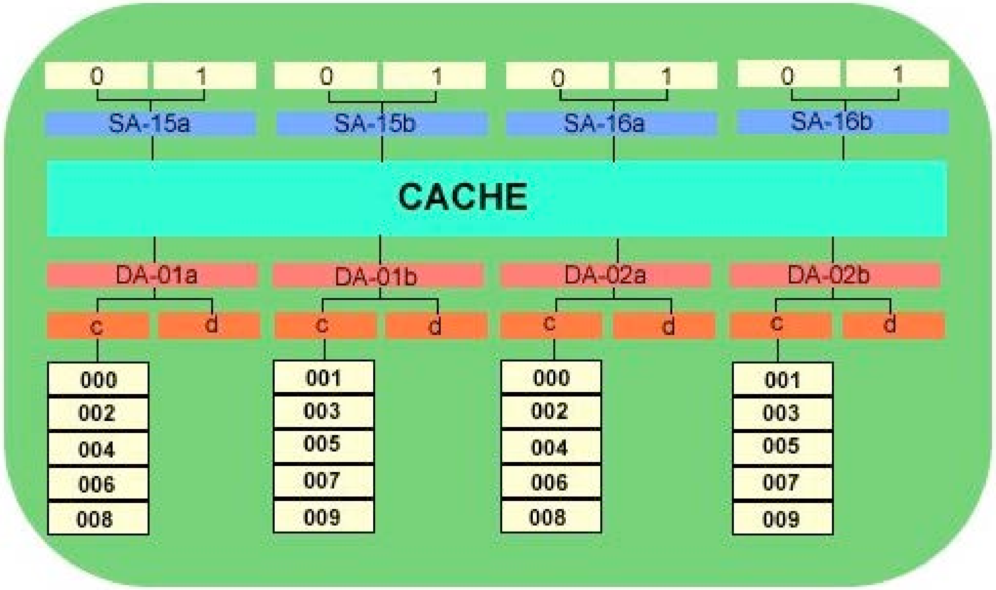
Getting Applications to Move at the Speed of Flash

Matt Goldensohn

Director, WHIPTAIL Labs

More than 20 Years Ago This Was a Game Changer. WHY?



0 1 0 1 0 1 0 1

SA-15a SA-15b SA-16a SA-16b

CACHE

DA-01a DA-01b DA-02a DA-02b

c d c d c d c d

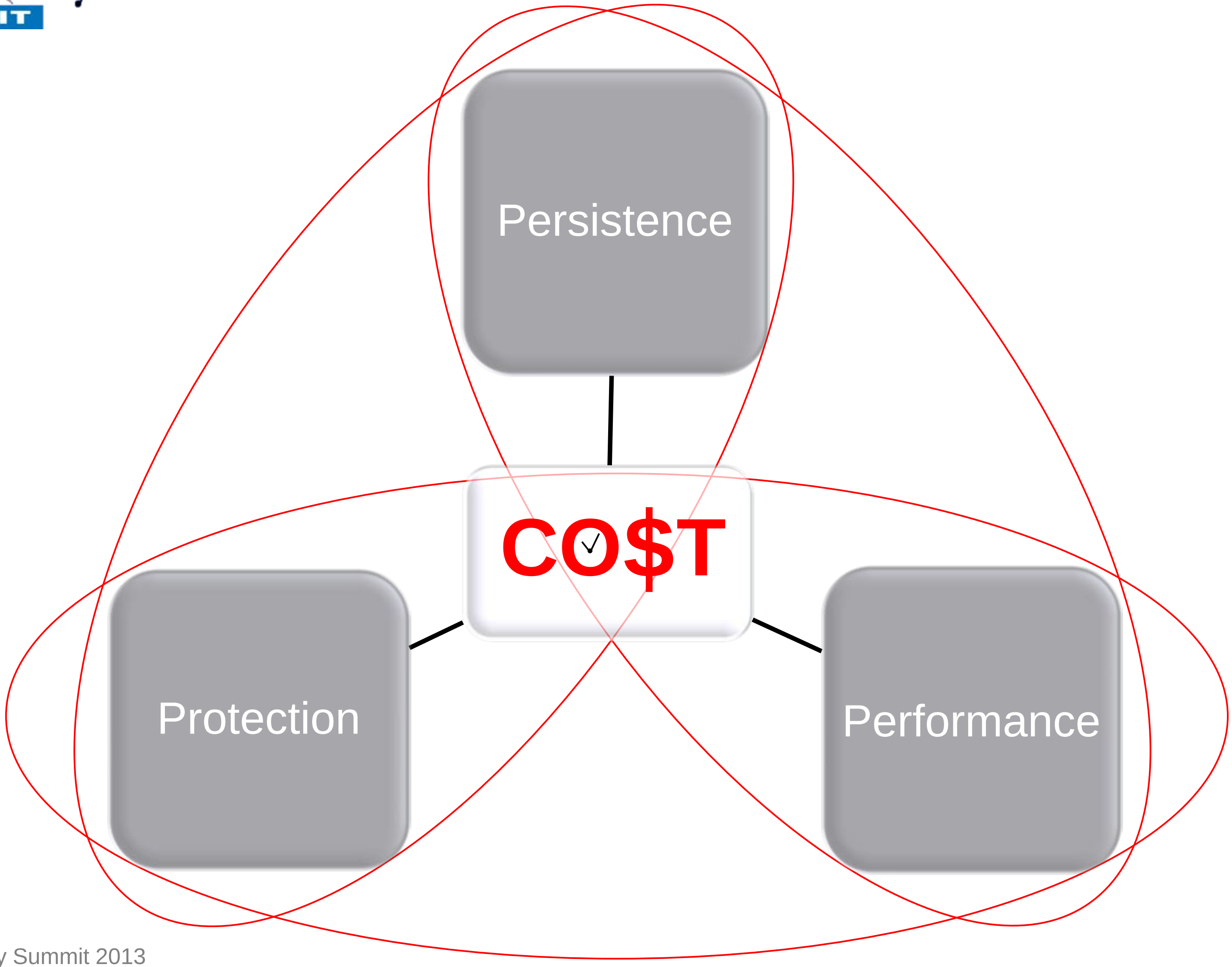
000 002 004 006 008

001 003 005 007 009

000 002 004 006 008

001 003 005 007 009

Cost/GB | The Storage Trade-Off



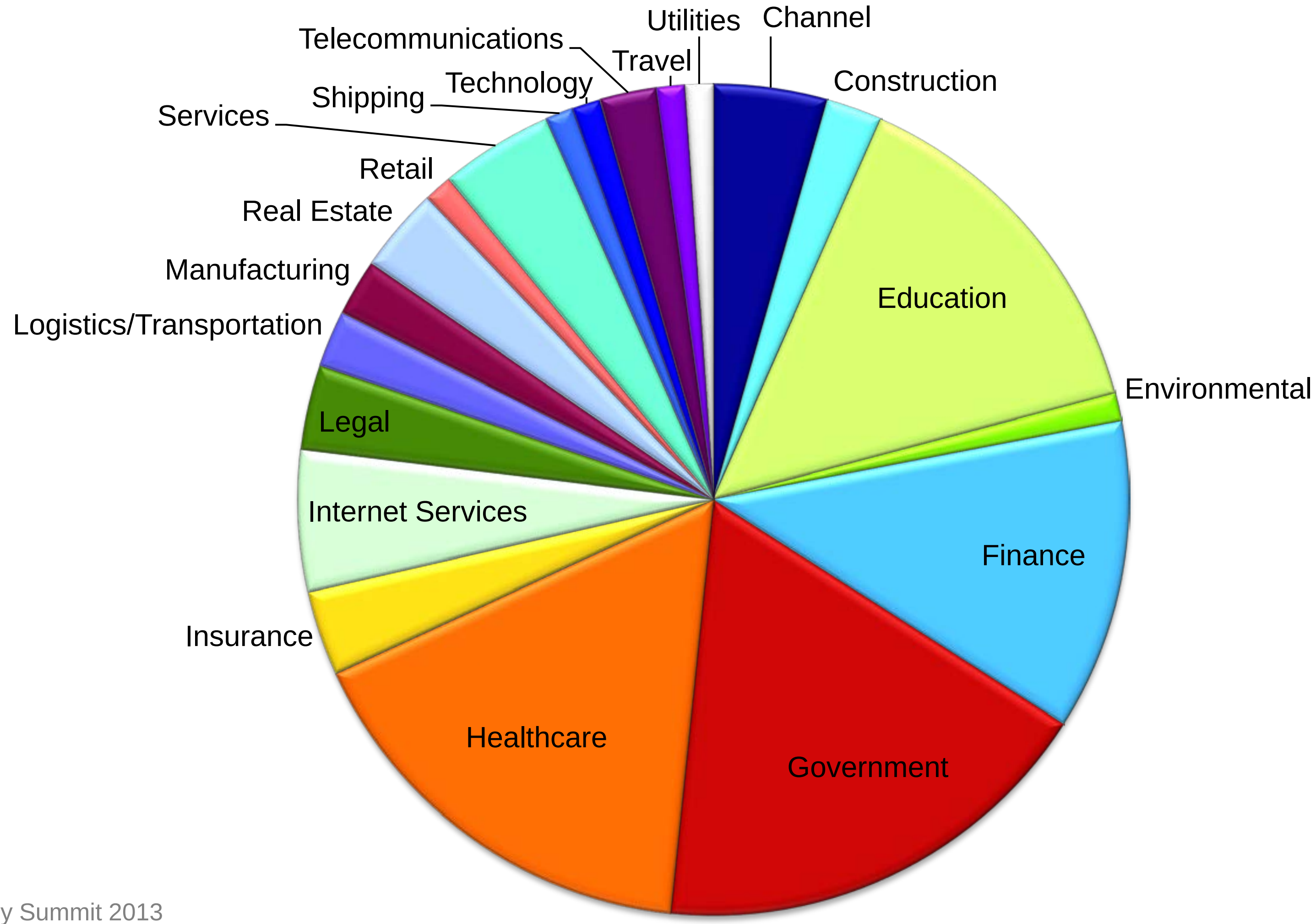
Flash is Today's Game Changer

- Traditional Storage Metrics *No Longer Apply*
  - Chasing IO, Writes & Reads
  - Measurement Based vs. Duration Based Testing
  - Use real data whenever possible
- SSD Moves Performance Constraints Up the Compute Stack
  - Operating System Configuration
  - Application
  - Network
  - Cache Management
- Expanding Pools of Flash Storage
  - Implications at the Application Level
    - Programs Have Been Focused on Mitigating Writes
    - Capacity to support an application is shrinking
    - With an abundance of capacity, *application design will be able to grow*
  - Meet tighter recovery time and point objectives

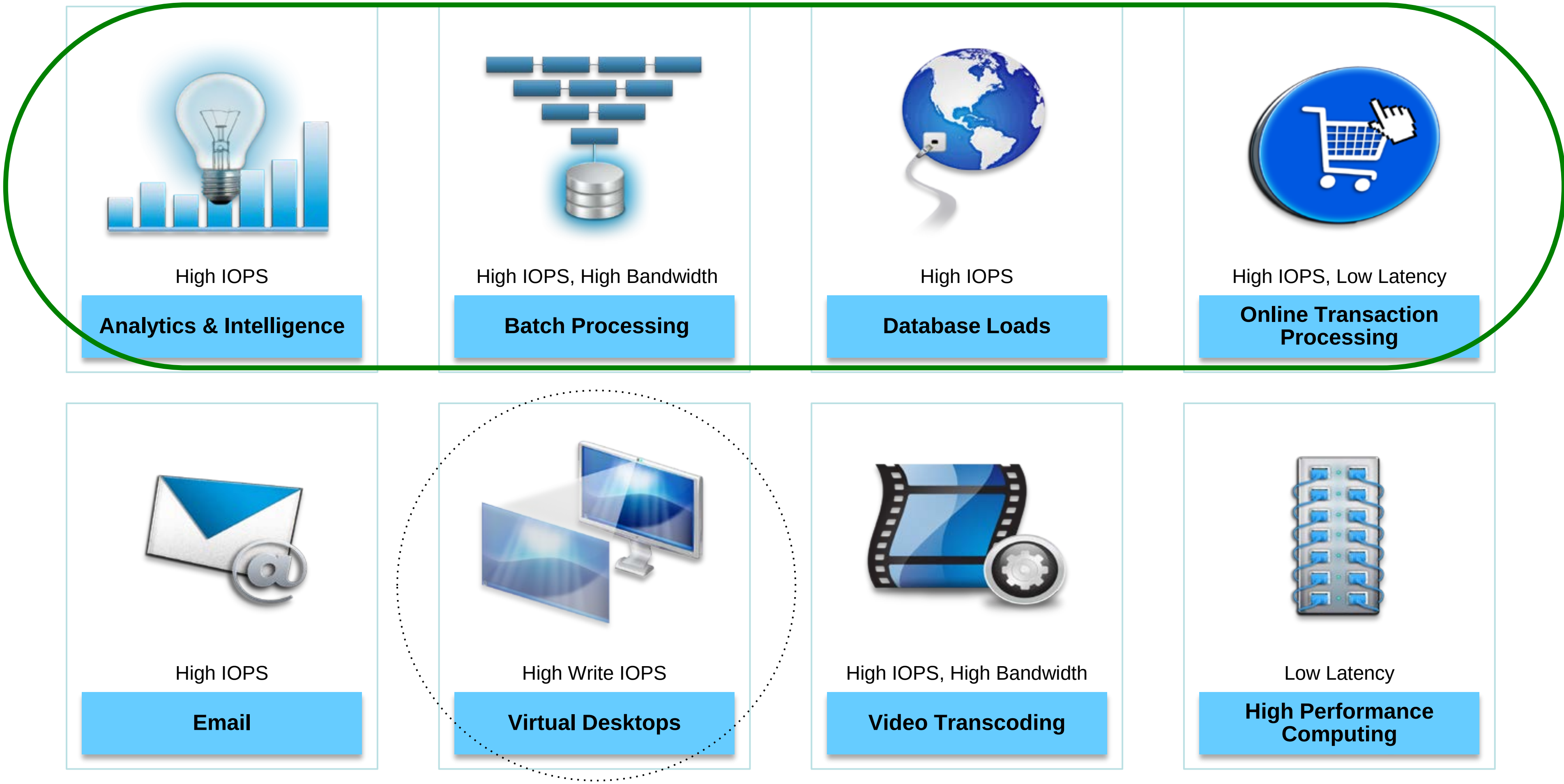
Flash Deployment will Affect Enterprise Application Behavior

- Enterprise Applications are Heavily WRITE Dependent
  - Lock, Group, & Join are All Writes
- A Flash Aware BTL is **Critical** to Sustain High WRITE Performance
  - Mitigate poor write behavior of SSDs
  - Garbage Collection
  - Minimize Write Amplification
  - Protection – RAID

Flash Accelerates Workloads Across All Industries



Enterprise Applications Deployed on Flash Platforms



High IOPS

Analytics & Intelligence

High IOPS, High Bandwidth

Batch Processing

High IOPS

Database Loads

High IOPS, Low Latency

Online Transaction Processing

High IOPS

Email

High Write IOPS

Virtual Desktops

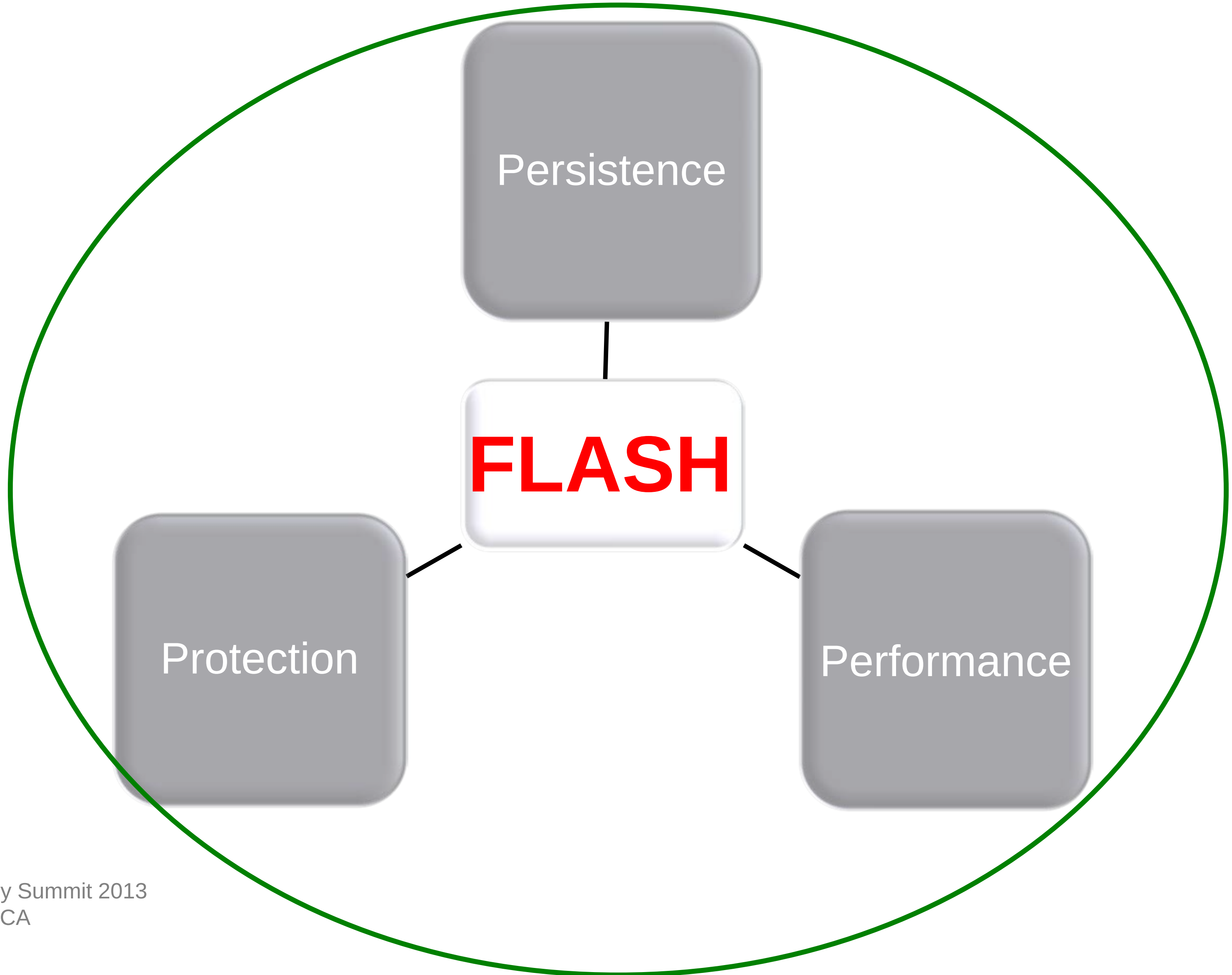
High IOPS, High Bandwidth

Video Transcoding

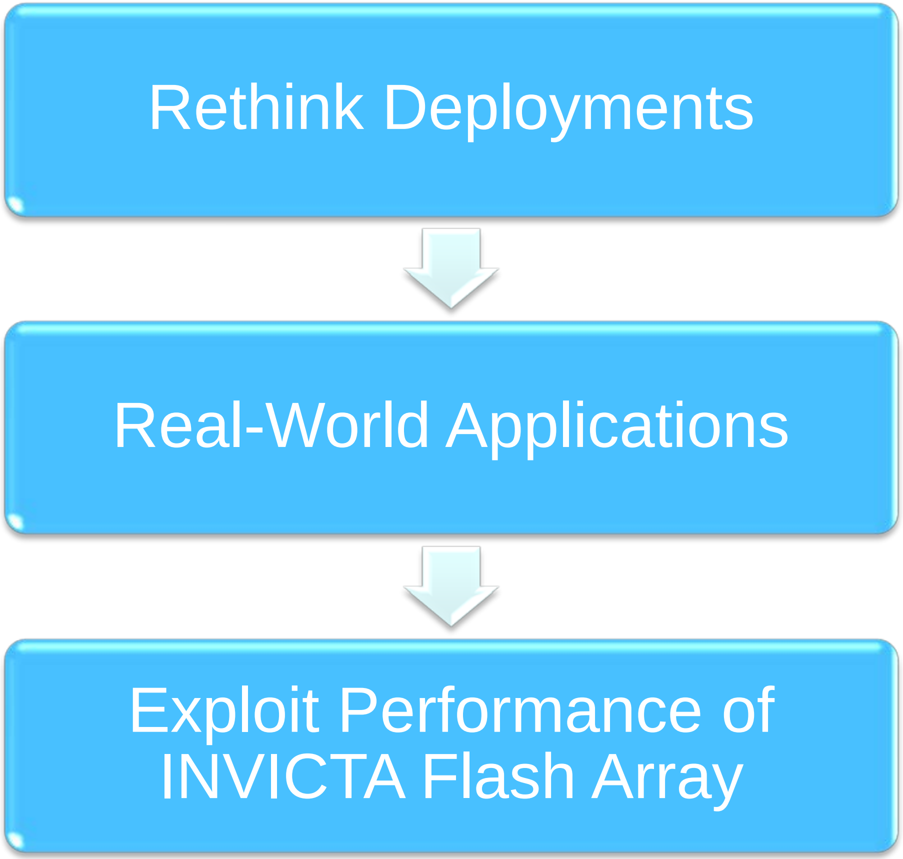
Low Latency

High Performance Computing

Cost/Transaction | You CAN Have It All



What does the Labs Do?



Rethink Deployments

Real-World Applications

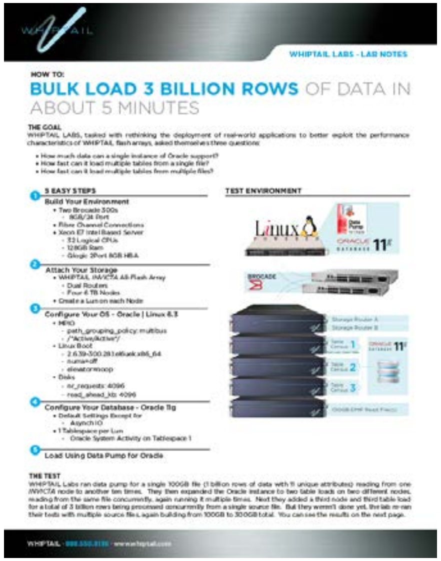
Exploit Performance of INVICTA Flash Array

What We Have Learned In The Lab

- Applications must be Optimized for Flash
  - Buffer Less
  - Read Faster
  - Write Faster
- Consolidation
  - Databases
  - Process
- Retune
  - Indexing Strategy
  - Placement
  - Block Size
  - OS IO Scheduler

Application Optimization

- Write Faster
- Read Faster
- Buffer Less



WHIPTAIL LABS - LAB NOTES

HOW TO:

BULK LOAD 3 BILLION ROWS OF DATA IN ABOUT 5 MINUTES

THE GOAL

WHIPTAIL LABS, tasked with rethinking the deployment of real-world applications to better exploit the performance characteristics of WHIPTAIL flash arrays, asked themselves three questions:

- How much data can a single instance of Oracle support?
- How fast can it load multiple tables from a single file?
- How fast can it load multiple tables from multiple files?

5 EASY STEPS

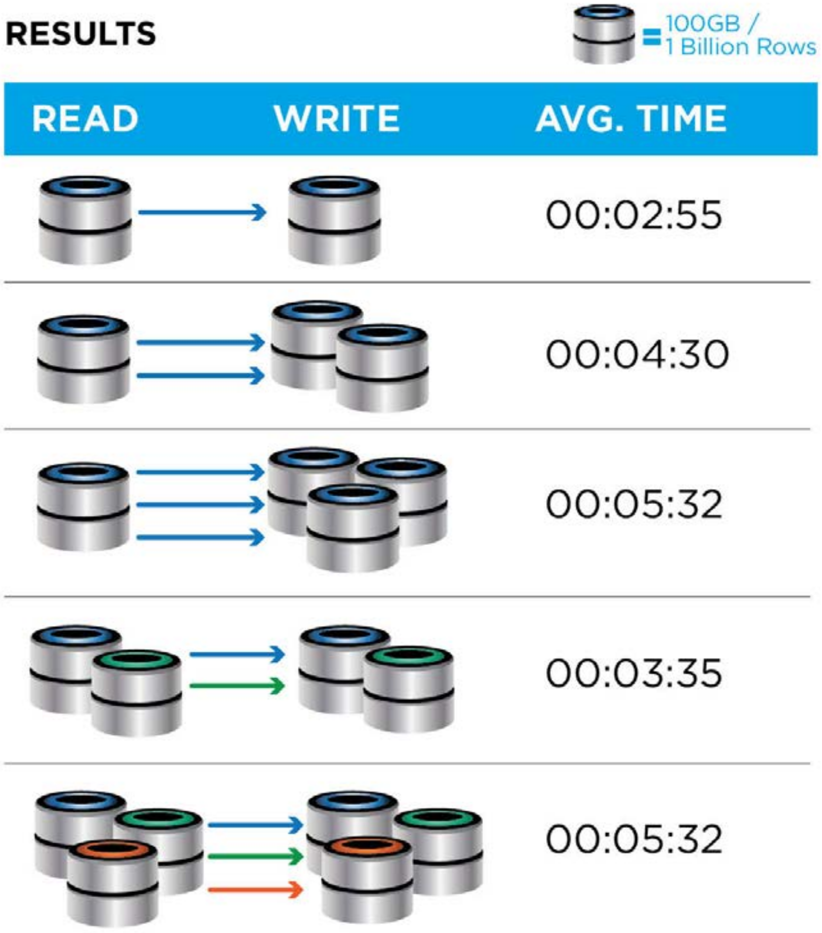
1. Build Your Environment
   - Two Brocade 300s
     - 8GB/24 Port
   - Fibre Channel Connections
   - Xeon E7 Intel Based Server
     - 32 Logical CPUs
     - 128GB Ram
     - Qlogic 2Port 8GB HBA
2. Attach Your Storage
   - WHIPTAIL INVICTA All-Flash Array
     - Dual Routers
     - Four 6 TB Nodes
   - Create a Lun on each Node
3. Configure Your OS - Oracle | Linux 6.3
   - MPIO
     - path_grouping_policy: multibus
     - /*Active/Active*/
   - Linux Boot
     - 2.6.39-300.28.1.el6uek.x86_64
     - numa=off
     - elevator=noop
   - Disks
     - nr_requests: 4096
     - read_ahead_kb: 4096
4. Configure Your Database - Oracle 11g
   - Default Settings Except for
     - Asynch IO
   - 1 Tablespace per Lun
     - Oracle System Activity on Tablespace 1
5. Load Using Data Pump for Oracle

TEST ENVIRONMENT

THE TEST

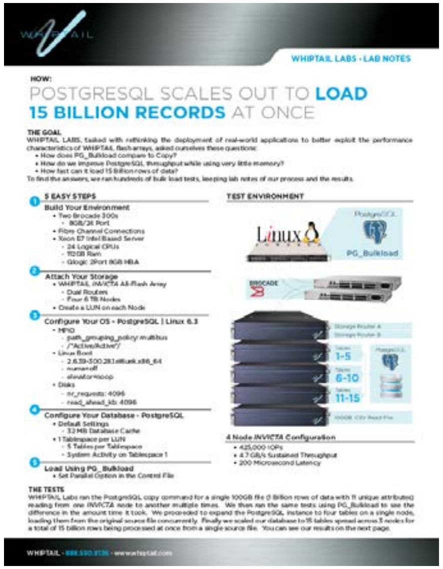
WHIPTAIL Labs ran data pump for a single 100GB file (1 billion rows of data with 11 unique attributes) reading from one INVICTA node to another ten times. They then expanded the Oracle instance to two table loads on two different nodes, reading from the same file concurrently, again running it multiple times. Next they added a third node and third table load for a total of 3 billion rows being processed concurrently from a single source file. But they weren't done yet, the lab re-ran their tests with multiple source files, again building from 100GB to 300GB total. You can see the results on the next page.

WHIPTAIL · 888.550.8136 · www.whiptail.com



Consolidation

- Databases
- Processes



WHIPTAIL LABS • LAB NOTES

HOW:

POSTGRESQL SCALES OUT TO LOAD 15 BILLION RECORDS AT ONCE

THE GOAL

WHIPTAIL LABS, tasked with rethinking the deployment of real-world applications to better exploit the performance characteristics of WHIPTAIL flash arrays, asked ourselves these questions:

- How does PG_Bulkload compare to Copy?
- How do we improve PostgreSQL throughput while using very little memory?
- How fast can it load 15 Billion rows of data?

To find the answers, we ran hundreds of bulk load tests, keeping lab notes of our process and the results.

5 EASY STEPS

1 **Build Your Environment**
- Two Brocade 300s
  - 8GB/24 Port
- Fibre Channel Connections
- Xeon E7 Intel Based Server
  - 24 Logical CPUs
  - 112GB Ram
  - Qlogic 2Port 8GB HBA

2 **Attach Your Storage**
- WHIPTAIL INVICTA All-Flash Array
  - Dual Routers
  - Four 6 TB Nodes
- Create a LUN on each Node

3 **Configure Your OS - PostgreSQL | Linux 6.3**
- MPIO
  - path_grouping_policy multibus
  - /"Active/Active"/
- Linux Boot
  - 2.6.39-300.28.1.el6uek.x86_64
  - numa=off
  - elevator=noop
- Disks
  - nr_requests: 4096
  - read_ahead_kb: 4096

4 **Configure Your Database - PostgreSQL**
- Default Settings
  - 32 MB Database Cache
- 1 Tablespace per LUN
  - 5 Tables per Tablespace
  - System Activity on Tablespace 1

5 **Load Using PG_Bulkload**
- Set Parallel Option in the Control File

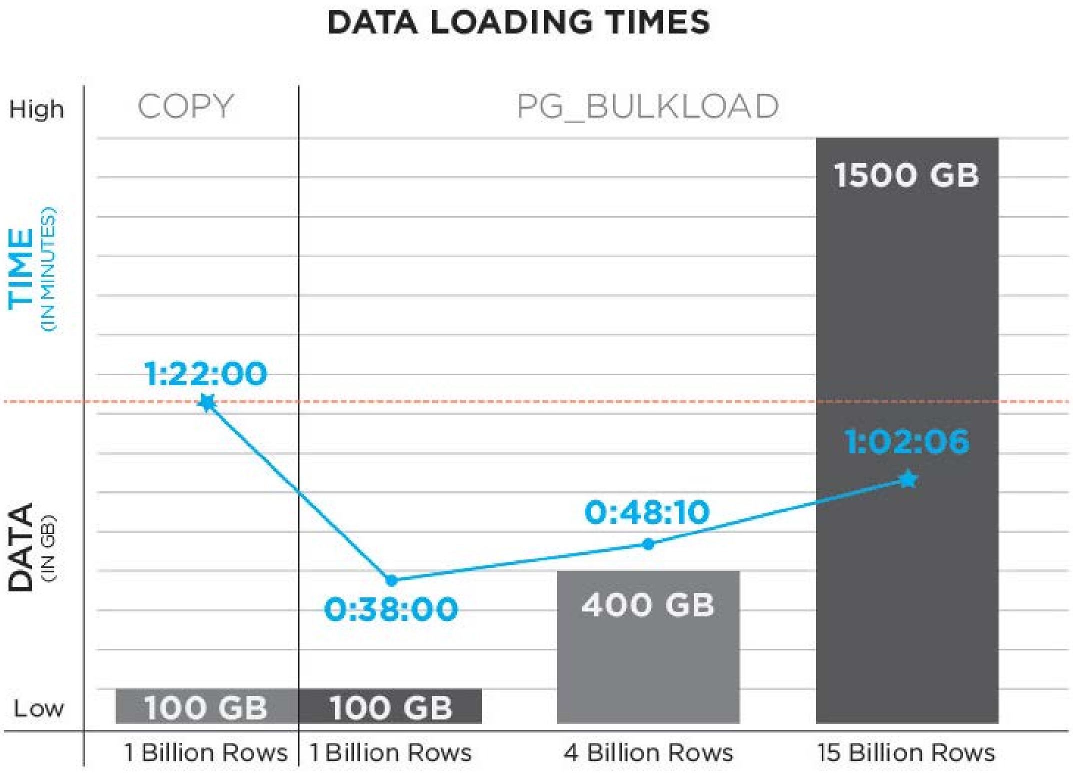
TEST ENVIRONMENT

4 Node *INVICTA* Configuration
- 425,000 IOPs
- 4.7 GB/s Sustained Throughput
- 200 Microsecond Latency

THE TESTS

WHIPTAIL Labs ran the PostgreSQL copy command for a single 100GB file (1 Billion rows of data with 11 unique attributes) reading from *one INVICTA* node to another multiple times. We then ran the same tests using PG_Bulkload to see the difference in the amount time it took. We proceeded to expand the PostgreSQL instance to four tables on a single node, loading them from the original source file concurrently. Finally we scaled our database to 15 tables spread across 3 nodes for a total of 15 billion rows being processed at once from a single source file. You can see our results on the next page.

WHIPTAIL - 888.550.8136 - www.whiptail.com



DATA LOADING TIMES

Rethink, Retune

- Indexing Strategy
- Placement
- Block Size
- OS IO Scheduler

WHIPTAIL LABS - BATCH LAB NOTES

HOW TO:

BATCH PROCESS 1 BILLION RECORDS
IN 10 MINUTES WITH ORACLE

THE GOAL

WHIPTAIL LABS, tasked with rethinking the deployment of real-world applications to better exploit the performance characteristics of WHIPTAIL flash arrays, asked ourselves "How fast can Oracle load and query a billion rows of data?" The test we ran and results are documented here.

6 EASY STEPS

1 Build Your Environment
- Two Brocade 300s
  - 8GB/24 Port
- Fibre Channel Connections
- Xeon E7 Intel Based Server
  - 32 Logical CPUs
  - 128GB Ram
  - Qlogic 2Port 8GB HBA

2 Attach Your Storage
- WHIPTAIL INVICTA All-Flash Array
  - Dual Routers
  - Four 6 TB Nodes
- Configured for RAID 5

3 Configure Your OS - Oracle | Linux 6.3
- MPIO
  - path_grouping_policy multibus
  - /"Active/Active"/
- Linux Boot
  - 2.6.39-300.28.1.el6uek.x86_64
  - numa=off
  - elevator=noop
- Disks
  - nr_requests: 4096
  - read_ahead_kb: 4096

4 Configure Your Database - Oracle 11g
- Default Settings Except for
  - AsynchIO
- Within 1 Node, Create 2 LUNS
  - 1 Tablespace on one LUN
  - Master Files on the other
- Set 8 Threads

5 Load Using Data Pump for Oracle

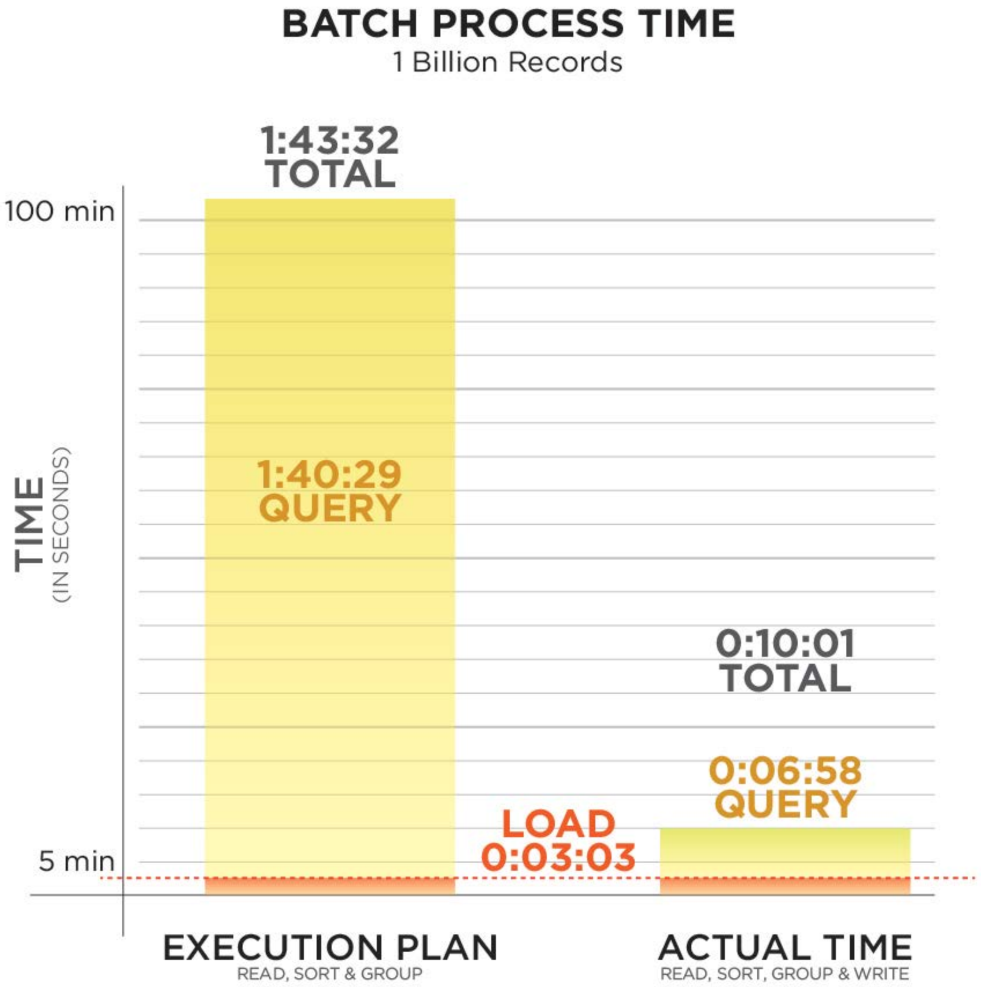
6 Sort and Group Using Oracle Database 11g
- Query String
  - select /*+ parallel(8) */ COUNTRY, STATE, CITY, ZIP, AGE, SEX, PRI_LANG, MAR_STATUS, Count(*) Counter
  - group by COUNTRY, STATE, CITY, ZIP, AGE, SEX, PRI_LANG, MAR_STATUS

TEST ENVIRONMENT

THE TEST

WHIPTAIL Labs took a single file containing 1 billion rows of data with 11 unique attributes and loaded it from one INVICTA node to another using data pump. We then designed a query to analyze the data we had just written, and made note of the execution plan Oracle projected. Our next step was to run the query without indexing and write the resulting 909 million record table on the same LUN within the same INVICTA node. You can see our results on the next page.

WHIPTAIL - 888.550.8136 - www.whiptail.com



The Take Away

- Perform Ad Hoc Queries Quickly & Easily
- Consolidate Databases & Processes
  - Manager Fewer LUNs
  - Lower Costs
    - Licensing
    - Server Management
    - Maintenance
  - Efficient Utilization of Existing Resources
- Critical Enterprise Applications Move to Flash
  - Prepare for New Challenges
  - Unique to Each Application