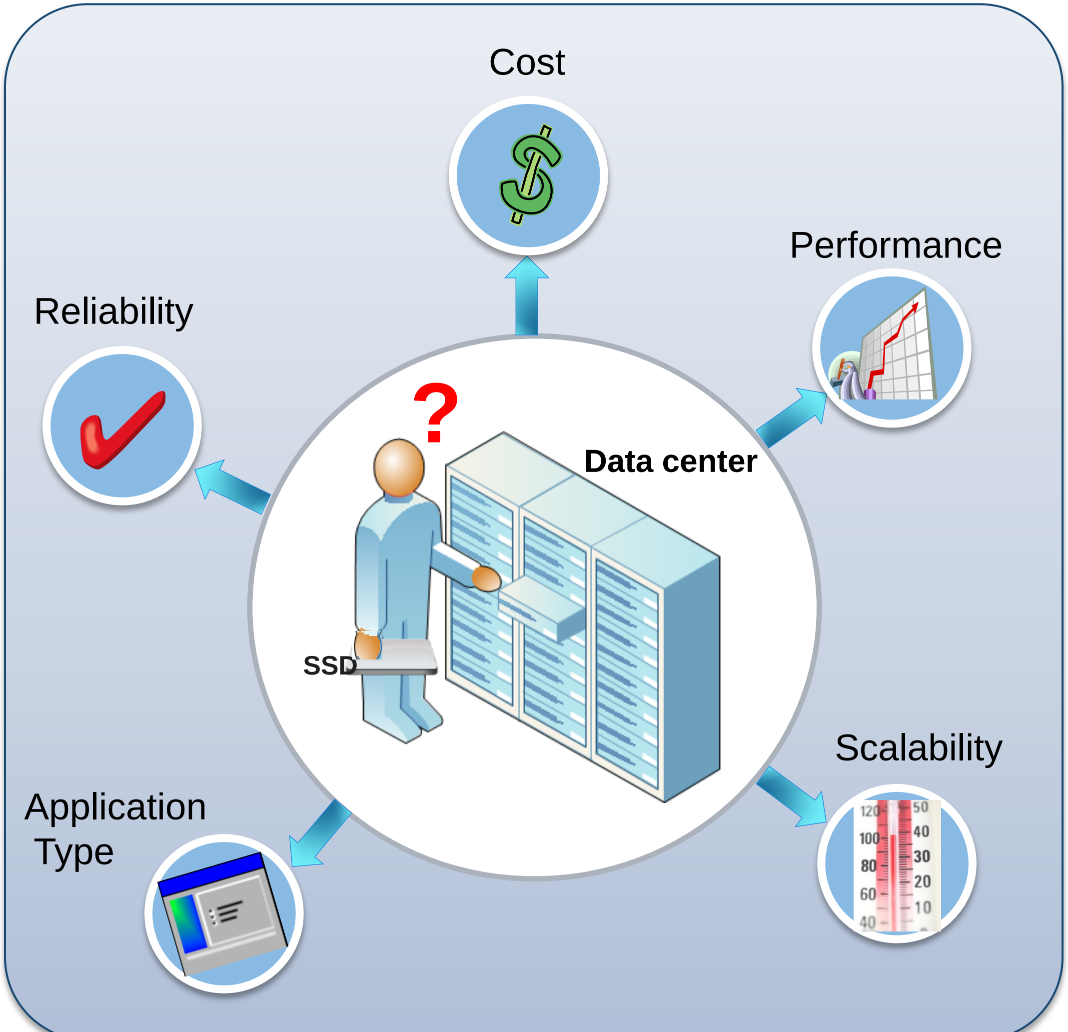
Cache To The Future

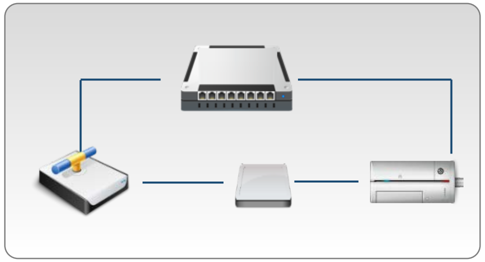
STEC®
The SSD Company™

Introduction

Flash – Factors & Placement



Server



Network

Storage

SSD in Datacenter: Server

Benefits

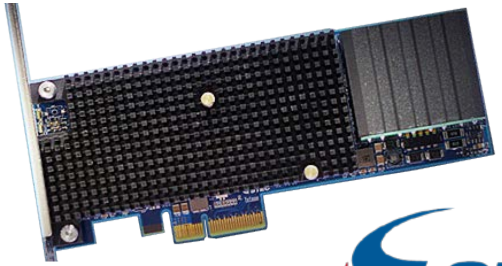
- Close to CPU
- Lower latency
- Higher bandwidth for PCIe form factor

Bottlenecks

- Scalability & sharing
- Complex management
- Data protection

Usage

- SSD as primary storage
- SSD as cache



STEC
The SSD Company™

SSD in Datacenter: Network

Benefits

- Data sharing
- Reliability & redundancy

Bottlenecks

- Network latency
- Higher cost/IOPS

Usage

- SSD as Cache

SSD in Datacenter: Storage

Benefits

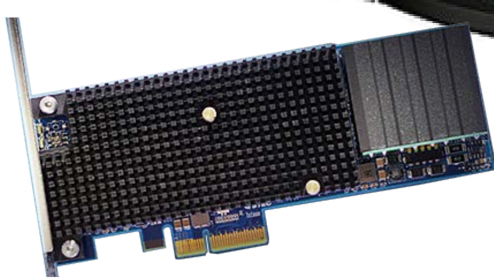
- Data distribution
- Reliability & Redundancy

Bottlenecks

- Higher cost / IOPS
- Network latency
- Protocol overhead

Use case

- SSD as pure storage
- SSD as tier
- SSD as cache



SSD Data Placement Strategy: Primary Storage

Pros

- High performance
- Simplified management
- Efficient data management
- Randomized read/write intensive environment

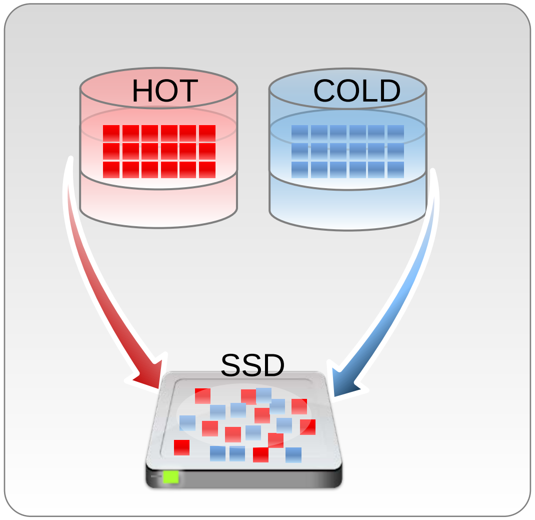
Cons

- Cost

Deployment

- Virtual environment
- Any application with large working dataset
- Limited power & cooling

Primary Storage



SSD: Data Placement Strategy: Tiered Storage

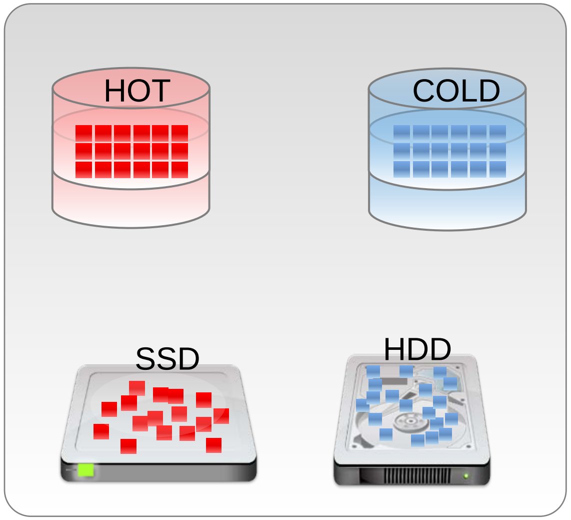
Pros

- Unstructured data
- Cost as compared to Primary storage

Cons

- No second copy of data
- Limited capacity
- Additional Stack
- Frequency of data migration

Tiered Storage



Deployment

- Virtual environment, Heavy traffic applications, Databases (Indexes, temps)

STEC The SSD Company™

SSD Data Placement Strategy: Cached Storage

Pros

- Cost
- Redundant copy of data
- Transparent

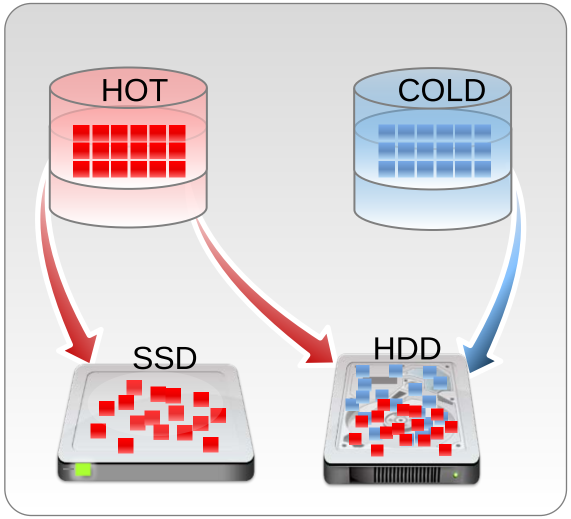
Cons

- Limited capacity
- Write intensive environments
- Additional layer in the stack

Deployment

- Virtual environment, Databases
- Application metadata, frequently accessed files

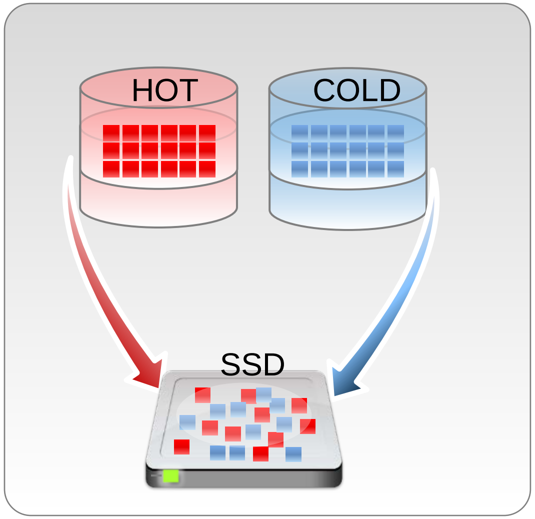
Cache



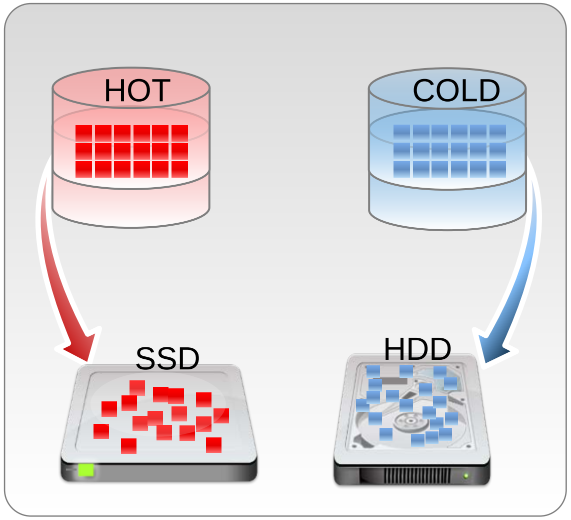
STEC The SSD Company™

SSD Data Placement Strategy: Collective Attributes

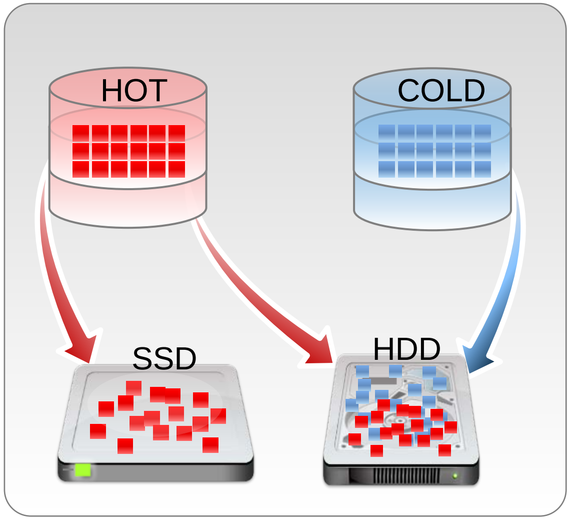
Primary Storage



Tiered Storage



Cache



Primary Storage	Caching	Tiering
All Data	**Copy of frequently accessed data**	**Frequently accessed data**
Failure of SSD is data loss	**Failure of SSD is trivial in operation**	**Failure of SSD means data loss**
Read/Write intensive environment	**Read intensive environment**	**Mixed Read/write, changing data access paterns**
Suitable for larger data sets	**Suitable for smaller slices of data**	**Preferable for mid size data sets**
SLC for performance	**eMLCs for performance**	**SLC for performance**

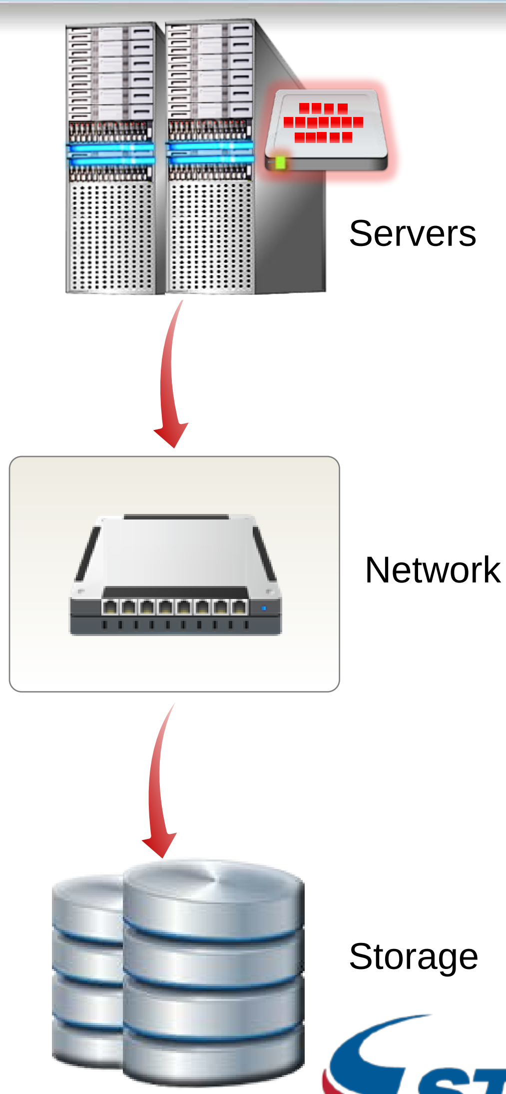
Cache Strategy: Server

Pros

- Closest to the CPU (for PCIe)
- Lowest latency, no network latency
- Highest Throughput (for PCIe)

Cons

- Complexity with clusters
- Dependency on OS
- Data protection



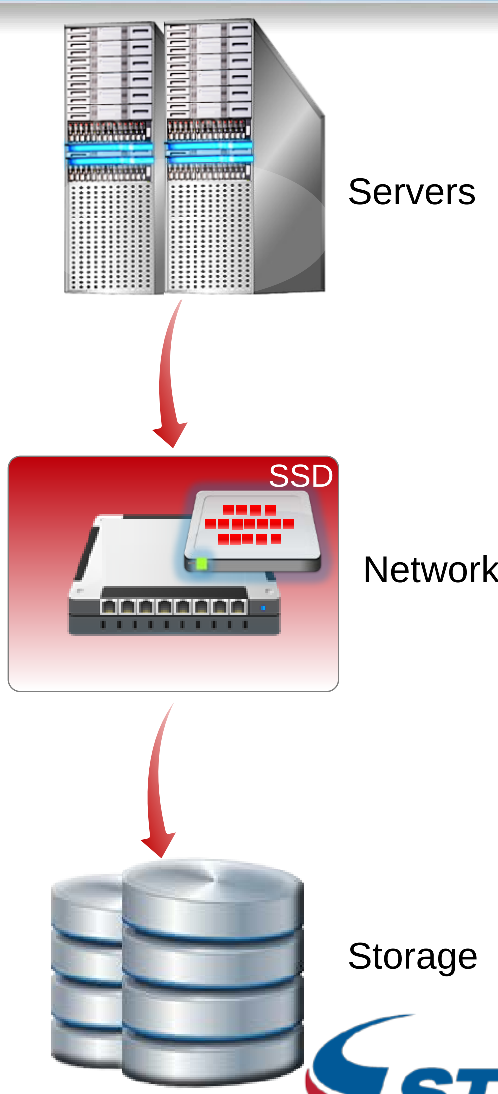
Cache Strategy: Network

Pros

- Well suited for cluster servers
- Manageability
- Data protection

Cons

- Network latency adds up
- Throughput Not as high as Server



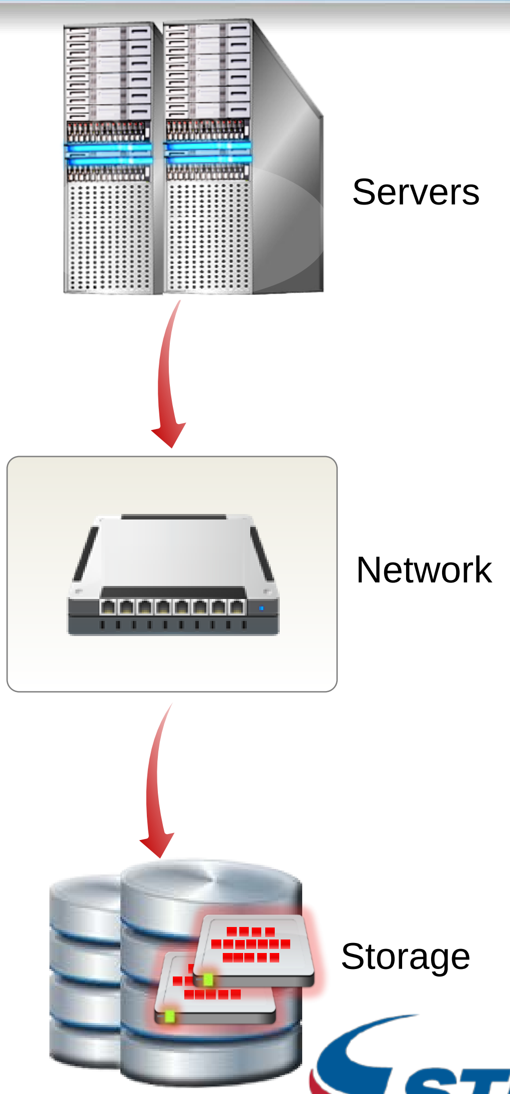
Cache Strategy: Storage Array

Pros

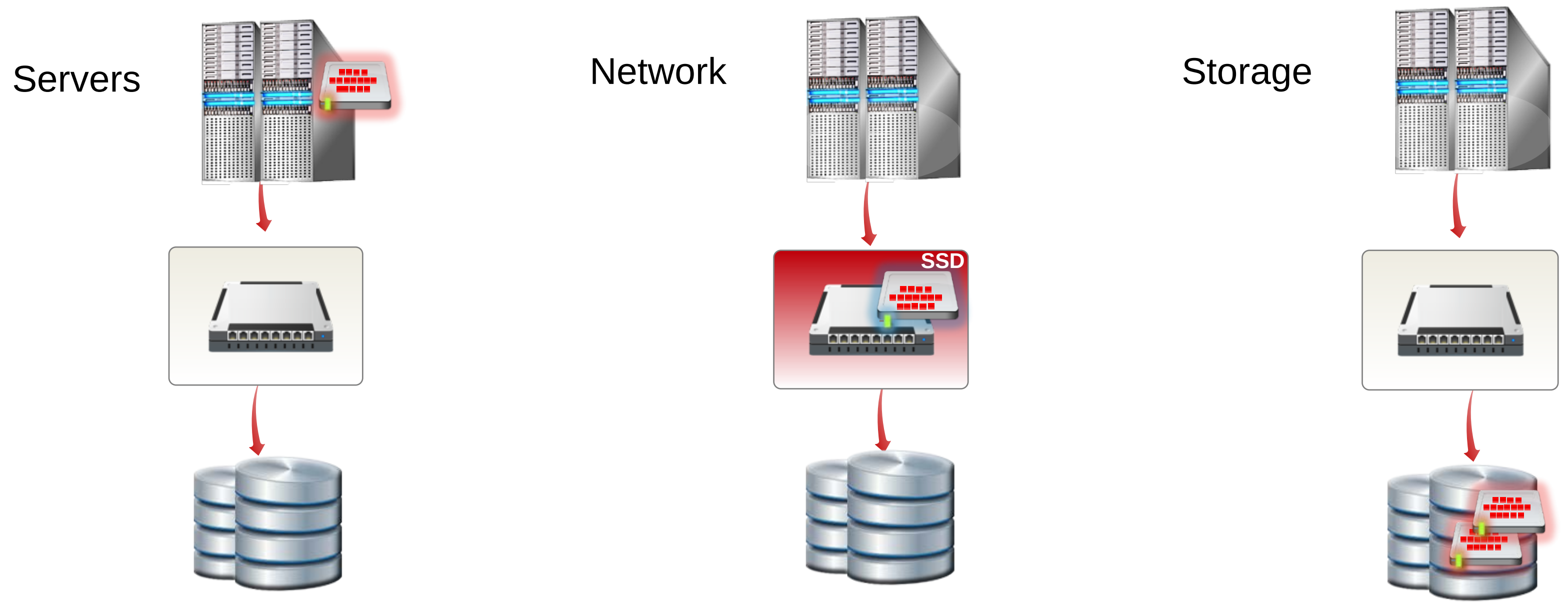
- Ease of scalability
- Works well in Clustered environments
- Data protection

Cons

- Network Latency adds up
- Not as high as server



Cache Strategy: Collective Attributes



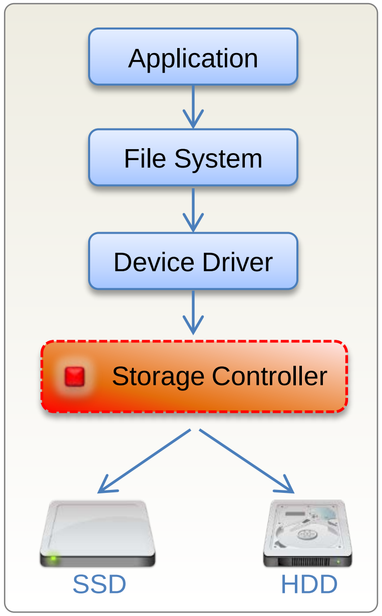
Server	Network	Storage
No network latency	Higher Network latency	Highest Network latency
Cluster complexity	Cluster easily managed	Cluster easily managed
Databases, application metadata	Virtualization, HPC	Virtualization, HPC

STEC The SSD Company™

Server Caching: Storage Controller

Pros

- Independent of OS
- Added redundancy
- Simplified management
- closest to storage in stack
- Cheap



Cons

- Hardware dependency
- OS dependency

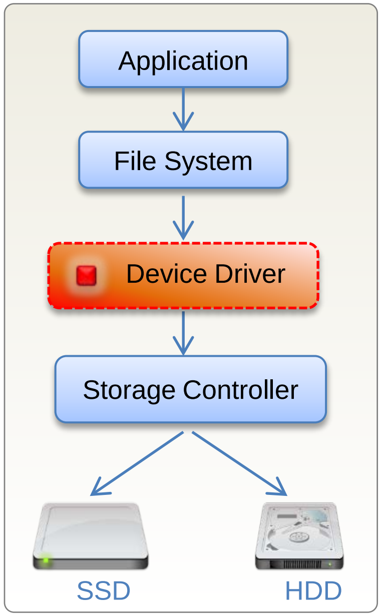
Server Caching: Device Driver

Pros

- Simplified management
- Independent of hardware
- High performance
- Application independent

Cons

- Addition to the storage stack
- Less control



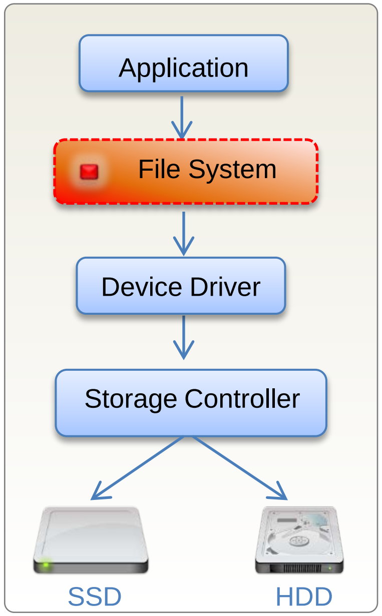
Server Caching: File Sytem

Pros

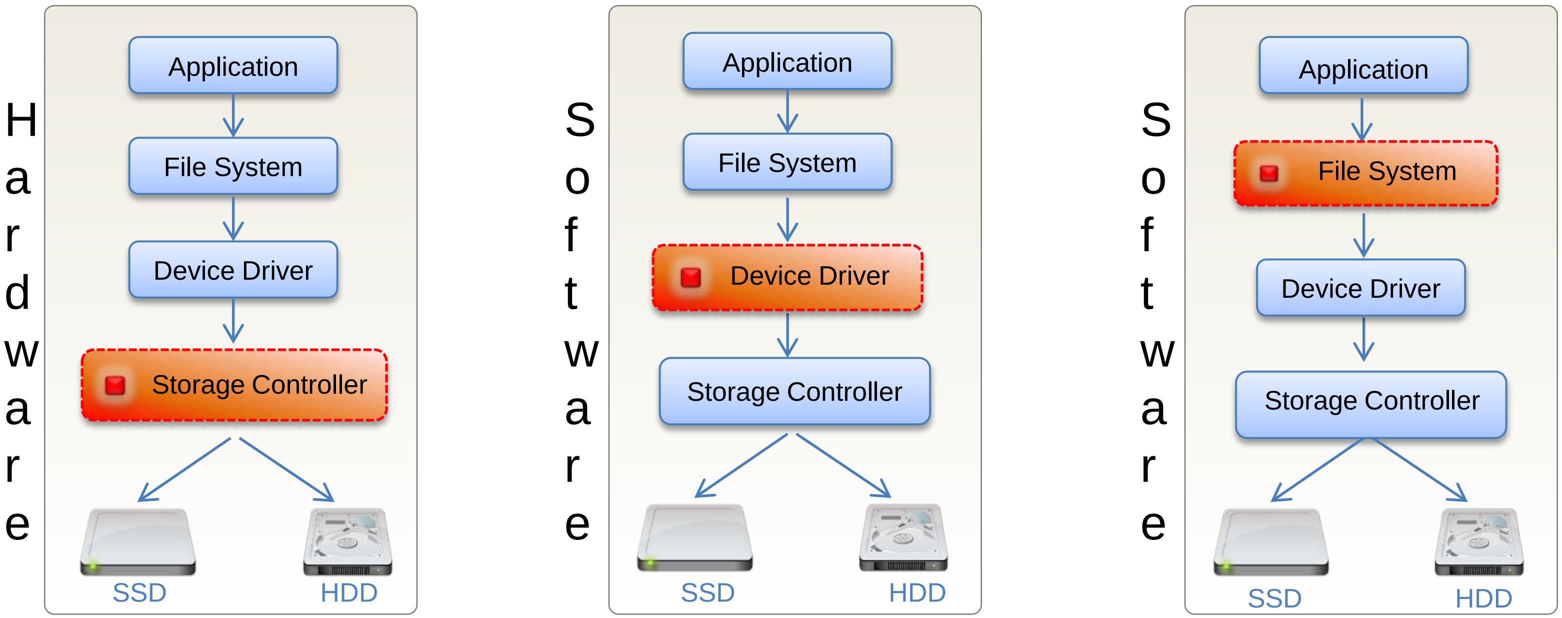
- Independence of hardware
- High performance
- Absolute control over cache

Cons

- OS dependency
- Application dependency
- Addition to the storage stack

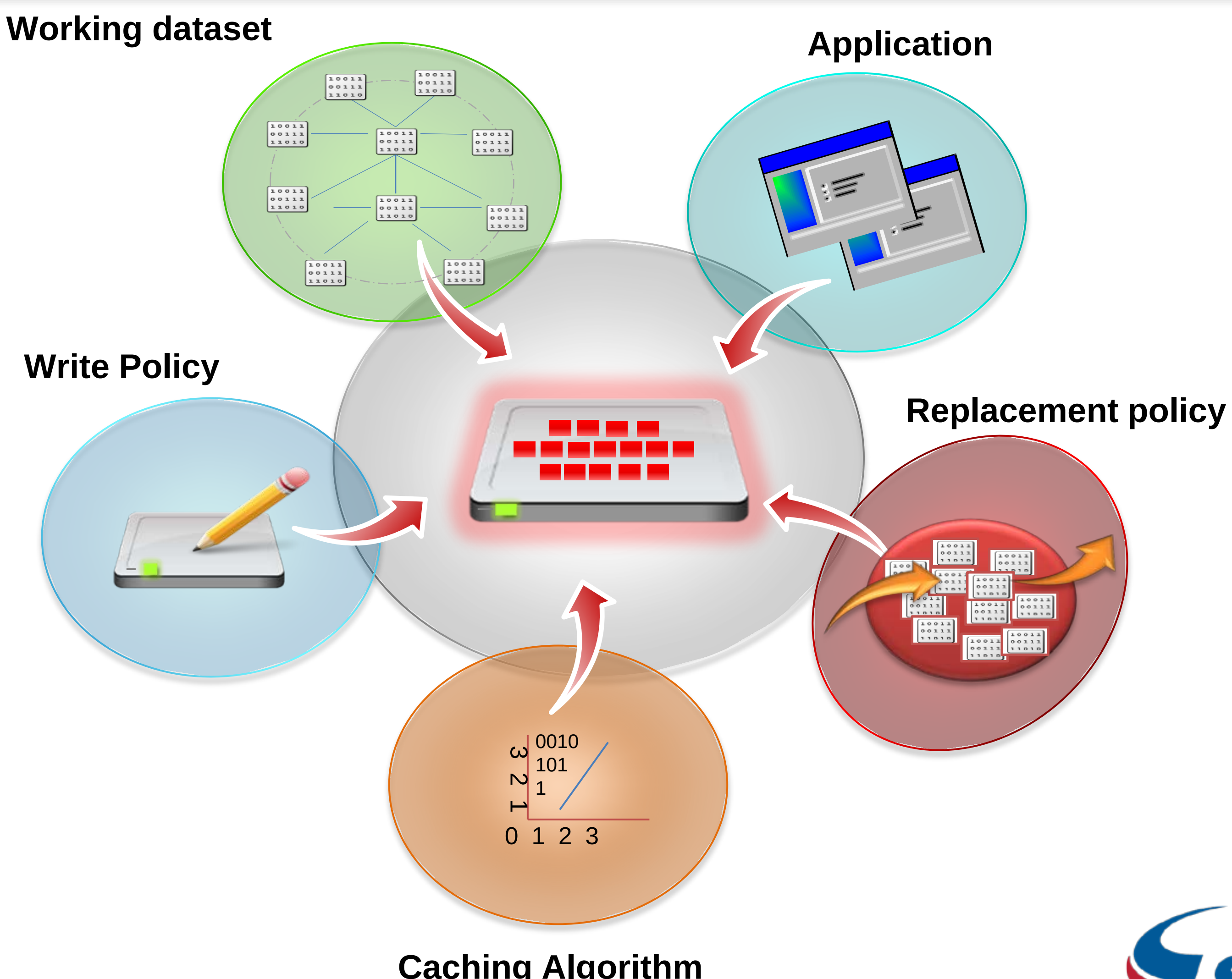


Server Caching: Attributes

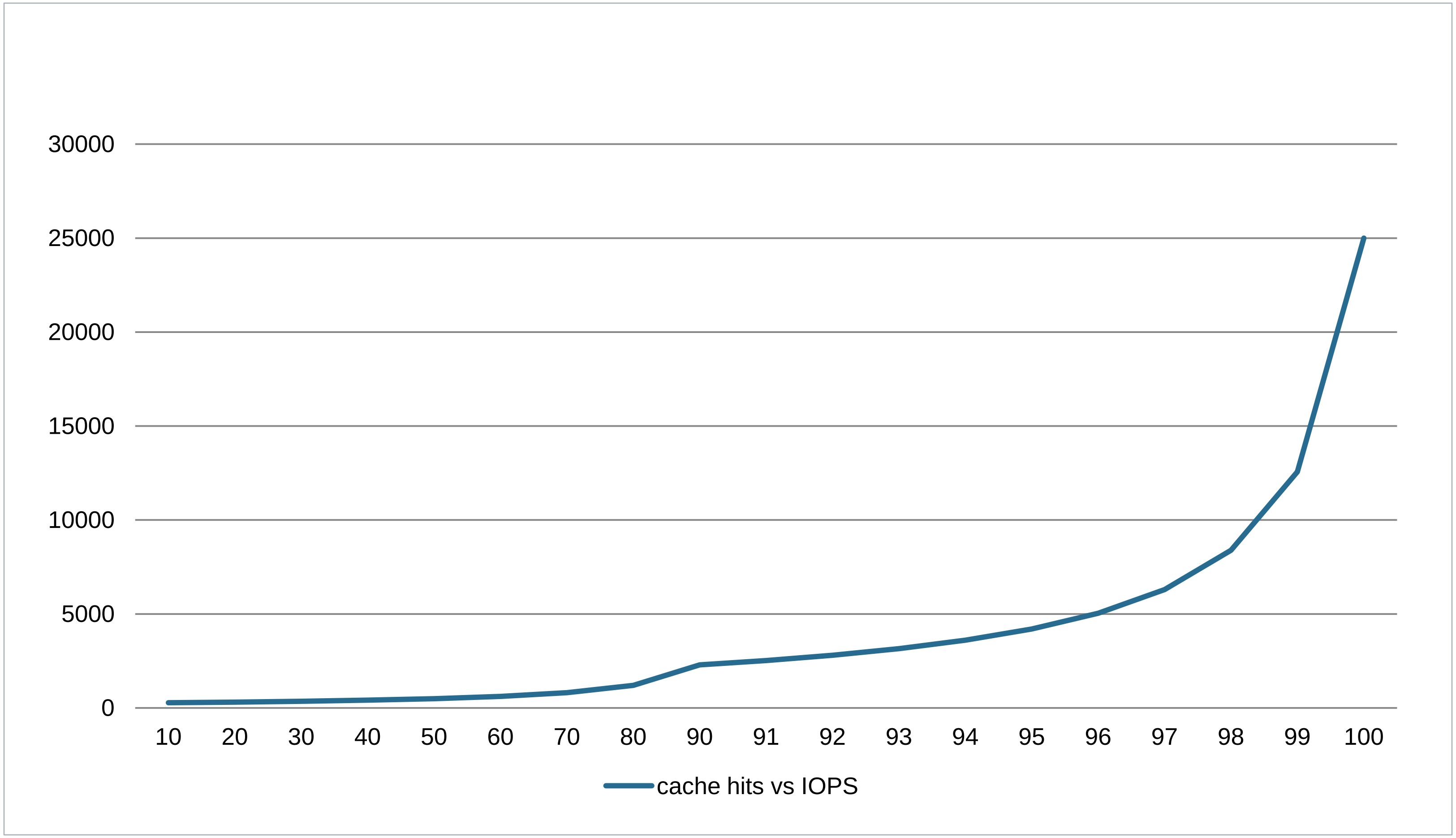


Storage Controller	Device Driver	File System
OS & Application independent	Application independent	OS & Application dependent
Hardware dependency	No hardware dependency	No hardware dependency
Built in hardware write back protection	Use 3rd party integration for write back protection	Use 3rd party integration for write back protection

Cache`nomics: Cache Performance Factors



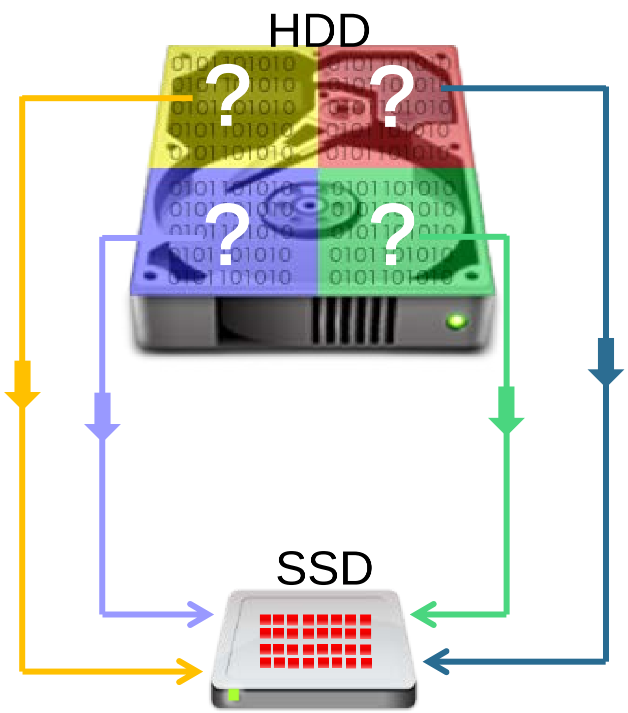
Cache'onomics: The Cache Performance Meter



STEC
The SSD Company™

Cache`nomics: Working Dataset

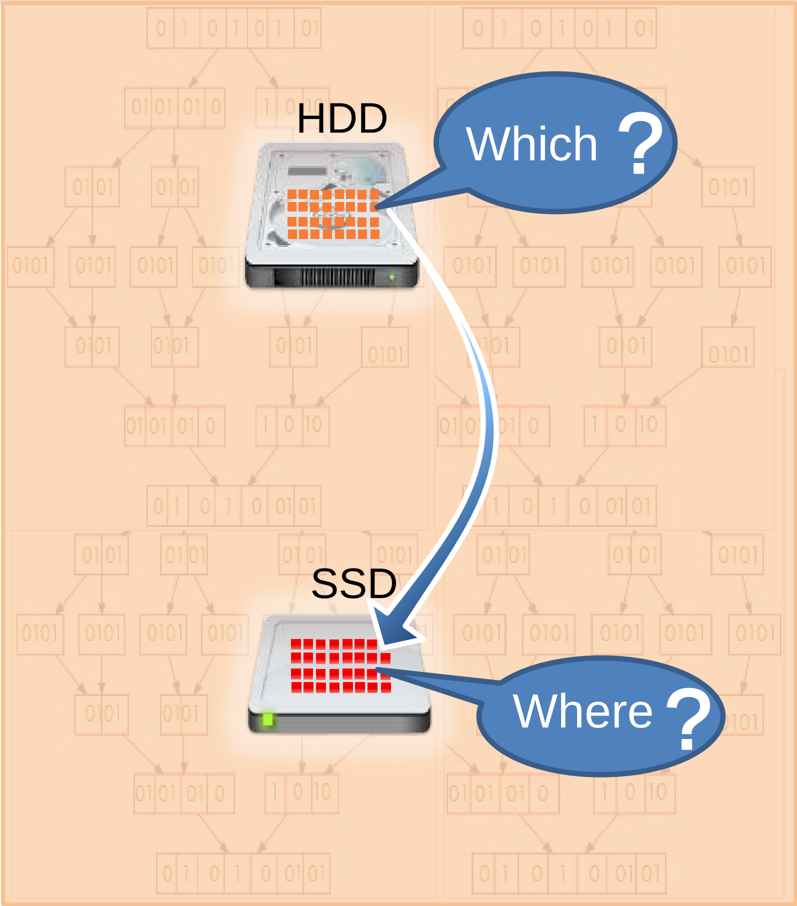
- Identifying working dataset for caching – The right mix
- Predicting working dataset size
- Influences SSD size used for caching
- Impacts cache hits



The SSD Company™

Cache`nomics: Caching Algorithm

- Spatial or Temporal or Proprietary
- Effectively identifying the hot data
- Prefetching intelligence
- Applying compression & dedup
- Identify sequential reads/writes
- Effective place data in cache
- Impacts cache hits and cache size

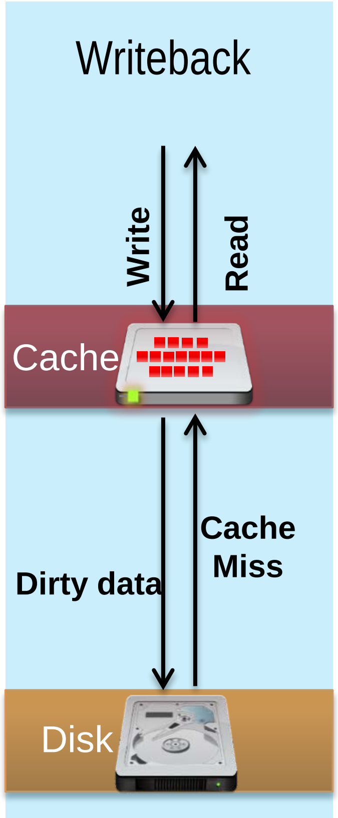


Caching Algorithm

STEC The SSD Company™

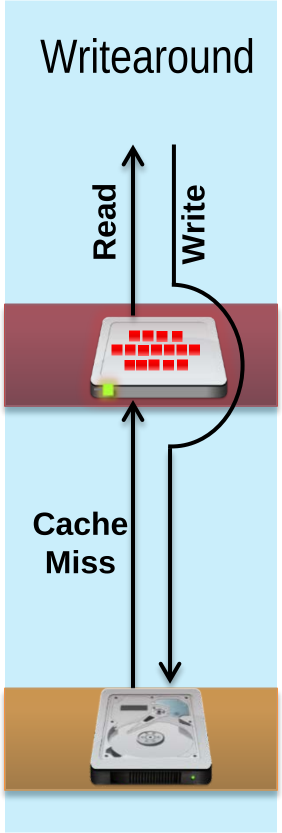
Cache`nomics: Write Back Policy

- Writes to SSD, later copies the ‘dirty data’ to HDD
- Fastest performance for read/write environment
- Data loss possibility
- Implementation: Less risky environments like back end analytics



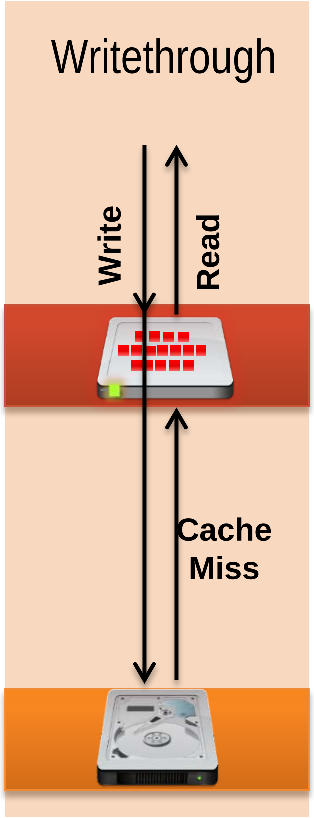
Cache`nomics: Read Only Policy

- No writes to SSD
- No data loss
- Best benefit in read intensive environment
- Databases, read intensive work environments



Cache`nomics: Write Through Policy

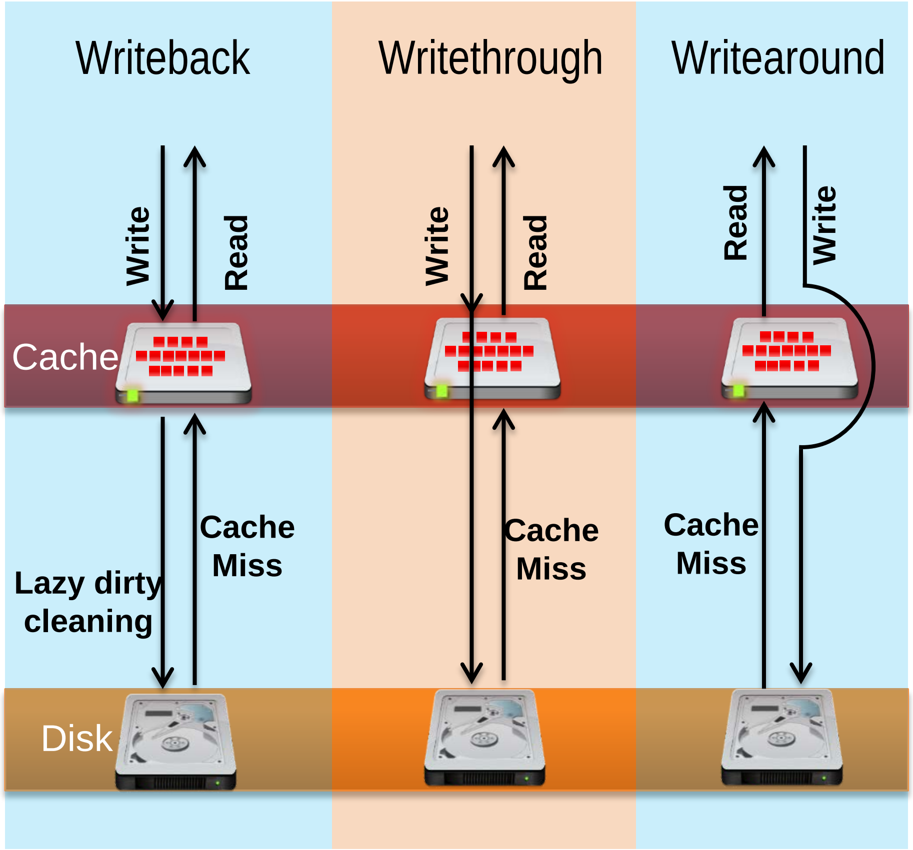
- Writes on SSD & HDD simultaneously
- No reliability issues
- Read/Write environment makes it less effective though more effective than read only
- Preferred over read only in similar environments with a mix of writes as well
- Good for data analytics,



STEC The SSD Company™

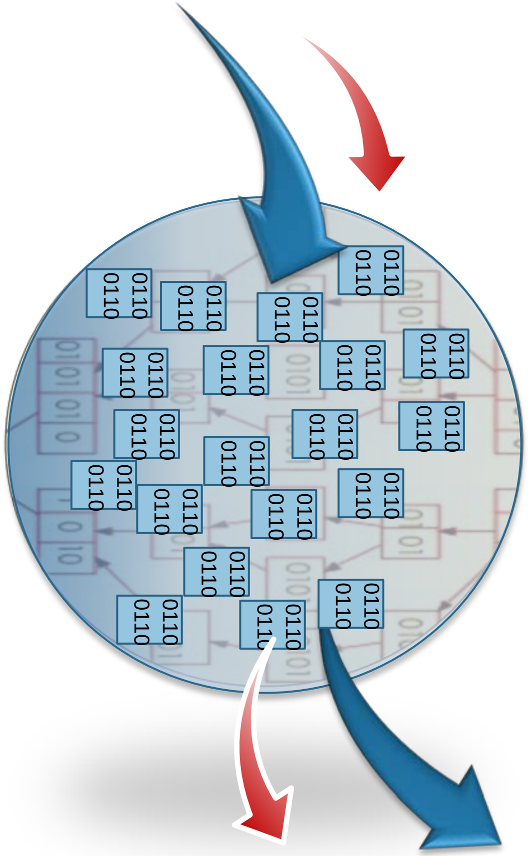
Cache`nomics : Write Policies Attributes

Read Only/Write around	Write –Through	Write-Back
No writes to SSD	Writes to SSD & HDD simultaneously	Writesto SSD, later copies to HDD
No data loss issues	No loss issues	Data loss possibility
Read intensive environment	Read/Write environment makes it less effective though more effective than read only	Read/Write environment, best option to go with
Databases, read intensive work environments	Preferred over read only in similar environments with a slight mix of writes as well	Less risky environments like back end, analytics can use this mode



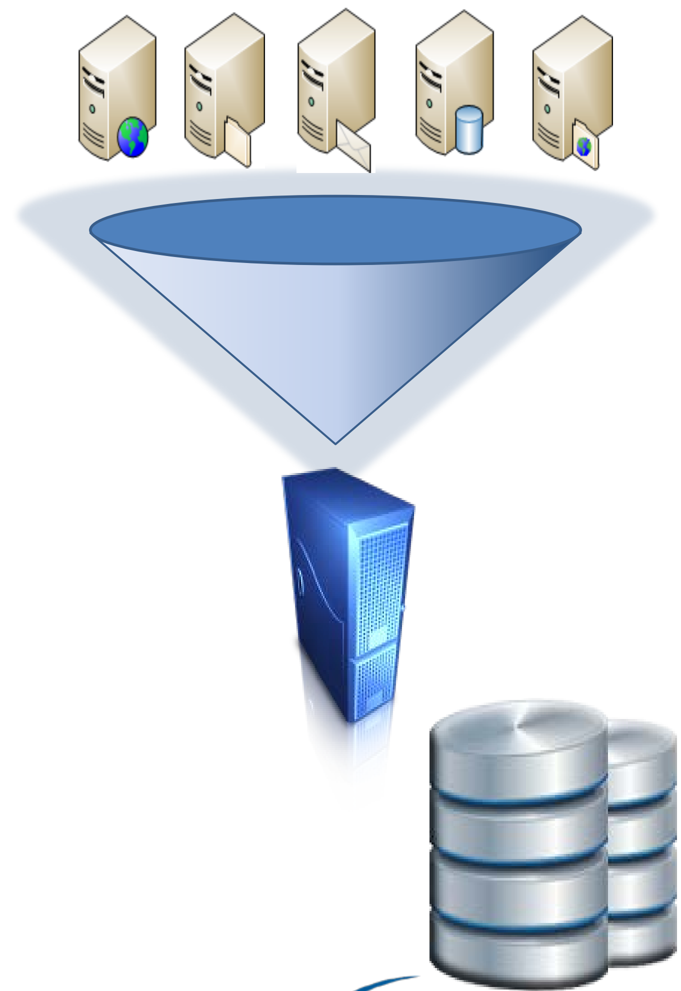
Cache`nomics: Replacement Policy and Block Size

- LRU,FIFO, Random, ARC, Proprietary
- Full cache block eviction accuracy
- Size of cache block based on working dataset
  - Bigger block size results in wasting diskspace
  - Smaller block size results in poor performance



Cache`nomics: Applications

- Vertical
  - OLTP, OLAP, Data Warehousing, Data Mining
- Horizontal
  - Database (Oracle, SQL, MySQL)
    - Transaction logs, tables, indexes
  - Virtualization
    - VDI, VM apps
  - Enterprise applications
    - Metadata



STEC The SSD Company™

Cache Solutions: Recent Trends

- Predictor to suggest the customer
  - Working dataset
  - Replacement policy
  - Block size
  - Write policy

- Application tweaking and best practices efforts

Summary

- Multiple plug-ins for SSD in enterprise
- Deployment strategies revolve around the key factors
- Caching solutions have evolved and keeps evolving
  - From array to server, trying to see as many possibilities as possible
- SSD cache aware application trends (Oracle for db example, ZFS possibly for File System (not sure on ZFS))

STEC
The SSD Company™