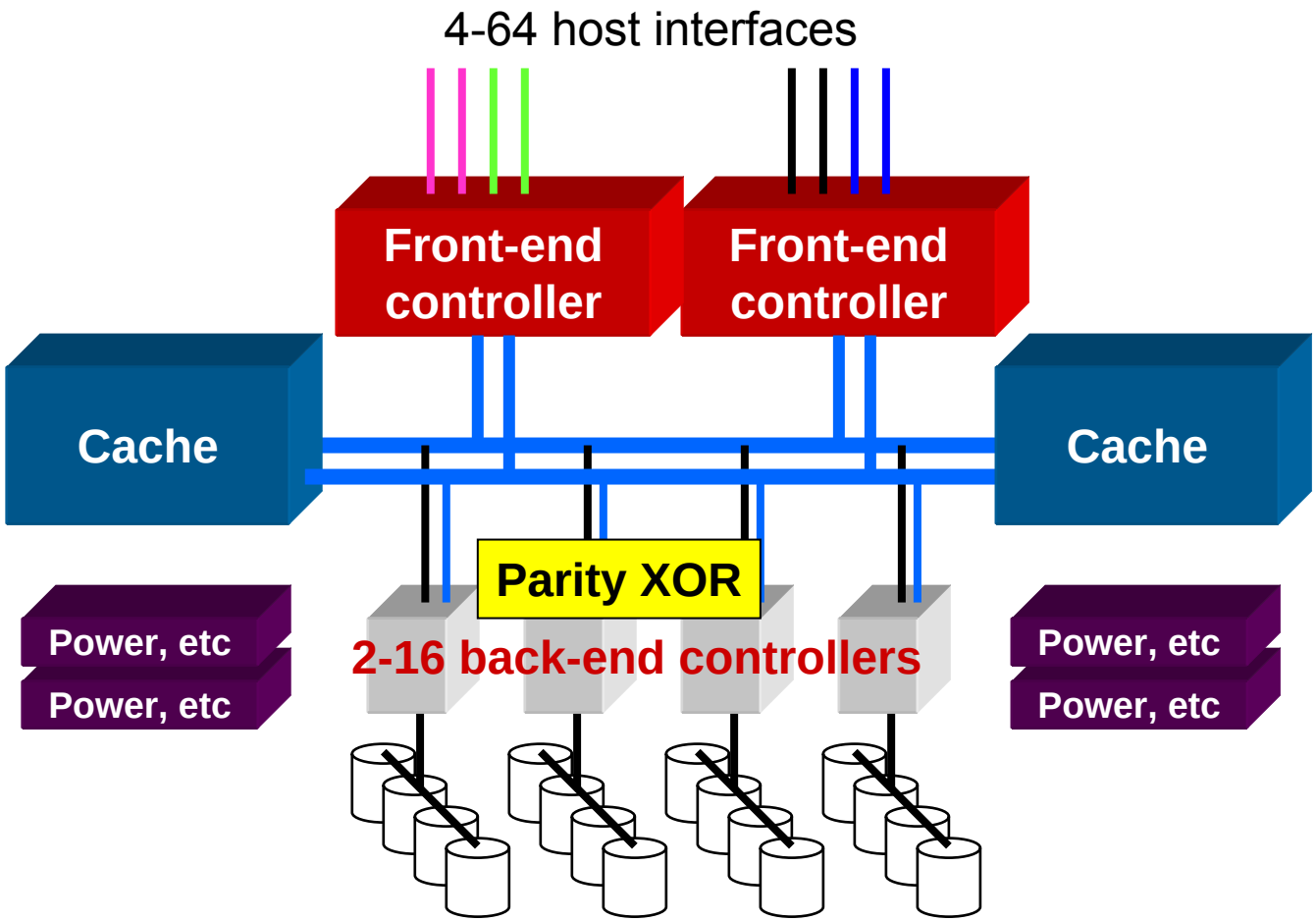


	Entry	Mid-range	High-end
Capacity	10-14 drives	100 drives	1200 drives
	4-6 TB	80 TB	332 TB internal
Cache	0.5 GB	4 GB	256 GB
Redundancy	Some	Dual	Everything
Form factor	Shelf	Cabinet	Multiple Cabinets

Disk Array models

- Simple availability models have existed as long as arrays [Patterson1988, Gibson1992]
  - Based on Markov assumptions
- Many over-simplified array performance models
  - Assumptions are frequently unrealistic
    - Poisson request arrivals
    - Focus on disks, ignore cache, controller, etc
  - Models are not validated against real disk array
    - Validated against unvalidated simulators
  - Our attempts to use such models found up to 300% errors vs. throughput measurements on FC-30 array (a low-end, 30-disk array, now obsolete)

Disk array: logical structure



Performance models of real disk arrays
Why it is hard

- Modern disk arrays are complex
  - Large array caches and complex interconnections
  - Many optimizations, some repeated at multiple levels
    - Request coalescing, adaptive prefetching, sequentiality detection, efficient cache management, etc.
  - Combined effects of multiple optimizations are hard to understand
- Workloads are complex
  - Irregular arrival processes, spatial & temporal locality, correlation with other workloads, etc.
  - Workloads can merge/split, with more complex results

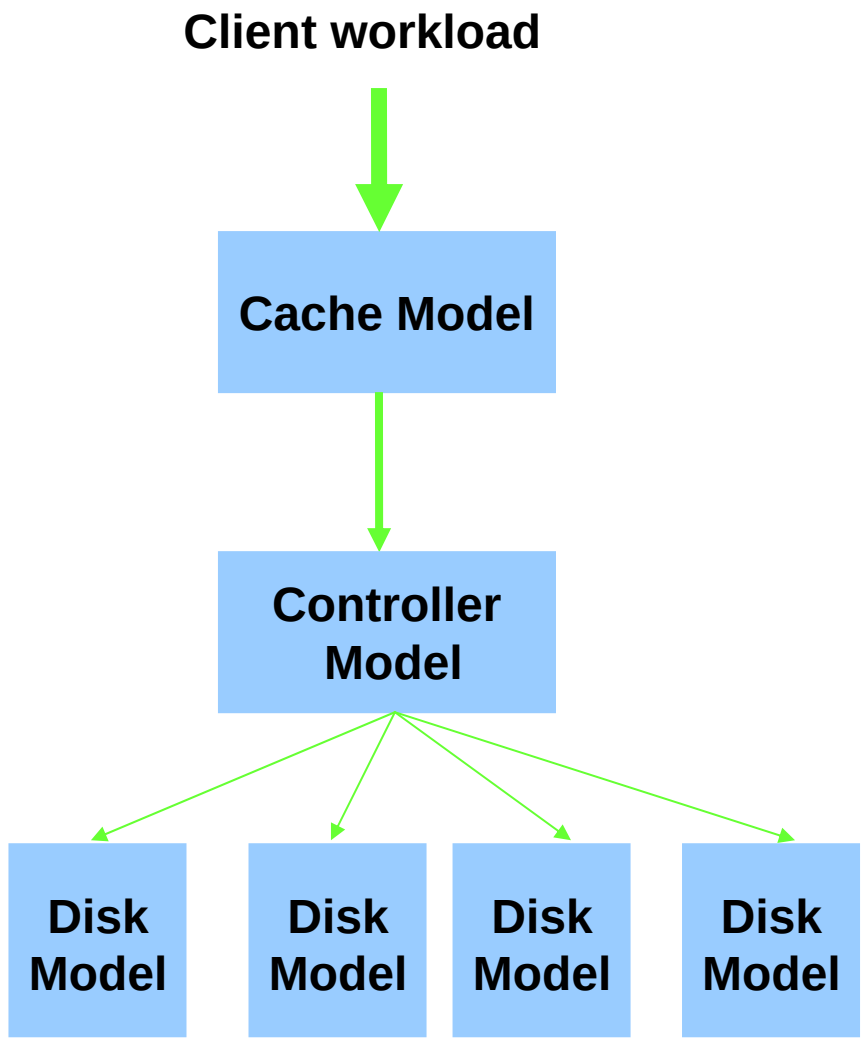
An analytical disk array model

Delphi [Uysal2001]

- Decompose the internal device structure as a tree
- Each node in the tree (*component model*) corresponds to one or more physical array components
- Component models
  - Optionally transform the workload before passing it down the tree
  - Optionally impose constraints on achievable performance
  - Compute local metrics (e.g., utilization)
- Highly modular ➔ reuse
  - Hopefully, models of different arrays can be derived by combining and tweaking modules

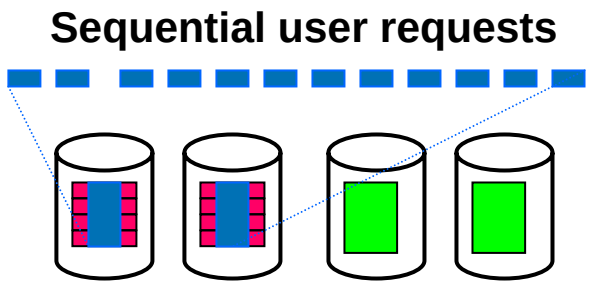
Overall Model Structure

- Throughput models
  - Inputs:
    - Array configuration
    - Workloads
- Hierarchical model
  - Mirrors array architecture
  - Relevant components only
- Component models
  - Throughput limits
  - Workload transformations



Component models

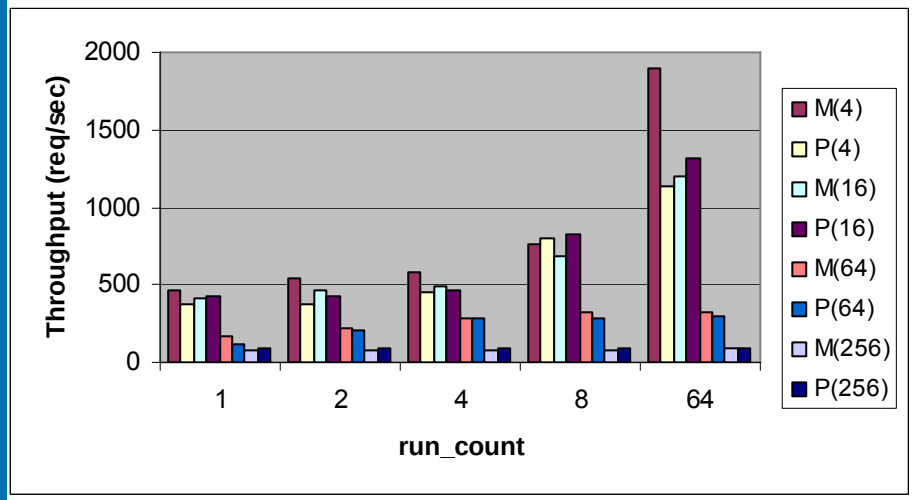
- Component models can be simple
  - E.g., cache model transforms workload by reducing read rate based on hit/miss probability estimate
- … or complex
  - Controller model handles splitting and coalescing of requests to account for the layout of data, the request size distribution and how sequential the workload is.



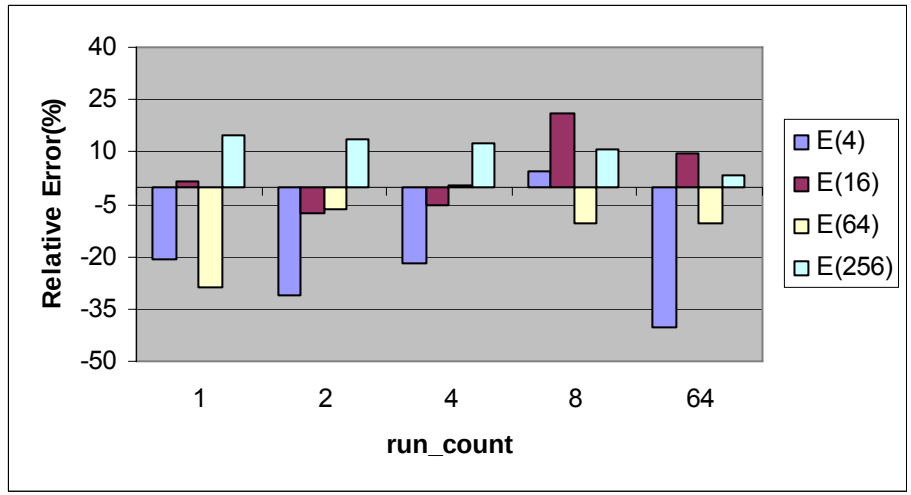
- Component models can be individually tuned until adequate

Empirical validation
RAID 1/0 model, reads (4-disk LU)

Throughput



Prediction Error



- FC-60 mid-range disk array, single controller
  - 256 MB battery-backed cache, 4 KB cache page size
- Synthetic workloads:
  - Up to 64 requests outstanding, size 4KB-256KB
- Prediction errors:
  - avg. absolute 14%, range: -37% to +20%

Limitations of analytical approach

- Pros
  - Good for understanding array behavior
  - Model is fairly robust for a given array – it can be applied to most configurations with consistent results
- Cons
  - Deep understanding of array required to build models
  - Labor intensive: too much human time (weeks to months)
  - Need to fine-tune each component model (second-guess array and disk policies)
  - Limited accuracy
  - Predicting some metrics can be tricky (e.g., response time)

Black box performance models

- Interpolating device performance from measurements [Anderson2001]
  - Parameterize space of workloads
  - Measure device performance for many, many workload parameter values using synthetic workloads
  - Predict performance of actual workloads by interpolating
- Pros
  - Minimal human time, expertise requirements
  - Accuracy is good (<20% error) if synthetic workloads representative
- Cons
  - Requires very large amounts of device time for measurements
- Some promising recent results
  - "Relative fitness" includes performance of workload on another device as input to model [Mesnier2006]

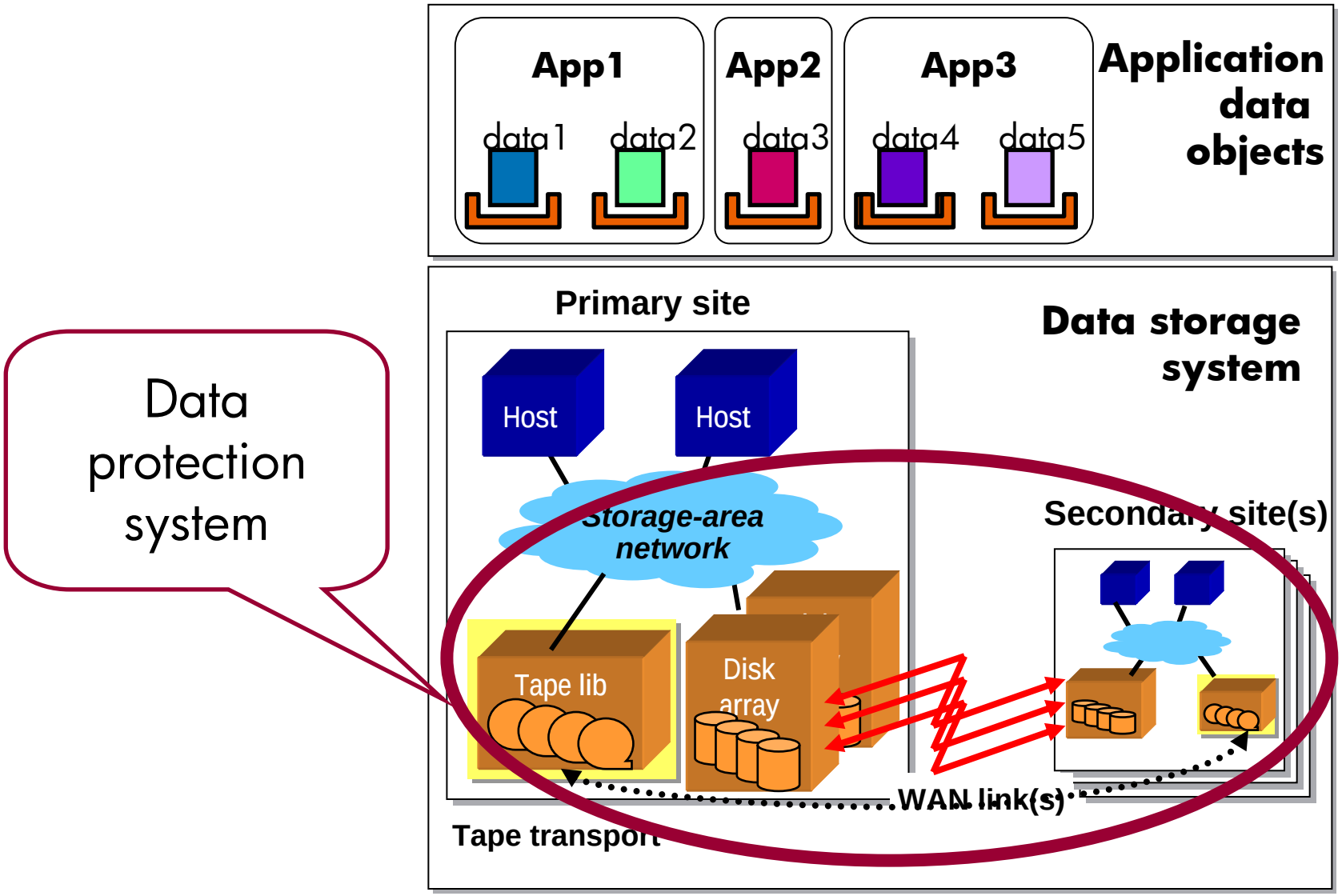
The state of storage performance modeling

- There are no good performance models of complex disk arrays
- Only simple subsystems (e.g. disk) can be modeled accurately
- Models of complex systems are not robust
  - Some are accurate, but only locally (e.g., black-box models)
  - Some cover a broader set of situations, but not accurate
  - Most break if you switch arrays
- Most models are not validated against real systems
  - Testing one model against another (e.g., analytical vs. simulation) is not convincing when neither is validated against a real system!

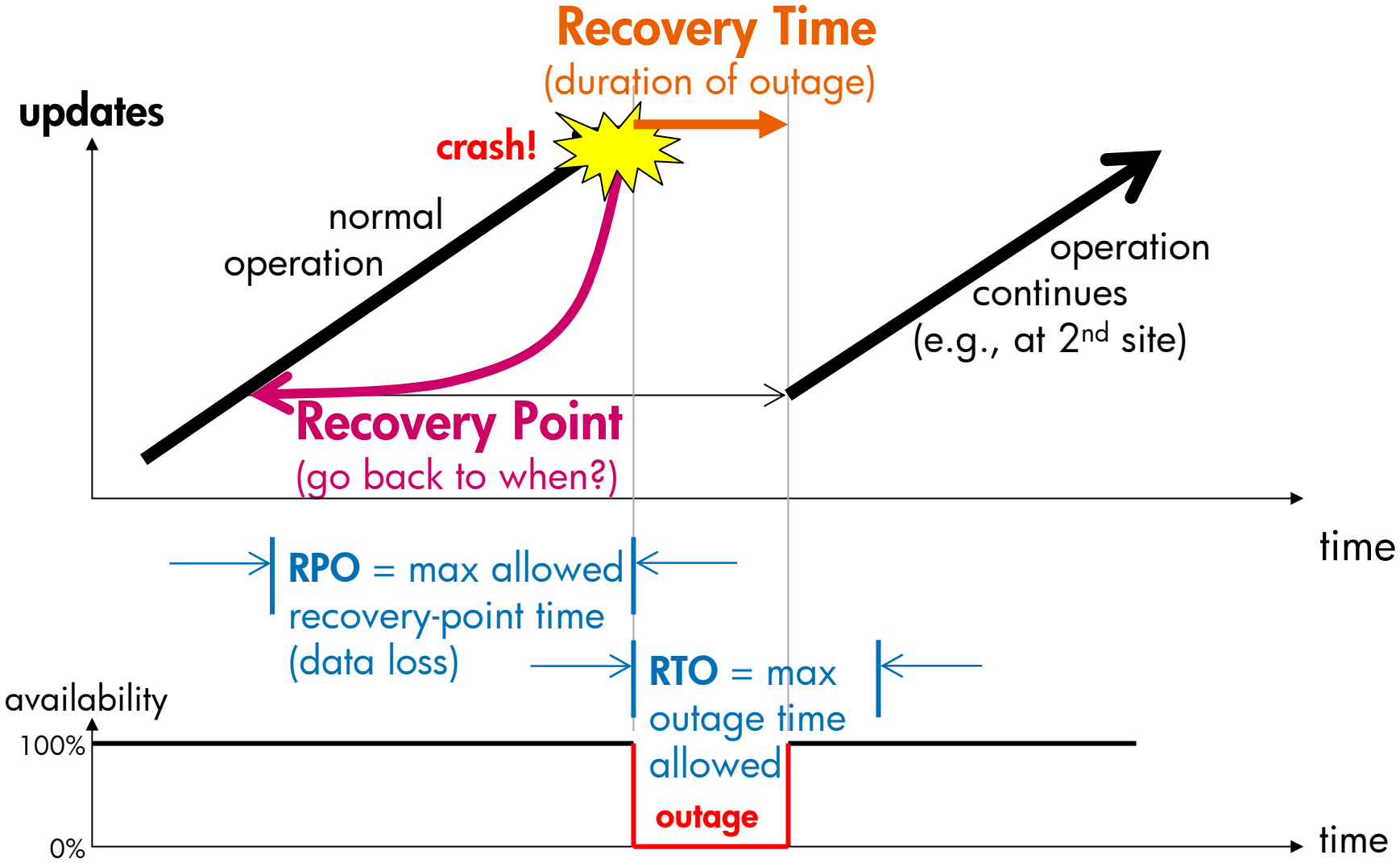
Open problems in storage performance modeling

- Accurate models of workload transformation
  - How does a controller transform a workload?
- Response time models (good ones)
- Performance in degraded modes (after failure, during recovery)
- Robust black-box models that are fast to build
- Simulators validated against real arrays

Components of storage models
Putting the parts together

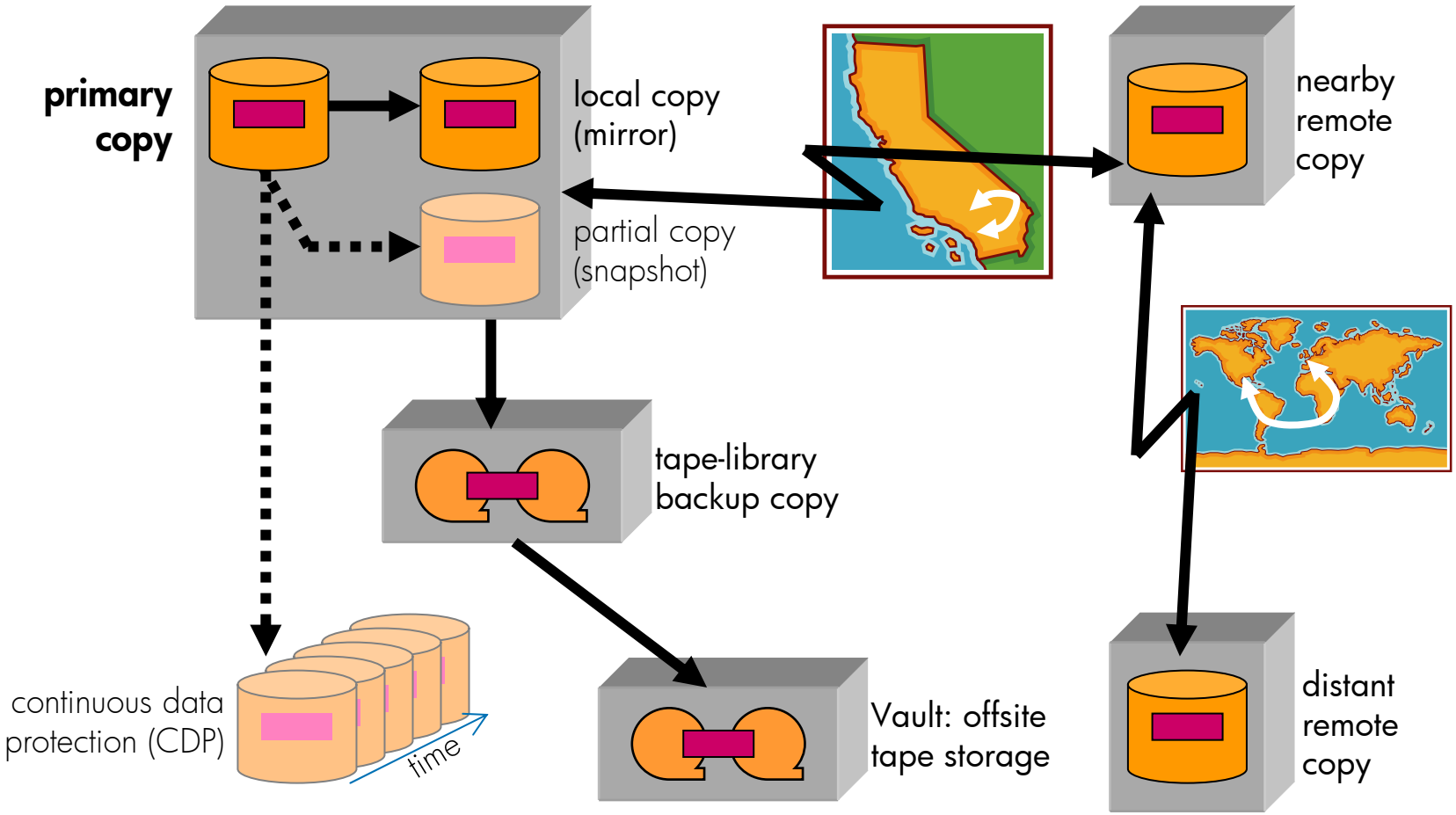


Anatomy of a failure
Recovery metrics

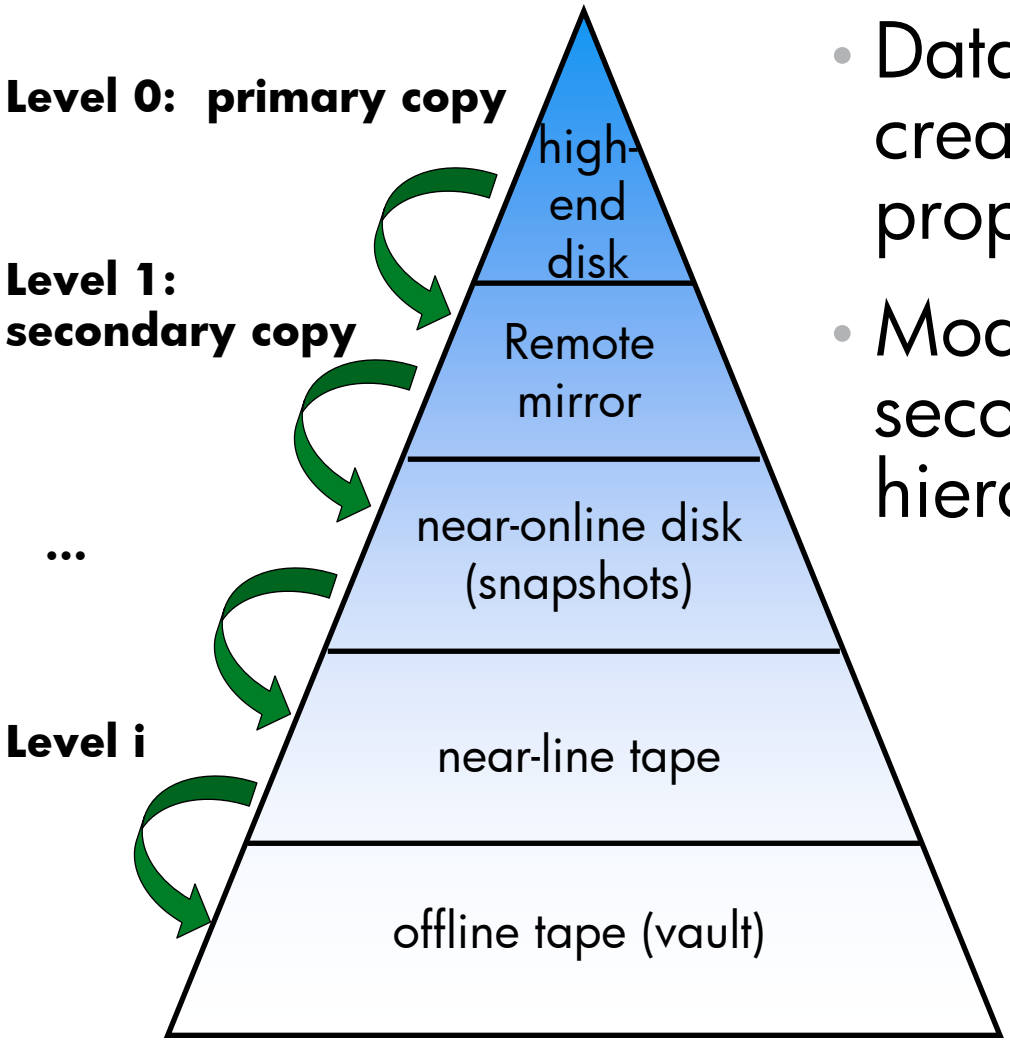


Data protection setups

Replication: propagating Point-in-Time (PiT) copies



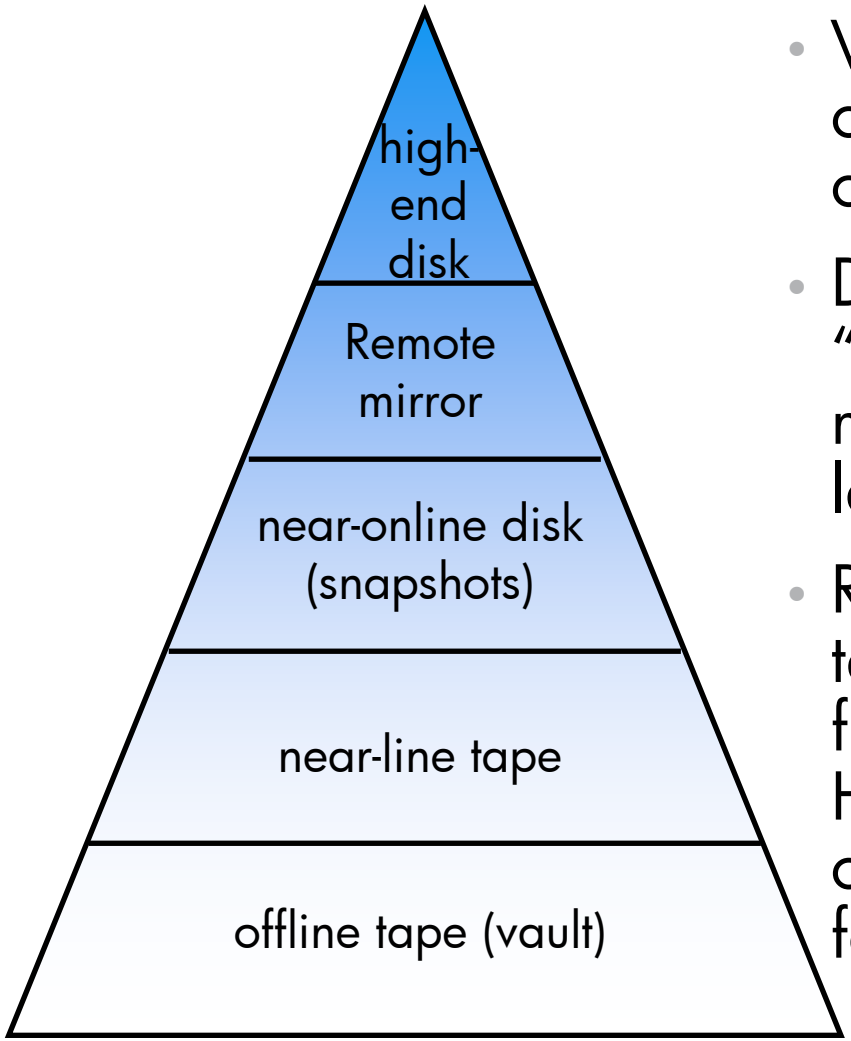
Data protection technique abstraction



- Data protection techniques create, retain and propagate PiT copies
- Model primary and secondary copies as hierarchy

Increasing levels:
- larger retention capacity
- longer recovery latencies
- less frequent PiT copies

Modeling the data protection system
A simple approach [Keeton2004]



- Verify resources: under normal operation, will replications overload resources?
- Data loss: at any time, how far "behind" is each level? How much data loss if intermediate levels fail?
- Recovery time: How long does it take to recover data to level ***n*** from lower levels if it has failed? How long does it take if some of the lower levels are still failed?

Case study: backup and vaulting

- Baseline: copy snapshot every 12 hours, weekly full backup (48 hr backup window), monthly remote vaulting
- Weekly vault: baseline, except weekly remote vaulting
- Weekly vault, F+I: weekly vault, plus incremental backups on weekdays (12 hr backup window)
- Weekly vault, daily F: weekly vault, baseline except daily full backups (12 hr backup window)

Case study

	Array failure			Site disaster		
Storage system design	**RT (hr)**	**DL (hr)**	**Total cost**	**RT (hr)**	**DL (hr)**	**Total cost**
Baseline	2.4	217	$11.94M	26.4	1429	$71.94M
Weekly vault	2.4	217	$11.96M	26.4	253	$14.96M
Weekly vault, F+I	4.0	73	$4.84M	26.4	253	$14.96M
Weekly vault, daily F	2.4	37	$2.98M	26.4	217	$13.18M

- Weekly remote vaulting policy improves site disaster recovery

Case study

	Array failure			Site disaster		
Storage system design	**RT (hr)**	**DL (hr)**	**Total cost**	**RT (hr)**	**DL (hr)**	**Total cost**
Baseline	2.4	217	$11.94M	26.4	1429	$71.94M
Weekly vault	2.4	217	$11.96M	26.4	253	$14.96M
Weekly vault, F+I	4.0	73	$4.84M	26.4	253	$14.96M
Weekly vault, daily F	2.4	37	$2.98M	26.4	217	$13.18M

- Adding daily cumulative incremental backups
  - Decreases recent data loss for array failure
  - Slightly increases recovery time

Case study

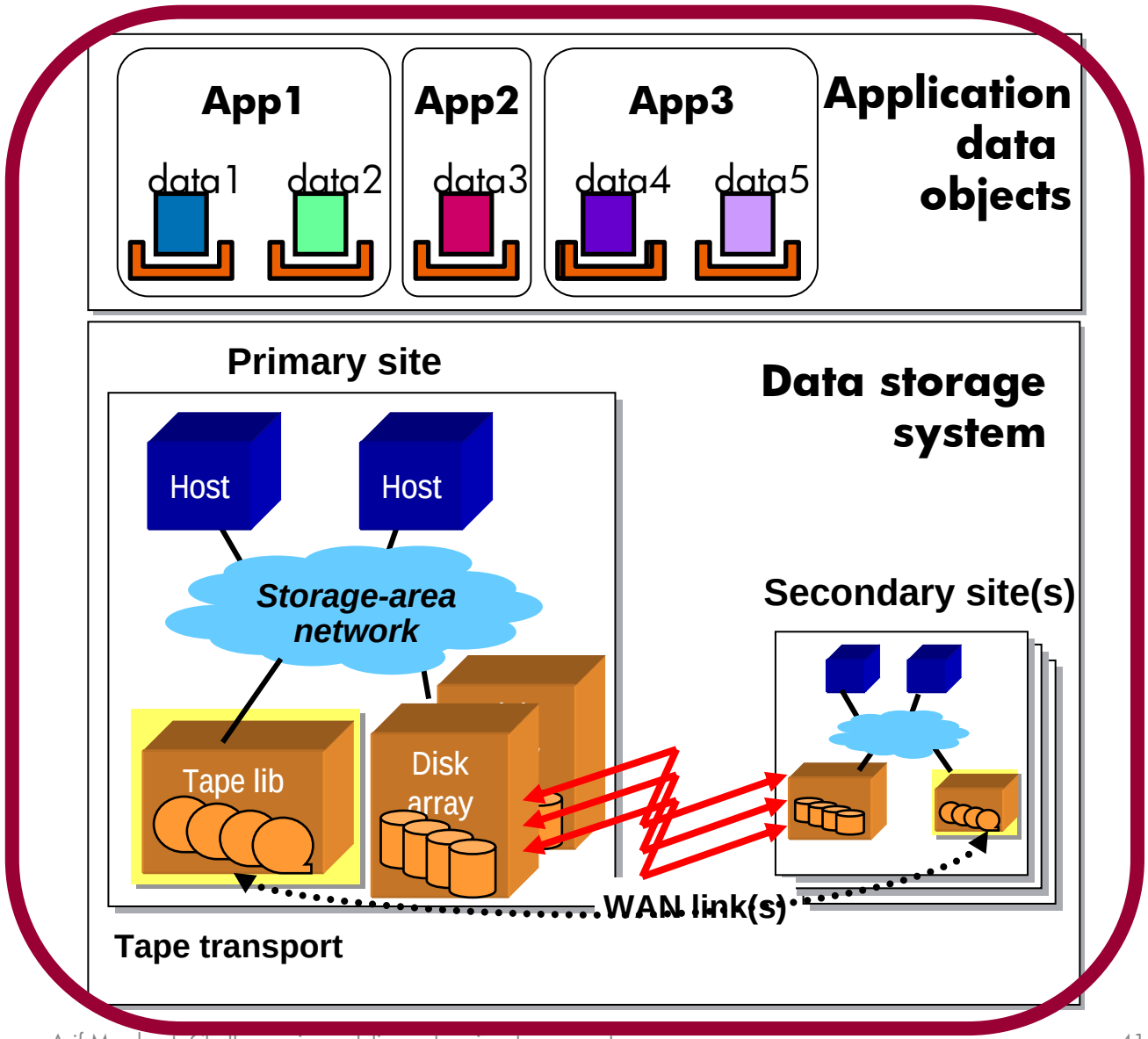
	Array failure			Site disaster		
Storage system design	**RT (hr)**	**DL (hr)**	**Total cost**	**RT (hr)**	**DL (hr)**	**Total cost**
Baseline	2.4	217	$11.94M	26.4	1429	$71.94M
Weekly vault	2.4	217	$11.96M	26.4	253	$14.96M
Weekly vault, F+I	4.0	73	$4.84M	26.4	253	$14.96M
Weekly vault, daily F	2.4	37	$2.98M	26.4	217	$13.18M

- Daily full backups:
  - Further reduce array failure recovery time and data loss
  - Result in shorter vault lag time (and reduced site disaster data loss)

Open issues in modeling the data protection system

- Combined models of primary store and secondary stores
  - Estimation of failure frequencies and types
  - Combining array/data protection performance models
  - Performance and dependability during recovery
  - Handling failures during recovery (e.g., tape is corrupt)
- Using better workload characterizations
  - Current models use simple, static estimates of transfer times
- Validation against real configurations

The whole system



Application/whole system requirements Open problems [Keeton2006]

- Users care about application and business-process performance and dependability
- Modeling system-level implications of storage configuration choices is unsolved
  - Current estimation methods are (mostly) manual
  - Based on experience/intelligent guesswork/rules of thumb
  - Best case: estimates based on measurements & benchmarks

Challenge:

- Can we describe the environment and requirements at the application level, and predict if a configuration will be acceptable?

Summary

- The increasing complexity and cost of storage demands accurate, fast models to support design and management.
- Models must represent real systems, real workloads, *and be well validated*
- Existing models need improvement in many areas: workload models, performance and dependability models of disk arrays and composite system models

There are lots of opportunities.

You can help!

An incomplete bibliography
Articles cited in this talk

- [Alvarez2001] Alvarez et al. *MINERVA: an automated resource provisioning tool for large-scale* storage systems. ACM ToCS 2001.
- [Anderson2001] Anderson. *Simple table-based modeling of storage devices*. Technical Report HPL-SSP-2001-4, HP Labs, 2001.
- [Anderson2002] Anderson et al. *Hippodrome: running rings around storage administration*. In Proceedings of the USENIX Conference on File and Storage Technologies (FAST), January 2002.
- [Ganger1998] Ganger et al. *The DiskSim simulation environment version 1.0 reference manual.* Technical report CSE-TR-358-98. Dept. CSE, University of Michigan, 1998.
- [Gibson1992] Gibson. *Redundant disk arrays: reliable, parallel secondary storage.* PhD Thesis, UC Berkeley, 1992
- [Keeton2004] Keeton and Merchant: *A Framework for Evaluating Storage System Dependability*. DSN 2004.
- [Keeton2006] Keeton and Merchant. *Challenges in managing dependable data systems.* SIGMETRICS Performance Evaluation Review, 2006.
- [Mesnier2006] Mesnier at al. *Relative fitness models for storage.* SIGMETRICS Performance Evaluation Review, 2006.
- [Patterson1988] Patterson et al. *A Case for Redundant Arrays of Inexpensive Disks (RAID).* SIGMOD, 1988.
- [Ruemmler1994] Ruemmler and Wilkes. *An introduction to disk drive modeling. IEEE Computer 1994.*
- [Schindler1999] Schindler and Ganger. *Automated Disk Drive Characterization.* CMU SCS Technical Report CMU-CS-99-176, 1999.
- [Shriver1998] Shriver et al. *An Analytic Behavior Model for Disk Drives with Readahead Caches and Request Reordering. Sigmetrics/Performance 1998.*
- [Uysal2001] Uysal et al. *A modular, analytical throughput model for modern disk arrays.* MASCOTS 2001.

Credits

- Most of the work in this talk is joint with members of the Storage Systems Department, HP Labs
- Slides & graphics from Christopher Hoover, Kim Keeton, Mustafa Uysal, and John Wilkes

Challenges in modeling enterprise storage systems

Thank you!

arif.merchant@hp.com

http://www.hpl.hp.com/research/ssp