Direct NFS - Design considerations for next-gen NAS appliances optimized for database workloads

Akshay Shah

Gurmeet Goindi

Oracle

Agenda

- Introduction
- Database Architecture
- Direct NFS Client
- NFS Server Improvements
- NFS Protocol Extensions
- Flash for Databases
- Conclusion

Agenda

- Introduction
- Database Architecture
- Direct NFS Client
- NFS Server Improvements
- NFS Protocol Extensions
- Flash for Databases
- Conclusion

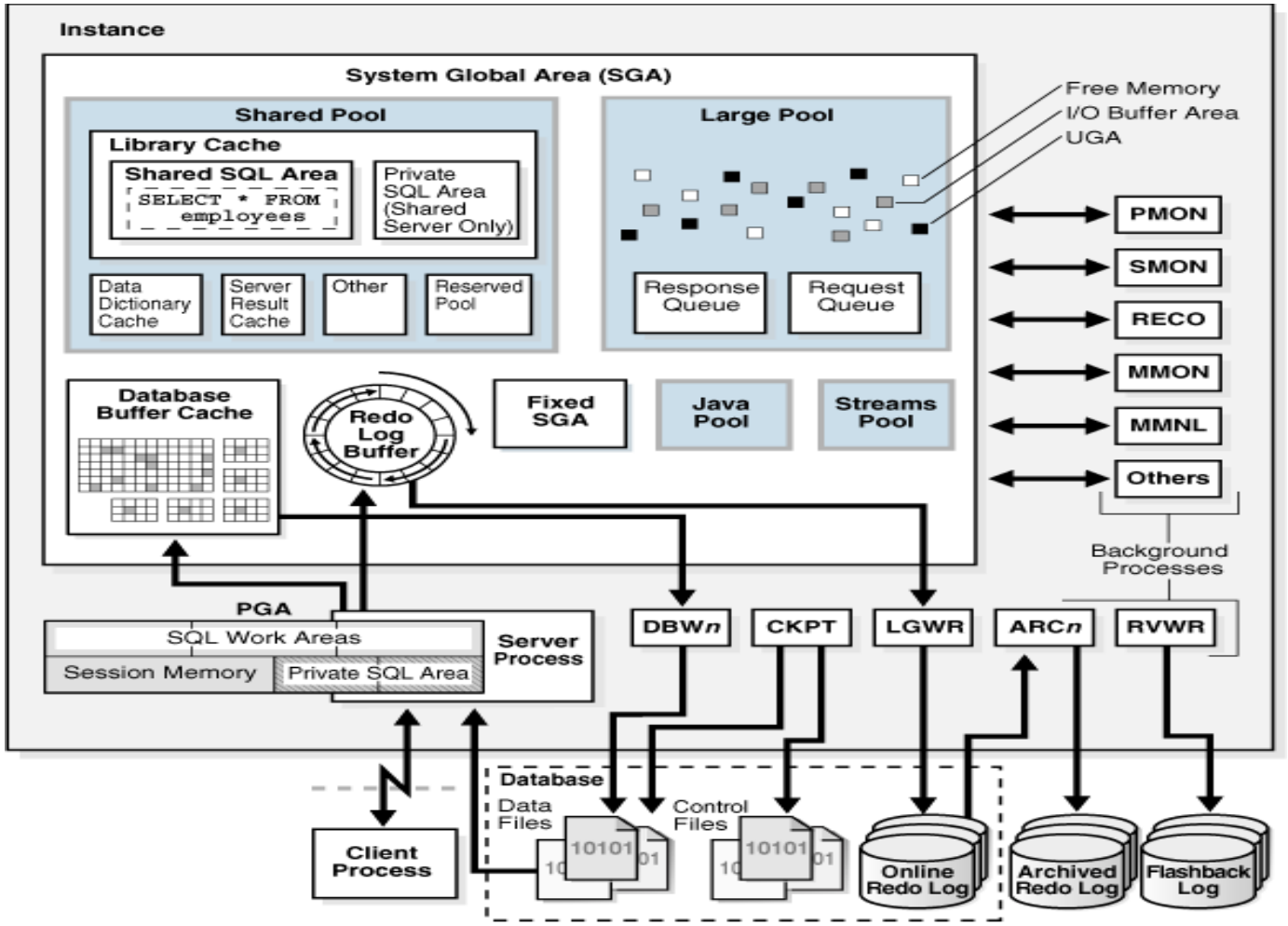
Introduction

- How do databases use storage?
- Pros and cons of NFS for databases
- What can improve database I/O performance?
  - Optimized NFS client
  - Scalable NFS server
  - Protocol extensions
  - Flash storage

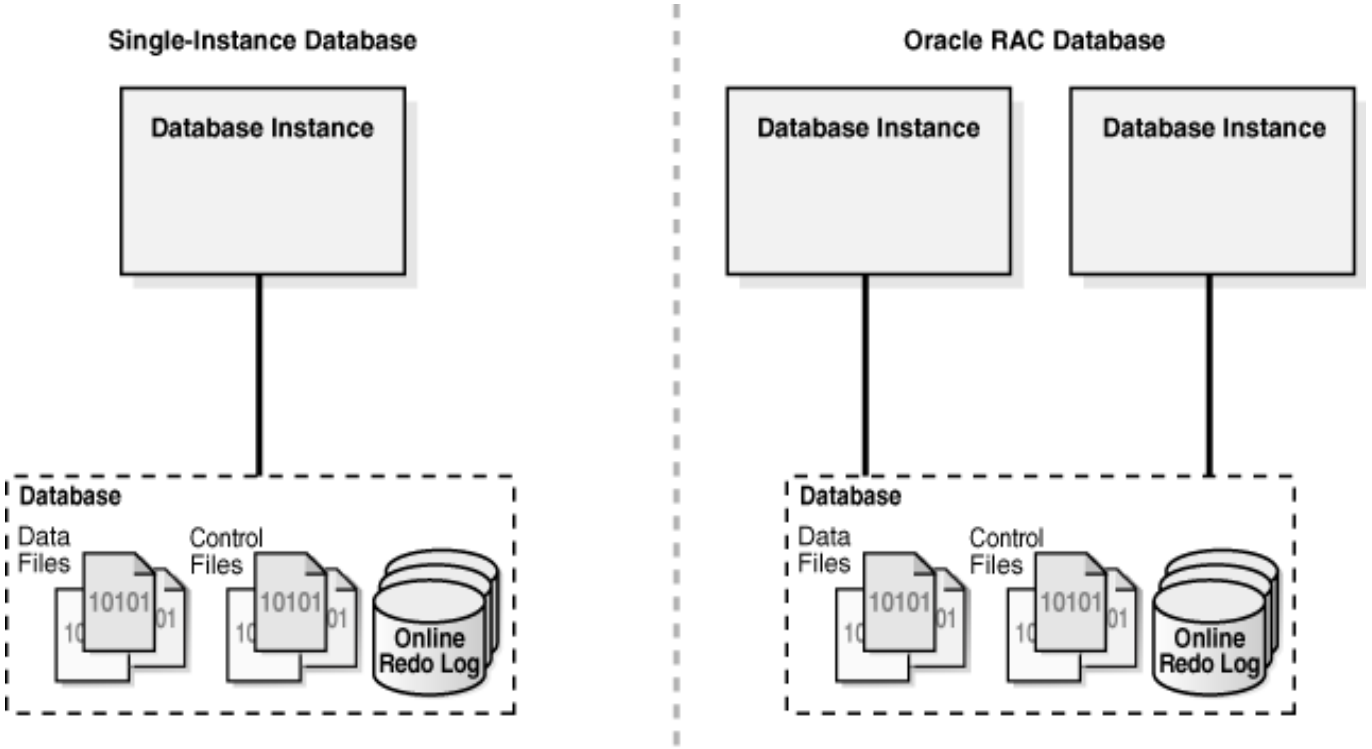
Agenda

- Introduction
- **Database Architecture**
- Direct NFS Client
- NFS Server Improvements
- NFS Protocol Extensions
- Flash for Databases
- Conclusion

Database Architecture



Database Architecture



Database Architecture

- Multiple processes issuing 1000s of I/Os concurrently
- Cluster database instances access shared storage from different nodes
- Buffer cache to cache hot blocks
- Strict consistency guarantees for database I/Os
  - Data consistency across database instances
  - Blocks written to persistent stable storage

Database I/O Workloads

- Transactional Processing workloads
  - Random small reads and writes
  - Typically, database block size e.g. 8 KB
  - Sequential Logging writes
    - Critical for database performance
    - Provide transactional consistency
  - Typically, latency sensitive I/Os

Database I/O Workloads

- Multi-user Large Sequential I/Os
  - 1 MB reads issued by table scans
  - Data loads issue 1 MB writes
  - Multiple streams disrupt on-disk sequentiality
  - Throughput intensive I/Os
  - Throughput higher when 1 MB I/Os hit disk without being split into smaller chunks

Database I/O Workloads

- Other miscellaneous workloads
  - Database backups
    - Typically, write once read never
    - Sequential full backups
    - Random incremental backups
  - Database recovery
    - Parallel reads of database backups
    - Extremely latency sensitive to reduce downtime

NFS for Databases

- Attractive option for database storage
  - Easy to deploy and administer
  - Standard Ethernet backbone
  - Cheaper than Storage Area Networks (SAN)

NFS for Databases

- NFS is not very popular for database storage
  - Inferior performance compared to SAN
  - NFS client not optimized for database
  - Client management complexities
    - Configuration and mount options vary across different platforms
    - Performance characteristics different across different platforms

Agenda

- Introduction
- Database Architecture
- **Direct NFS Client**
- NFS Server Improvements
- NFS Protocol Extensions
- Flash for Databases
- Conclusion

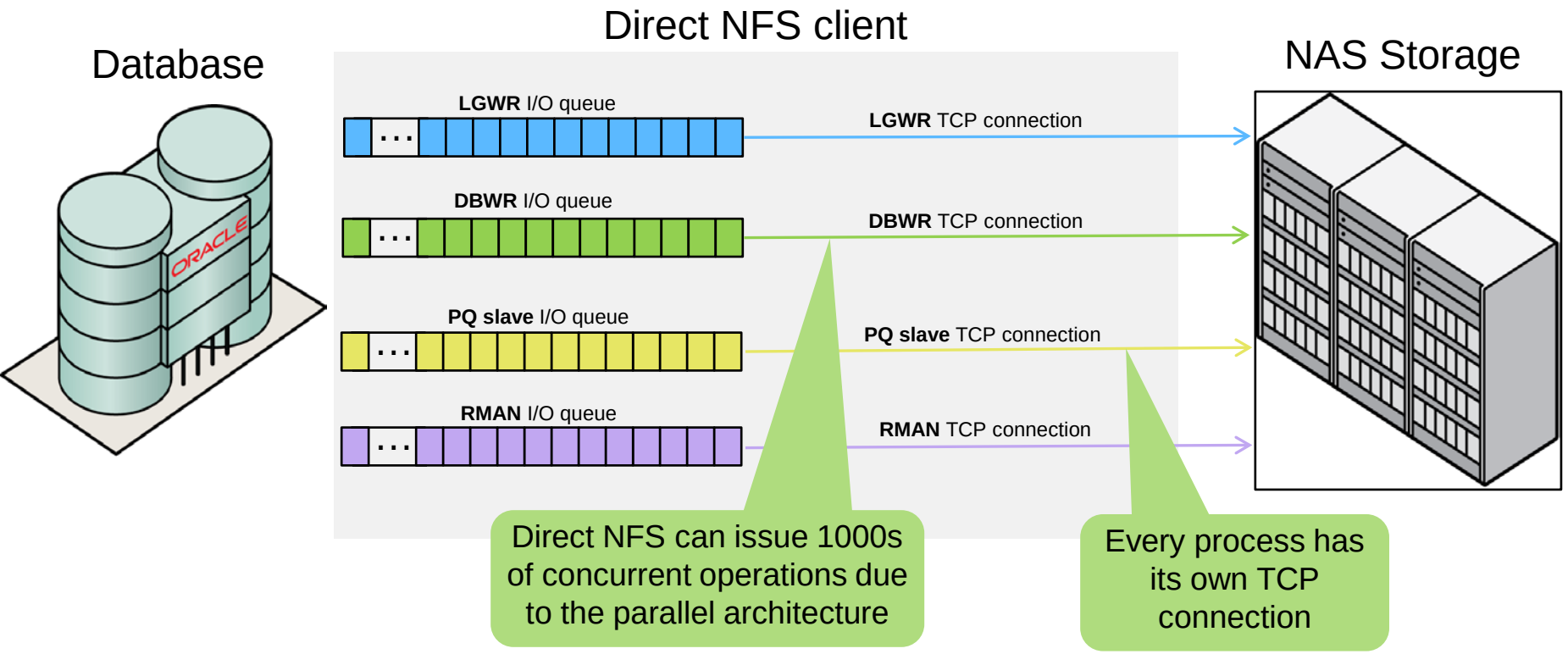
Direct NFS Client

- Direct NFS Client
  - User space NFS client
  - Optimized for database access patterns
  - Easy to configure and administer
  - Massively Parallel I/O architecture
  - Scalability and High Availability

Direct NFS Client

- Optimized for database access patterns
  - No locking at NFS layer
  - Not POSIX compliant
  - I/O size does not depend on mount options
    - Query NFS server for largest supported I/O size
    - Client behavior tuned for database requirements

Direct NFS Client

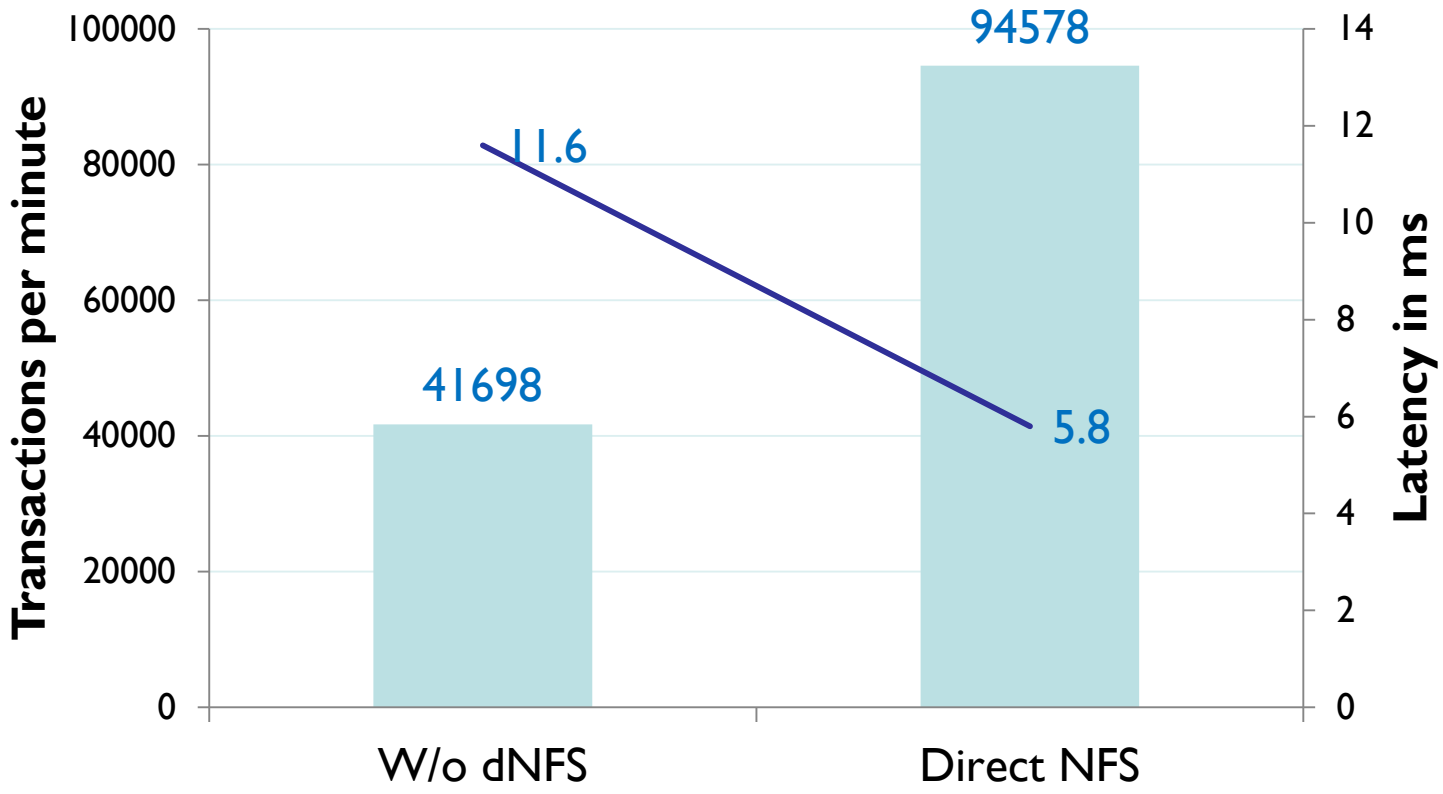


Direct NFS Client

- Massively Parallel I/O Architecture
  - Separate TCP connection for each process
  - Direct I/O and Asynchronous I/O
  - No limit on number of concurrent operations
- Scalability and High Availability
  - Optimize server scalability with multiple paths
  - Load balance across available network paths
  - Improve high availability by managing failover of network paths

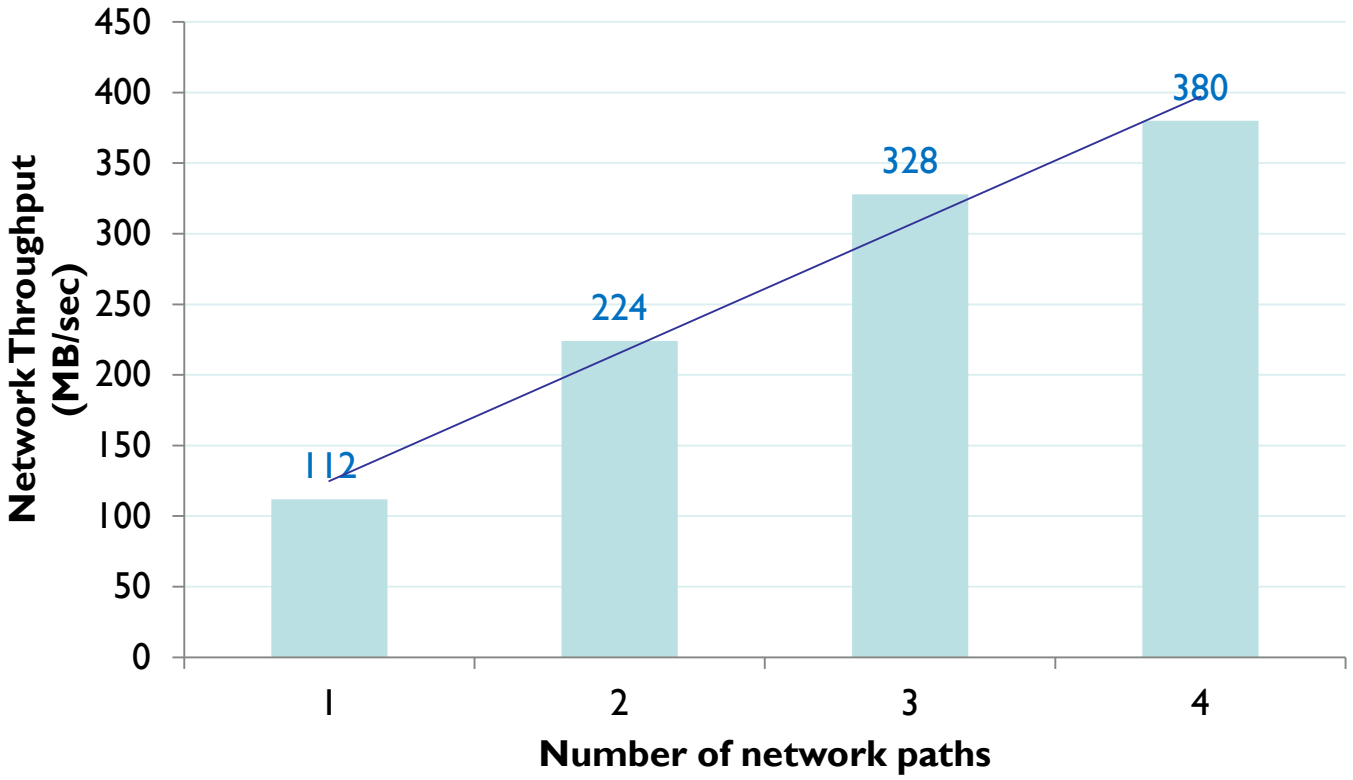
Direct NFS Performance

- Transactional Processing Workload
  - 1 TB database, 8000 warehouses, 250 concurrent users
  - 220% IOPS improvement, 50% latency reduction



Direct NFS Performance

- Large Parallel Table Scan – 150 million rows
- Linear throughput scaling with multiple paths



Agenda

- Introduction
- Database Architecture
- Direct NFS Client
- **NFS Server Improvements**
- NFS Protocol Extensions
- Flash for Databases
- Conclusion

NFS Server Improvements

- Large (1 MB) maximum I/O size
  - Optimized clients can issue large (1 MB) I/Os
  - Max I/O size < 1 MB causes NFS I/Os to be split into multiple small chunks
  - 1 MB NFS I/Os improve overall throughput and reduce protocol overhead
- Scalable Duplicate Reply Cache (DRC)
  - DRC doesn't scale to 1000s of concurrent operations
  - DRC not designed to handle 100s of TCP connections from a single machine
  - DRC behavior critical to database consistency requirements

NFS Server Improvements

- NFS server file system block size should be a multiple of database block size
  - Mismatch in block sizes can lead to fractured database blocks
  - Performance can be impacted due to read modify writes during updates
- Alignment can be configured manually
  - Direct NFS Client can tune this automatically
  - Pass block size hints during file creation

Agenda

- Introduction
- Database Architecture
- Direct NFS Client
- NFS Server Improvements
- **NFS Protocol Extensions**
- Flash for Databases
- Conclusion

NFS Protocol Extensions

- Workload agnostic NFS clients
  - No notion of workload access pattern
    - Sequential access or Random access
    - Likelihood of subsequent reads
  - No notion of frequency of access
    - Is this block frequently accessed?
    - Should the block be cached?
  - No notion about nature of the I/O
    - Is the I/O latency sensitive?

NFS Protocol Extensions

- Direct NFS Client
  - Intimate knowledge of workload access pattern
    - Distinguish sequential scans from random I/Os
    - Trigger storage prefetching for sequential scans
  - Ability to identify critical latency sensitive I/Os
  - Ability to identify frequently accessed blocks
    - Hot blocks accessed frequently in buffer cache
    - Only the database knows the true access counts for different blocks
- Direct NFS Client can pass hints to the storage

NFS Protocol Extensions

- Protocol Extensions to pass hints to storage
  - I/O access pattern hints
    - Sequential/ Random
  - Storage Prefetch hints
    - Prefetch/ Don't prefetch
  - Storage cache hints
    - Positive cache hints for frequently accessed blocks
    - Negative cache hints for scans, backups
  - Latency hints for critical I/Os
    - Critical I/Os can be served by a different storage tier such as flash instead of spinning disks
  - Quality of Service (QoS) hints

Agenda

- Introduction
- Database Architecture
- Direct NFS Client
- NFS Server Improvements
- NFS Protocol Extensions
- **Flash for Databases**
- Conclusion

Flash for Databases

- Promise of Flash Storage
  - Low latency, High IOPS
  - Lower energy and cooling costs
- Cons of Flash Storage
  - Lower scan bandwidth
  - Complex firmware for write wear leveling
  - Prohibitively expensive

Flash for Databases

- Server attached Flash Storage
  - Cannot scale out to cluster database
  - Single point of failure without mirrored flash
  - Poor resource utilization
- All Flash arrays
  - Best in class performance, extremely expensive
  - Limited throughput
  - Missing functionality (cloning, snapshots) compared to traditional arrays
  - Availability not the same as traditional arrays

Flash for Databases

- Hybrid Storage with Flash
  - Different storage tiers based on performance
  - Controller can move data between tiers
  - Shared storage permits database clustering
- Cache hints can improve flash performance
  - Cache hot blocks in flash
  - Critical latency sensitive I/Os served by flash
  - Don’t cache scan blocks in flash

Agenda

- Introduction
- Database Architecture
- Direct NFS Client
- NFS Server Improvements
- NFS Protocol Extensions
- Flash for Databases
- **Conclusion**

Conclusion

- NFS clients and servers can be optimized for better database performance
- Generic protocol extensions will allow intelligent applications to pass hints to the storage
- Flash storage can significantly improve database performance when deployed correctly
- Integrated database and storage appliances can leverage these performance optimizations

Conclusion

- Thank You!!
- Questions?