Database Performance Fundamentals

Main Topic

Main Topic

performance tuning overview

- performance planning
  - understanging investment options
  - understanding scalability
  - system architecture
  - application design principles
  - workload testing, modeling, and implementation
  - deploying new applications
- instance tuning
  - allocating memory to database structures
  - determining I/O requirements of differents parts of the database
  - tuning de the operating system for optimal performance of database
- performance principles
  - identifying most significant bottle neck
  - make changes to reduce or eliminate the effect of bottleneck
  - performed reactively
- baselines
  - application statistics
  - database statistics
  - operating system statistics
  - disk I/O statistics
  - network statistics
- symptoms and problems
  - slow physical I/O
    - poorly configured disks
  - latch contention
    - resolved through application changes
  - excessive cpu usage
    - little idle cpu on the system
- when to tune?
  - proactive monitoring
    - regularly scheduled interval
  - bottleneck monitoring
    - effective way = change application
- performance tuning features and tools
  - automatic performance tuning features
    - awr
    - addm
    - sql tuning advisor
    - sql access advisor
    - end to end application
    - server-generated alerts
    - v$ performance views

designing and developing for performance

- oracle methodology
  - system performance designed and built into a system
  - careful planning and design of databases
  - prevent exhaustion
- understanding investment options
  - adding hardware
    - short term relief
    - does not always improve system's performance
  - long term
    - more valuable
- scalability
  - systems ability to process more workload
  - poor scalability
    - incresead locking activities
    - increased operating system workload
    - transactions requiring increseas in data access as data volumes
  - unscalable if
    - exhausts a system resource to the point where no more throughput is possible when workload is increased
    - examples
      - hardware exhaustion
      - table scans in high volume causing inevitable disk I/O shortages
      - memory allocation causing paging and swapping
  - factors preventing scalability
    - poor app design, implementation and configuration
    - incorrect sizing of hardware components
    - limitations of software components
    - limitation of hardware components
- system architecture
  - hardware and software componentes
    - hardware
      - CPU
      - Memory
      - I/O Subsystem
      - Network
    - software
      - User interface
      - business logic
      - resources for managing user requests
      - data and transactions
  - configuring the right system architecture for your requirements
    - questions to be asked
      - satisfy the system requirements within budget and schedule
      - how many users must the system support
      - what will be the user interaction method
      - where are the users located
      - what is the network speed
      - how much data will users access
      - what is the user responde time requirement
      - do users expect 24 hour service
      - must all changes be made in real time
      - how big will the database be
      - what are the availability requirements
      - do skills exist to build and administer the app
- application design principles
  - simplicity in application design
  - data modeling
  - table and index design
  - using views
  - SQL execution efficiency
    - good database connection
    - good cursor usage and management
      - hard parsing
      - soft parsing
  - implementing the app
  - trends in application development
- workload testing, modeling and implementation
  - sizing data
  - estimating workloads
  - application modeling
  - testing, debugging, and validating design
- deploying new applications
  - rollout strategies
    - big bang approach
    - trickle approach
  - perfoprmance checklist

performance improvement methods

- oracle performance improvement method
  - perform standard checks
  - check top 10 common mistakes
  - build conceptual model
  - propose a series of remedy actions
  - validate changes
  - repeat last 3 steps
- emergency performance methods
  - survey performance problem
  - sanity check the hardware
  - determine if the database server is constrained on CPU
  - Apply emergency action to stabilize system

configuring a database for performance

- performance considerations for initial instance configuration
  - initialization parameters
  - unod space
  - redo log files
  - tablespaces
- creating and maintaining tables for optimal performance
  - table compression
  - reclaiming unused space
  - indexing data
- performance considerations for shared servers
  - identifying contention for shared servers
  - identifying and reducing contention using the dispatcher-specific views
    - adding dispatcher for dispatcher processes
    - enabling connection pooling
    - enabling session multiplexing
- improved client connection performance due to prespawned processes — V$PROCESS_POOL — shows information about server process pools — manage — DBMS_PROCESS package