*TOMORROW starts here.*

CISCO

Cisco live!

# Introduction to Solid State Storage Systems

BRKCOM-1601

James Candelaria
Director of Engineering
Solid State Systems Group

Cisco live!

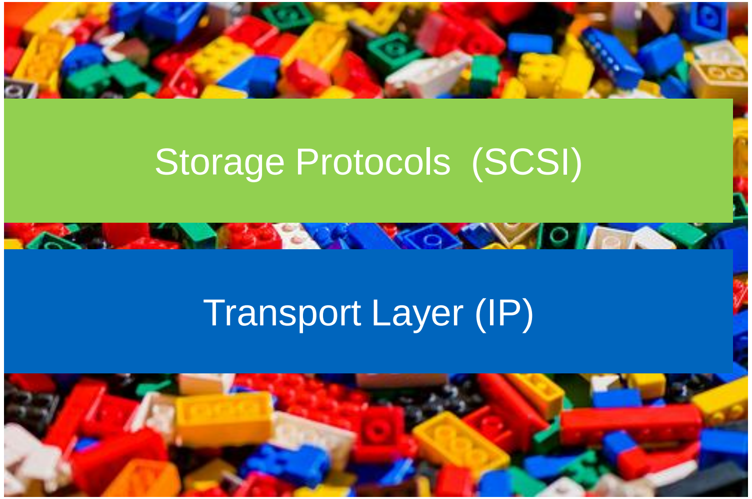
# Session Agenda

- History and Background
  - Block Storage Connectivity
  - Comparisons
  - NAND Overview
- Managing RAW Media deficiencies
- Implementation Techniques
- Use Cases

# History and Background

# Blocks and Protocols

- Protocols
  - SCSI
  - ATA
- Transports
  - Parallel
  - Serial
  - Fibre Channel
  - IP
- Blocks
  - Lots of sizes
  - 512 Bytes +

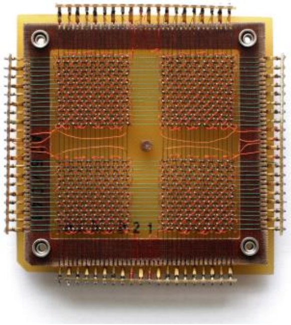


# Server I/O Systems

- PCI/PCI-x
  - Shared Parallel Bus
  - PCI - 32 Bit 33/66Mhz, 64 Bit 33/66Mhz
  - PCI-x – 64 Bit 66-533Mhz
- PCIe
  - Serial, point-to-point
  - 128b/130b (PCIe 3) encoding
  - 8GT/s per lane ~980 MB/s

# Solid State Memory History

- Solid state memory isn't new
  - Core memory (1955-1975)
    - Polarised magnetic rings
    - Poor density w/ destructive READ
  - RAM (SRAM/DRAM) (BC-> Present)
    - Fast
    - Volatile
    - Poor density
  - NOR/NAND Flash (1980-> Present)
    - High density (NAND), Fast READ
    - Poor endurance, poor write performance, error prone (NAND)



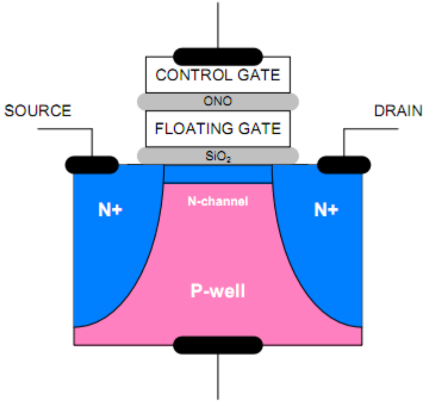
# Comparison with Hard Disk Tech

- Hard disks have poor *random* performance, due to head movement
  - 15,000 RPM 200 IOPS / drive
  - 10,000 RPM 150 IOPS / drive
  - 7,200 RPM 100 IOPS / drive

- Solid state memory has excellent random READ performance
  - SSD 50,000 IOPS / drive

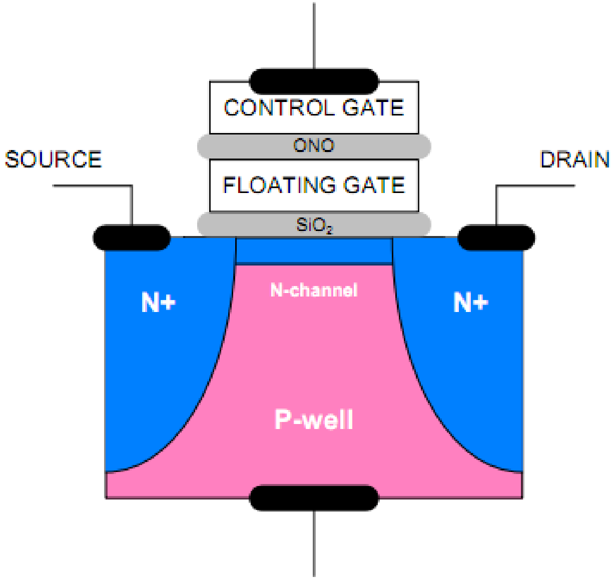
# Basic NAND Cell Construction

- NAND cell stores data inside of a floating gate
- Strings of cells are organised together to form a “page”
- Pages are grouped together to create a block
- Pages include spare area for error correction
- Pages are independently programmable, but not independently erasable



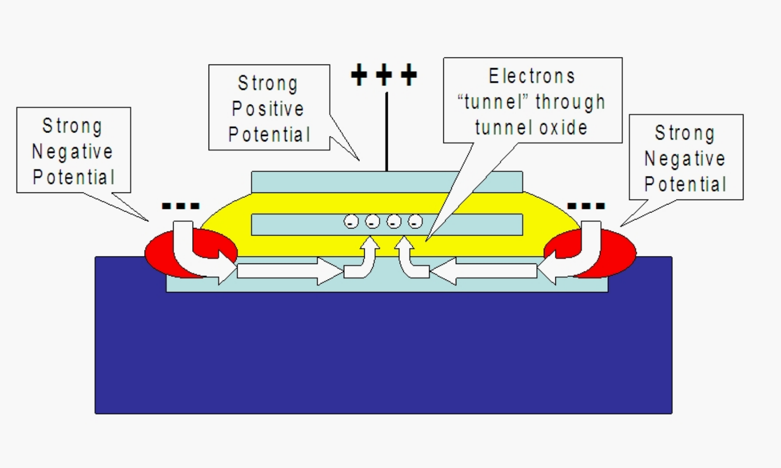
# Reading Data

- Current is applied to the control gate, depending on the potential of the floating gate, current may or may not flow reflecting a binary value.
- Unlike DRAM, READ is non destructive and does not disrupt the charge of the floating gate*
- *Repetitive reading of cells in a block may introduce errors in surrounding blocks known as “read-disturb”



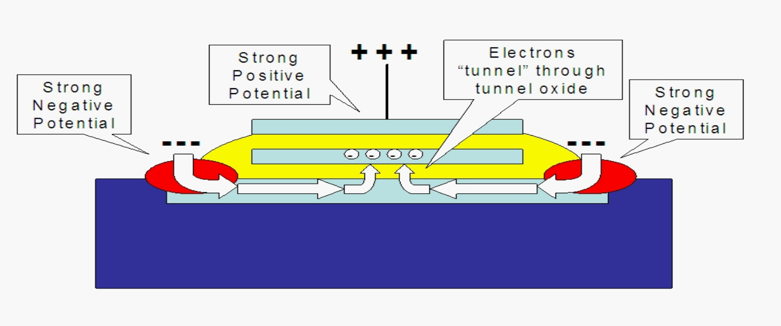
# Cell Programming / Erase

- NAND cells are programmed and erased by placing a high voltage between the source, drain, and control gate forcing electrons through the oxide layer and into or off the floating gate.
- Programming and erasing cells is a destructive process: silicon doping, weakening of the insulating layer, etc
  - Endurance gets worse as the cell shrinks due to fewer electrons in the floating gate smaller insulators, and other cells in closer proximity


Strong Negative Potential
Strong Positive Potential
+++
Electrons "tunnel" through tunnel oxide
Strong Negative Potential
---
---

# NAND Cell Types

- SLC – Single Level Cell (1 bit per cell)
  - High endurance (30,000 cycles @ 20nm)
  - High speed
  - High cost per bit
- MLC – Multi Level Cell (2 bits per cell)
  - Lower endurance (3,000 cycles @ 20nm)
  - Moderate speed
  - Much lower cost per bit
- eMLC – Enterprise MLC (2 bits per cell)
  - Moderate endurance (10,000 cycles)
  - Slower program speed than SLC
  - Higher cost per bit than MLC


Strong Negative Potential
Strong Positive Potential
+ + +
Electrons "tunnel" through tunnel oxide
Strong Negative Potential
- - -
- - -

- TLC – Triple Level Cell
  - LOW endurance (500-1500 cycle)
  - Slower program time
  - Ultra low cost per bit

# Managing the Deficiencies of the RAW Media

# Controllers

- Raw NAND unsuitable for direct consumption due to error rates / programming issues
- NAND controllers provide:
  - FTL
    - Wear leveling
    - Garbage collection
    - Error detection / correction
    - Bad block management
  - Write amplification management
    - OP
    - Deduplication / Compression
  - Host interface
    - PCIe, SATA, SAS, NVMe, etc

# Flash Translation Layer

- Provides sector to NAND page translation
  - Block device (SATA/SAS/NVMe) is addressed in LBA
  - FTL converts LBA to NAND location
- Tracks used and “free” pages / blocks
  - Blocks can be freed from host issuing TRIM or overwrite
- Typically runs on embedded processor on
  - Limited system resources
    - Low clock frequency
    - Limited DRAM
- FTL typically log structured
  - Data is written only to free pages

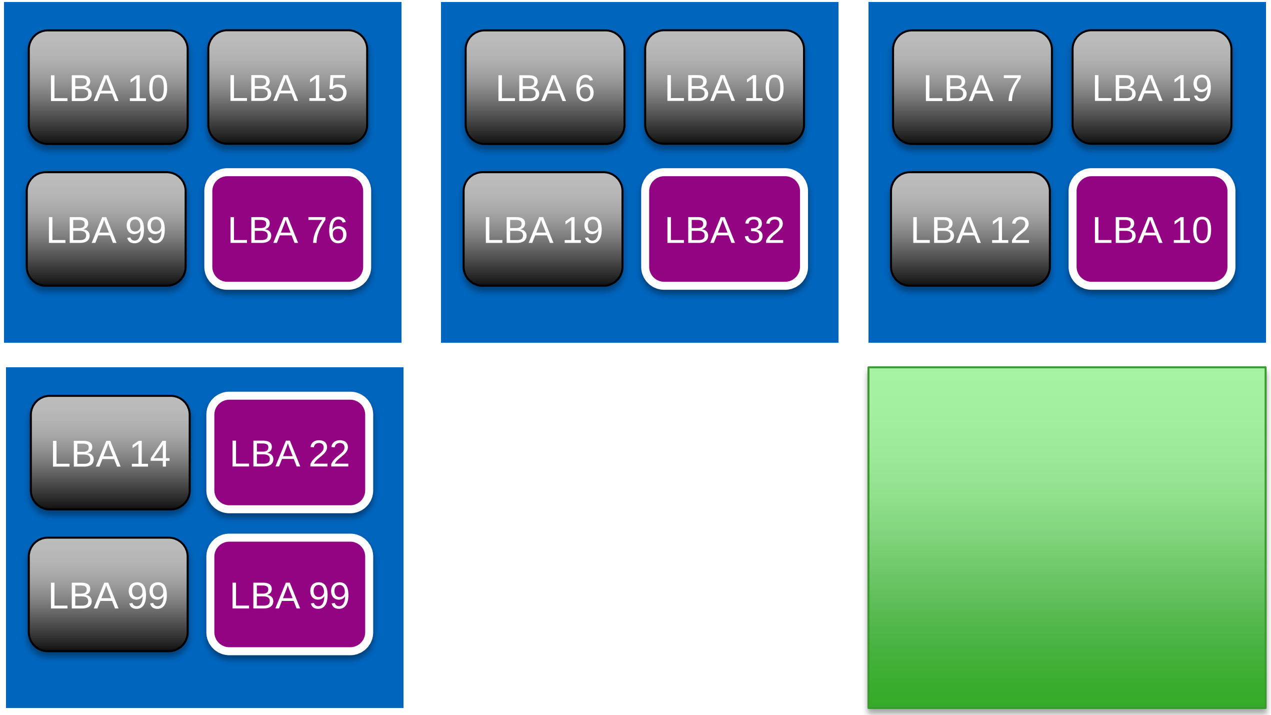
# FTL – Wear Leveling

- Individual flash cells have specific endurance
- To decrease failure controller and FTL can use both dynamic and static wear leveling
  - Dynamic
    - LBA overwrite results in flash page invalidation, new write in free page with lowest age
    - Old block age considered during garbage collection and new writes
    - Static data (system files, old documents, etc) NEVER eligible for wear leveling
    - Simple
    - In-line operation
  - Static
    - Controller moves pages with static data to new locations after thresholds are met
    - Complex
    - Background operation

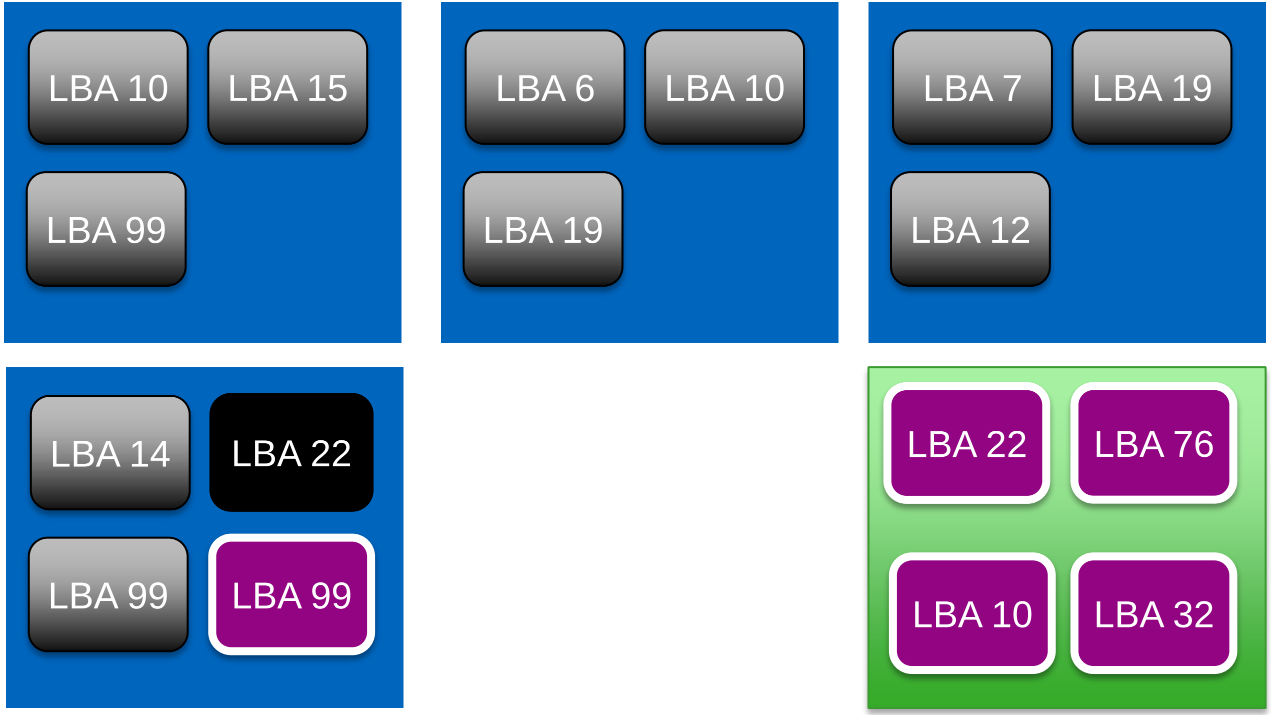
# FTL – Garbage Collection

- Dynamic wear leveling leaves stale pages/blocks on device
- Eventually all blocks/pages are in use
- GC collects all valid pages and consolidates
  - Freeing blocks for erasure
- Controller can pre-erase blocks after GC
  - Enhance performance
- Background process
  - Enhanced by TRIM but not required

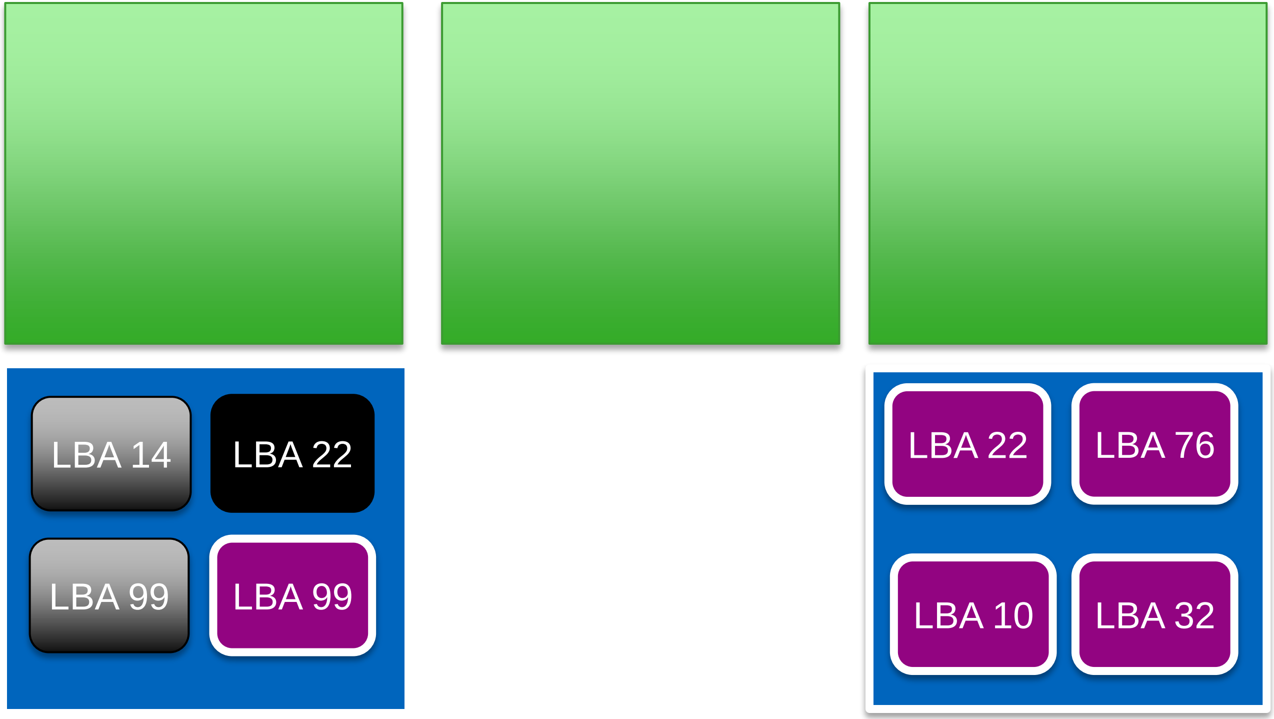
# Pre GC Example “Dirty Media”



# Post GC Move


LBA 10
LBA 15
LBA 99
LBA 6
LBA 10
LBA 19
LBA 7
LBA 19
LBA 12
LBA 14
LBA 22
LBA 99
LBA 99
LBA 22
LBA 76
LBA 10
LBA 32

# Post GC Move Pre-erase



# Controller – Error Detection

- NAND is error prone
  - Programming errors raise vT
  - Read distrub
  - Noisy bus
- Two ratings
  - rBER
  - uBER
- Controllers reserve spare area on page for ECC
  - BCH / LDCP
    - Possible but unlikely undetected errors
  - Correction performed when computed hash doesn't't match hash in spare area

# Controller – Error Correction

- Correction performed when computed hash doesn't match hash in spare area
  - Read retry with different vT settings
    - Requires deep knowledge of NAND properties
  - Traditional correction with BCH/LDCP
  - Combination of techniques

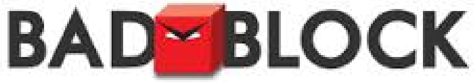
- Advanced error correction can increase usable PE count of blocks 10x
  - DSP technology

# Controller – Bad Block Management

- Blocks past endurance limits may not erase completely

- Blocks with high rBER may be unreliable

- Controller MUST track above at BLOCK level
  - Non fully erased blocks may continue to be used with error correction
  - High rBER cells must be retired or resort to higher levels of ECC



# Controller – Write Amplification Mgmt

- When NAND is FULL, new writes require an ERASE of a block.
  - Small writes (4-8K) can result in MB of writes

- Write coalescing
  - DRAM caches can aggregate inbound writes to avoid small write commits
    - Delays the issue
  - Compression and dedupe
    - Reduces the amount of data on media

- Intelligent GC
  - Maintains enough free blocks to avoid forced erase

# Controller – Host Interface

Controllers must provide an interface to the NAND

- PCIe
  - Proprietary
    - Requires custom driver for every OS
    - Controller functions can be offloaded into driver software
  - SATA/SAS bridge
    - PCIe card uses SAS HBA chip
  - NVMe
    - Elimination of SCSI as transport layer for decreased latency
- SATA/SAS
  - Highest compatibility
  - Straightforward integration
  - Performance compromise*

# RAID

- Enterprise systems require data availability (DAS included)
  - Mirroring (RAID 1 & 10)
    - Expensive
    - Effective with HDD due to low cost per bit and offers excellent READ and good write perf
  - Parity based (RAID 4,5,6)
    - Severe performance compromise on write
    - Much less expensive ($/GB)
    - Exasperates write amplification on flash

# Media Management & RAID Mitigation

- Despite good controller design, random write performance and write amplification is a challenge for the best controller.  Best case performance can be achieved with
  - Sequential fixed length IO
  - Never writing less than an entire erase block
- Parity based RAID requires full stripe writes to avoid the read/modify write issue
  - 24 drive RAID6 can require 21 reads and 3 writes for a short stripe write
  - Solution: Always write full stripes

# Flash Implementation Approaches

# Flash as a Cache

- Large NAND read cache in traditional array
  - Acceleration of "second read"
    - Inconsistent performance for enterprise apps
  - Pushes limits of controllers
    - Media can service more IO than host CPUs can handle interrupts
  - Solves only half (or less) of the problem

- Read + writeback cache
  - Challenges with NAND as described earlier require intelligent design
    - Typically front end many TB or PB
      - Must absorb EVERY write
      - Requires high endurance SLC
  - Typically limited in size
    - Over-run of cache flushes at media rate
      - Destage is the bottleneck, cache often full due to backing media perf

# Flash Array Design Concerns

- CPU / Memory requirements differ from traditional array
  - Flash management techniques required
  - Interrupt bound performance
- Advanced services
  - Deduplication
  - Compression
- Scalability
  - Difficult to exploit all aspects of flash
- Cost management
  - Parity based RAID?

# The Cisco Invicta Approach

# Introducing the Cisco  UCS Invicta Series

## UCS Invicta Scaling System

Using Invicta OS 5.1.0

First release:

Up to 1.3 Million IOPS
Up to 13.2 GB/s Bandwidth
Up to 240TB RAW

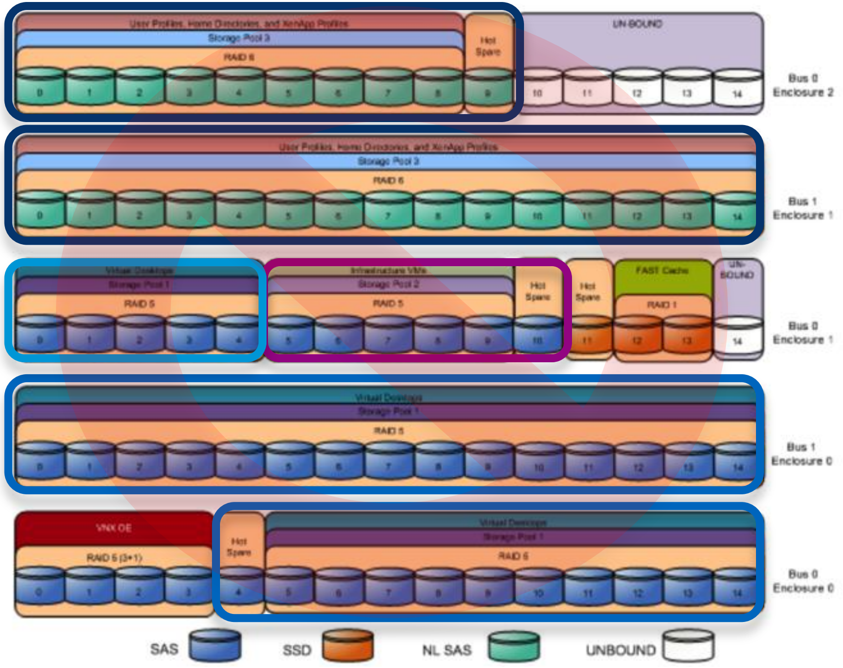
## UCS Invicta Appliance

First release:

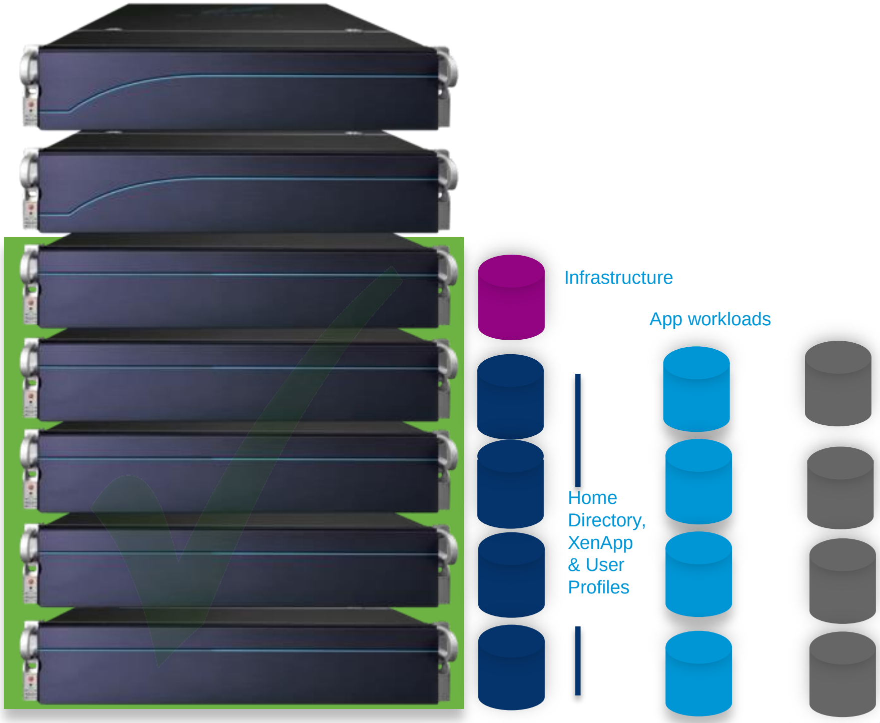
250,000 IOPS
1.6 GB/s Bandwidth
Up to 24 TB RAW

- ✓ Scalability
- ✓ Modularity
- ✓ Application Acceleration
- ✓ Data Optimisation
- ✓ Multiple Workloads
- ✓ Tuning-Free Performance

# Simplify Your Life with Flash



**114,950 IOPS 141TB**



**825,000 IOPS 150 TB**

# UCS INVICTA Series OS

Leverage the advantages of flash while mitigating the issues

## Receive

- Data blocks of Various size arrive from Hosts from network interconnects

## Protect

- Data Blocks are stored in the power loss buffer and passed onto the Block Translation Layer (BTL)

## Optimise

- The Block Translation Layer Aggregates and sizes  Data Blocks  for the RAID Layer and Flash Media
- Hash tags are created for each block

## Commit

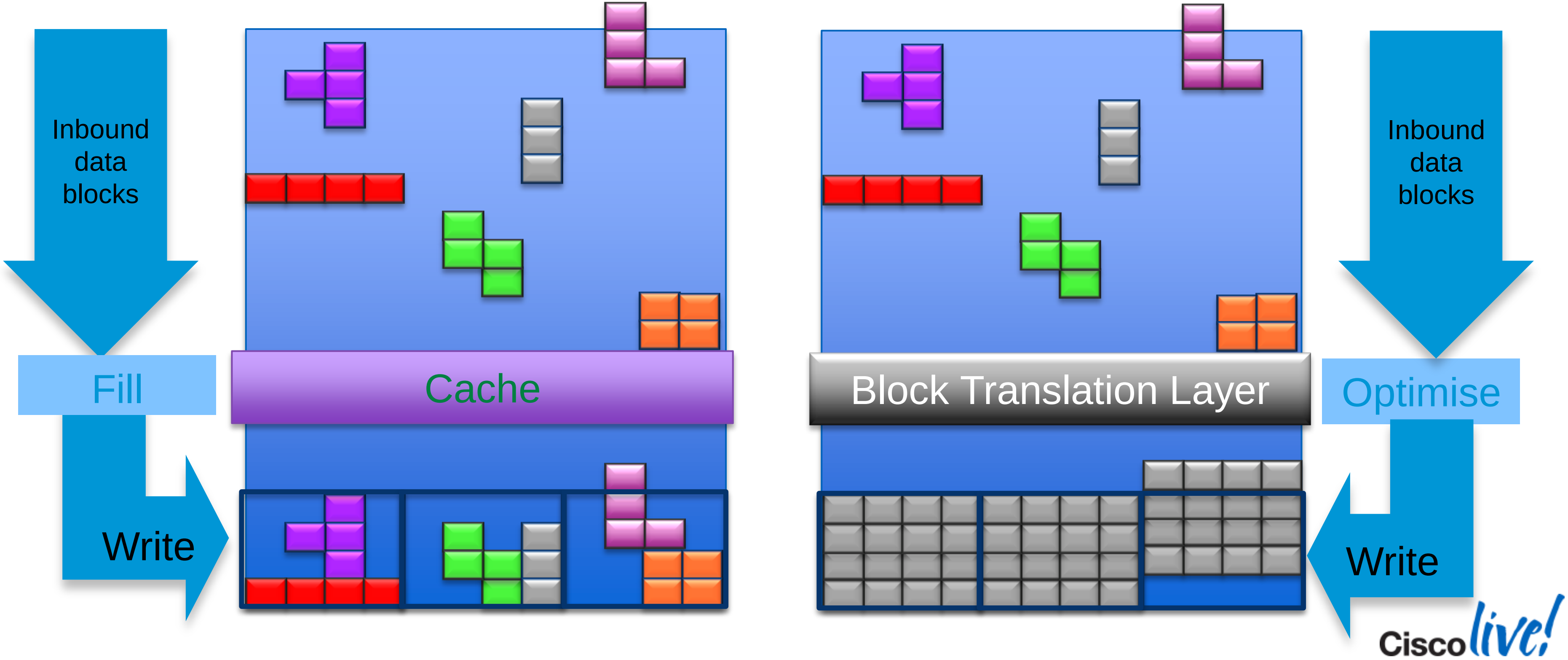
- BTL Optimised Data is flushed across the RAID stripe and Flash erase blocks  concurrently

# Block Translation Layer

- Log structured write IO
  - Long linear chains
- Erase block aligned
  - Large sequential IO to RAID layer
  - RAID chunks combine at device NCQ buffer
- Data integrity validation
  - Every 4K PBA can be validated
  - Granular stripe level repair
- Non-Volatile staging
  - Super capacitor backed DRAM
- Backed by high horsepower CPU and memory
  - 10 core x86 + GB of DRAM
- Data reduction

# UCS INVICTA Series OS

## Optimisation = High Performance



# Invicta Scale Up and Out

- Flash management implemented at node level
  - Never bottlenecked
  - Consistent linear performance

- Fabric services implemented at “router” level
  - Multiple routers can provide fabric attach, availability and data-services

# Use Cases Today

# VDI

- Virtual desktops are HARD to service for traditional storage
  - Thousands of users
  - Write dominant
  - IO blender
  - Extremely burst sensitive
- Sessions need to be considered “interactive”
  - Every delay hurts user experience
  - Users already gave up device control
    - Wont tolerate unpredictable performance
- IOPs and latency is king

# Traditional SQL

## SQL Server, Oracle etc

- DBAs are used to “tuning for storage”
  - Required due to lack of “performance by default” with hard disks
  - Complex RAID group management, stripe configuration, write caches, etc
- Well architected flash allows for “performance by default”
  - tempDB, transaction logs, and database files on the SAME RAID group
  - Organisation and separation is only required for administrative purposes
- Results can be dramatic
  - Many database customers report batch runtime reduction of > 10x
  - Ability to create make critical business decisions on LIVE data instead of lagging indicators

# Fast Data

## Real-time NoSQL

Fast store/retrieve

- Unstructured Data – tweets, sensor data, clickstream
- Data typically stored and retrieved as key-value pairs in flexible column families
- High transaction rates, many reads and writes, small block/chunk sizes (1K-1MB)
- Less well-suited for ad-hoc analysis than Hadoop
- TB to PB scale

## Batch-oriented Hadoop

Heavy lifting, processing

- Unstructured Data – emails, syslogs, clickstream data
- Optimised for large streaming reads of large blocks (128-256 MB) comprising large files
- Dynamic schema effectively applied on read
- Optimised to compute data locally at cost of normalisation
- Write once, read many
- Linear scaling to thousands of nodes and tens of PB
- Entire data set at play for a given query

## MPP Relational Columnar Database

"Scale-out BI/DW"

- Structured Data
- Optimised for DW/OLAP, some OLTP (ACID-compliant)
- Data stored via frequently-access columns rather than rows for faster retrieval
- Rigid schema applied to data on insert/update
- Read and write (insert, update) many times
- Somewhat limited linear scaling
- Queries often involve a smaller subset of data set vs. Hadoop
- TB to low PB size

# Read and Write Caches

## Inside hypervisors and operating systems

- Cache is everywhere
  - L1/L2/L3 in the processor
  - DRAM on the OS
  - DRAM on the array
- Cache is usually volatile
  - Write cache MUST be protected
    - If you ACK the write you MUST have it on stable media
- Cache is a crutch
  - Read cache exists to avoid reading a slow persistent datastore
  - Write cache exists to avoid delaying an application response
  - The faster you make the media, the less you need the crutch
  - Poorly designed systems can have a 100x performance variance for a cache miss

# Summary

- Flash is NOT disk
- Flash has unique strengths
  - Low latency READ
  - Low power
  - Media stability
- Flash has unique weakness
  - Random write performance
  - Endurance
  - Data retention
- Successful implementation requires unique design
  - Leverage the strengths
  - Mitigate the weakness

# Q & A

# Complete Your Online Session Evaluation

## Give us your feedback and receive a Cisco Live 2014 Polo Shirt!

Complete your Overall Event Survey and 5 Session Evaluations.

- Directly from your mobile device on the Cisco Live Mobile App
- By visiting the Cisco Live Mobile Site www.ciscoliveaustralia.com/mobile
- Visit any Cisco Live Internet Station located throughout the venue

Polo Shirts can be collected in the World of Solutions on Friday 21 March 12:00pm - 2:00pm

**Learn online with Cisco Live!**

Visit us online after the conference for full access to session videos and presentations. www.CiscoLiveAPAC.com

CISCO TM