RamSan-630 SSD levert zeer hoge databaseprestaties

Versnelling van de Oracle Database

Bram Dons

De Oracle database wordt voor de meeste veeleisende applicaties als back-end relationele database toegepast. Het is daarom van belang dat kritische applicaties die van deze database gebruik maken optimaal presteren.

De traditionele benadering voor prestatieverbetering is de toevoeging van meer processorvermogen. Maar de praktijk leert dat dit niet of nauwelijks de prestaties verbetert. Dit komt omdat de processor, hoe snel dan ook, constant moet wachten op de data die afkomstig zijn van de mechanische hard disks (HD's). Terwijl iedere component in de 'dataketen' zich verplaatst met computersnelheden, hebben disks te maken met mechanische snelheden. Want het is afhankelijk van de fysieke verplaatsing van de disk-arm over de magnetische schijf en de omwentelingssnelheid van de schijf.

In de laatste twintig jaar zijn processorsnelheden enorm toegenomen, terwijl de conventionele toegangstijd tot data nauwelijks is verbeterd (zie afbeelding 1). Het resultaat is een enorm prestatieverschil dat blijft. Vooral transactiegebaseerde database servers ondervinden daar de meeste last van omdat daarbij de vraag naar IO transacties aanzienlijk groter is in vergelijking met andersoortige applicaties. De aanschaf van supersnelle processoren en grote hoeveelheden netwerkbandbreedte zijn dus verspild geld als een mechanische disk er nog steeds milliseconden over doet om data op te halen. Wanneer een server op een disk moet wachten, dan moet de gebruiker dat uiteindelijk ook. Deze zogenaamde 'IO wachttijd' is een probleem dat, zoals we hierna kunnen lezen, met een Solid State Disk (SSD) kan worden opgelost.

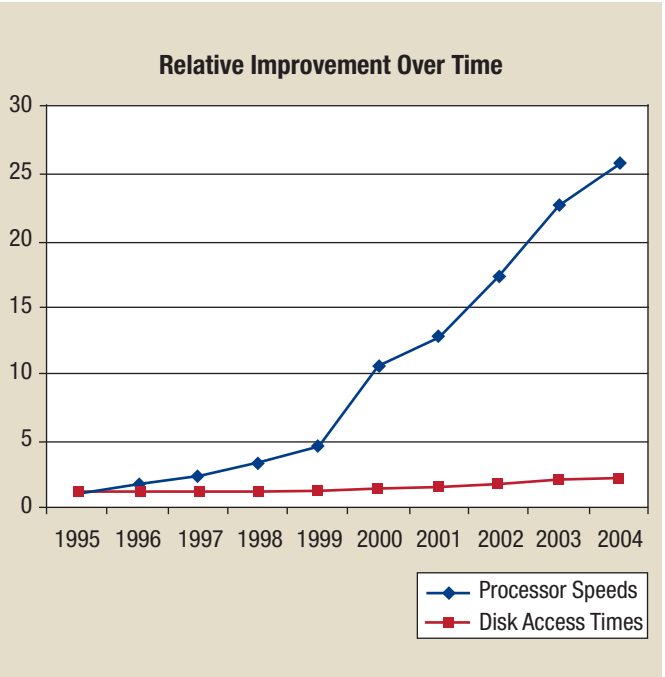
Oracle's database prestatiepijnpunten

In het algemeen is Oracle voor wat betreft data, read latency gevoelig en write latency ongevoelig. Een vertraging bij het lezen van data heeft directe gevolgen voor de Oracle prestaties. Bij het schrijven van data kan het van de 'lazy-write' methodologie gebruik maken; dit staat beter bekend als 'delayed block clean-out' methodologie. Oracle maakt alleen van deze methode gebruik om een data block naar het IO subsysteem te schrijven wanneer dat block nodig is voor een ander proces. Het block kan daardoor enkele seconden langer in cache bewaard blijven voordat het daadwerkelijk naar disk wordt weggeschreven.

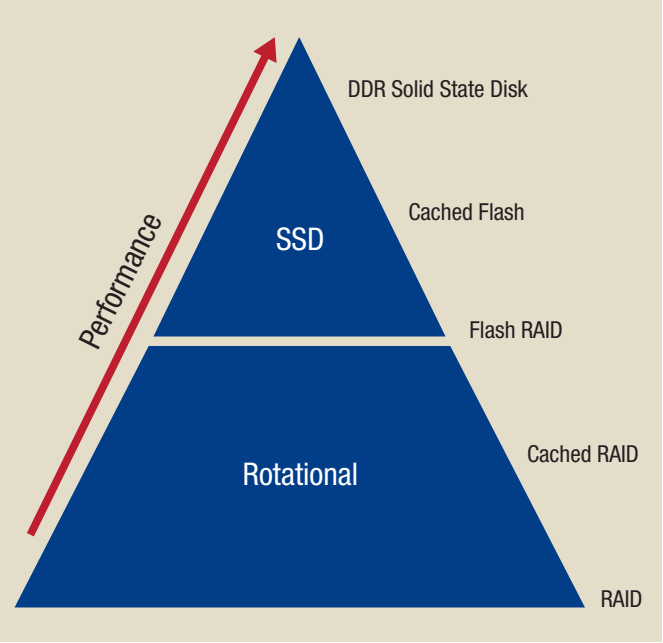
Er bestaan echter drie typen schrijfprocessen die zo snel mogelijk moeten plaats vinden, te weten: redo, undo en temporary block writes. Elk van deze writes kan als gevolg hebben dat een transactie moet wachten tot de write is uitgevoerd. Een hoge wachttijd (high latency) is niet acceptabel in een drukbezet systeem. In gevallen waarbij sprake is van een groot aantal database commits per seconde kunnen redo writes tot problemen leiden. Bij grote aantallen rollbacks kunnen undo blocks een read en write probleem veroorzaken en verder vormen alle intensieve temporary IO's bij elke database een zorgenkind. Redo en undo operaties zijn vaak sequentiële writes en moeten voor zover mogelijk worden afgescheiden van andere soorten IO. Tijdelijke tablespace IO, zelfs bij systemen met zero disk sorts, kunnen een belangrijke invloed uitoefenen op het totale IO plaatje. Dan nog te bedenken dat temporary IO niet alleen voor sorts maar ook bij hash joins, bitmap-operaties en temporary table operaties worden gebruikt die de PGA-limieten kunnen overschrijden.

Invloeden op prestaties IO subsysteem

De drie belangrijkste factoren die van invloed zijn op de prestaties van een IO subsysteem zijn capaciteit, bandbreedte en



Afbeelding 1: Processor speed versus Disk Access Time.



Afbeelding 2: Storage Piramide (bron TMS).

vertraging. Het capaciteitsprobleem is waarschijnlijk het eenvoudigst om op te lossen. Afhankelijk van de gewenste prestaties bepaalt men hoeveel diskcapaciteit van elke disk kan worden gebruikt, men trekt de rest daarvan af en deelt de benodigde capaciteit door de hoeveelheid die beschikbaar is op elke disk. Vervolgens vermenigvuldigt men dit getal voor concurrency en RAID-niveau, waarmee uiteindelijk het aantal fysieke disks wordt bepaald.
De hoeveelheid beschikbare bandbreedte tussen een storage systeem en server is afhankelijk van het toegepaste transportmedium (meestal Fibre Channel, Ethernet, of InfiniBand) en het aantal geïnstalleerde interfaces (host bus adapters, netwerk interfaces) tussen beide systemen. De bandbreedte van een IO subsysteem wordt meestal aangeduid als de hoeveelheid getransporteerde data per seconde; let op, leverancierspecificaties vermelden de bandbreedte in bits of bytes, dus Mbps of MBps: een factor 8 verschil! Het aantal IOP's kan ruwweg worden bepaald door de totale bandbreedte te delen door de gemiddelde transfergrootte, dus een 400 Mbps bandbreedte met een 128 KB IO diskblock grootte levert maximaal 3200 IOP's per interface op. De bandbreedte kan eenvoudig worden uitgebreid door de installatie van meerdere interfaces.

Vaak worden vertraging (latency) en IOP's aan elkaar gerelateerd. Het is zo dat meer IOP's eenvoudig zijn te bereiken, zelfs als dit gepaard gaat met een grote latency. Voor alle duidelijkheid, een lage latency resulteert in hogere IOP's maar een lage latency is niet strikt noodzakelijk om hogere IOP's te halen. Weliswaar kunnen met genoeg beschikbare bandbreedte disks aan een SAN worden toegevoegd om het aantal IOP's te vergroten, maar elke IO op zich neemt nog steeds 2 tot 5 ms in beslag. Het grootste voordeel van een lage latency is hogere prestatie. Want, des te lager de latency des te hoger is de kans dat een transactie in korte tijd kan worden beëindigd, de responstijd verminderd, en de prestaties vermeerderd.
In een drukbezette, met cache uitgerust SAN, zijn responstijden van 1 ms of beter mogelijk, totdat de cache verzadigd is waarna de basis diskgebaseerde latency-waarden weer gelden. In de meeste gevallen is 1 milliseconde de beste latency die men in een diskgebaseerd SAN kan bereiken. Om de 1 ms barrière te doorbreken is een DDR- of flash-gebaseerde SSD-oplossing nodig. In een DDR-gebaseerde SSD bedraagt de latency minder dan 20 microseconden en in een flash-gebaseerde SSD tussen 80 en 250 ms. De meeste enterprise SSD-systemen maken gebruik van een combinatie van DDR buffers en flash voor permanente storage. Zie afbeelding 2.

Hoeveel hard disks nodig?

De enige manier om het aantal IO's van traditioneel op HD's gebaseerde systemen op te voeren is het aantal actieve disks te vergroten. Een gemiddeld presterende disk kan maximaal 200 random IOP's afleveren. Als het computersysteem 10.000 IOP's vereist dan zijn daar dus tenminste 50 diskdrives voor nodig. Is er sprake van grote aantallen gelijktijdige gebruikers of worden veel volledige databasetabellen gescand, dan kan het aantal benodigde diskdrives snel toenemen. Bijvoorbeeld, uit een TPC-H datawarehouse benchmark blijkt dat (teneinde het vereiste aantal IOP's te kunnen garanderen) het aantal benodigde drives overeen komt met 30 tot 40 maal de benodigde storagecapaciteit. In veel gevallen zien we dat 10 tot 20 procent van de database goed is voor 90 procent van alle IOP's. De IOP's intensieve datatabellen en indexen belasten het IO subsysteem dusdanig dat de prestaties van de gehele applicatie onder de maat blijven.

RamSan-630 Flash SSD

De SSD-technologie bestaat al meer dan dertig jaar. Want in strikte zin genomen is een SSD elke storage device die niet gebruik maakt van mechanische onderdelen voor de in- en uitvoer van data. De term refereert aan storage devices die van geheugen (DDR of flash) als primair opslagmedium gebruik maken. De data worden direct in geheugenchips opgeslagen. SSD's hebben een interessante combinatie van karakteristieken: ze zijn duur in relatie tot de opslagcapaciteit maar extreem goedkoop voor de prestaties die ze kunnen leveren.

Eind vorig jaar kondigde de firma Texas Memory Systems (TMS) 'The world's fastest storage' systeem aan. Het RamSan-630 op SCL flash gebaseerde storagesysteem is in staat om maar liefst 1 miljoen IOP's en 10 GBps bandbreedte te leveren. De read latency is 250 microseconde, de write latency slechts 80 microseconde. Het storagesysteem heeft een opslagcapaciteit van 10 TB en is gehuisvest in een 3U package met aansluitingen voor Fibre Channel en InfiniBand. Het stroomverbruik is slechts 500 W. Ter vergelijking, voor het leveren van 1 miljoen IOP's zouden meer

dan 3.000 high performance HD's nodig zijn die minimaal 30.000 W zouden verbruiken. De 630 vormt ook de basis voor het RamSan-6300 systeem waarbij in een enkel 19 inch rack veertien 630's kunnen worden gecombineerd tot maximaal 140 TB aan storagecapaciteit met een totale bandbreedte van 140 GBps en 14 miljoen IOP's!

I/O's per Second

De RamSan-630 kan zoals gezegd 1 miljoen random IOP's ten behoeve van applicaties leveren. Dit met nadruk op het woord 'random' omdat de meeste leveranciers van harddisksystemen bijna altijd hun IOP's prestaties als sequentiële IOP's opgeven (die altijd vele malen hoger zijn dan die van random). Insgelijks worden de prestaties van flash SSD's vaak als read prestaties opgegeven. Het probleem met sequentiële IOP's is dat bijna geen enkele enterprise applicatie gebruik maakt van sequentiële disktoegang met kleine diskblocks.

Waarom zijn random IOP's dan zo belangrijk? Het antwoord is: ze worden veelvuldig toegepast in database transactions. Database transactions kennen twee eigenschappen: ze zijn klein (gemiddeld 8K groot) en ze zijn random van aard. De toegang van dergelijke kleine random files is zeer belastend voor harddisks. Een snelle harddisk levert ongeveer 300 random IOP's zodat er honderden drives nodig zijn om de snelheid van een RamSan-630 te kunnen evenaren. De snelste in de duurste harddiskgebaseerde storage array's ingebouwde cache kunnen hoogstens 150.000 IOP's leveren. Dat brengt ons terug tot de oorspronkelijke vraag: is het zo belangrijk om een hoog aantal IOP's te ondersteunen omdat servers toch ook een hoog aantal IOP's kunnen leveren? Het antwoord is simpel. Wanneer de processor sneller is dan het storagesysteem, dan moet de processor letterlijk op het storagesysteem wachten voordat het de gevraagde data krijgt, de in het begin van dit artikel genoemde 'IO wachttijd'. Daarbij komt; als de processor steeds moet wachten dan moeten de gebruikers dat ook. Het is bovendien een verspilde investering in de allerlaatste typen processoren en softwarelicenties. De RamSan-630 elimineert het IO wachtprobleem doordat het zeer hoge random IOP's voor alle opgeslagen files kan bieden.

Latency en IO

Piekprestaties van applicaties worden (afhankelijk van het aantal threads, blockgrootte en read/write patronen) beïnvloed door een combinatie van response time (latency) en piek IO. HD's-gebaseerde RAID-systemen hebben meestal een 4 tot 8 ms toegangstijd. Om de responstijden en prestaties te verhogen, voegen leveranciers van deze systemen RAM of flash cache aan de array controller toe. Daarmee wordt de latency tot een 0,5 ms teruggebracht, althans als de data zich in cache bevinden. Is dat niet het geval dan moeten de data toch nog gewoon van de back-end disks worden opgehaald en geldt weer de 4 tot 8 ms toegangstijd. De RamSan-630 daarentegen biedt altijd 0,08 ms

Afbeelding 3: Texas Memory Systems RamSan-360.

toegangstijd voor writes en 0,25 ms voor reads. Dit is tenminste een factor twintig sneller dan de meeste RAID-systemen kunnen bieden.

Traditionele aanpak prestatieverbetering

De combinatie van een toenemend aantal databasegebruikers, grotere datavolumes en meer complexe database query's zijn vaak de oorzaak van een tragere database respons. De bijna automatische reactie van databasebeheerders op dit soort problemen is door te kijken naar de prestaties van server, processor en SQL statements. Echter, in veel gevallen leidt de verhoging van de serverprestaties en SQL tuning alleen niet tot de gewenste verbeteringen. De toevoeging van server en processoren heeft slechts een minimale invloed op de databaseprestaties. De tuning van SQL kan wel leiden tot wat prestatieverbeteringen maar zelfs de best gedefinieerde SQL's kunnen een slecht presterend storagesysteem niet compenseren.
Voor een oplossing voor storageprestatieproblemen zochten tot op heden databasebeheerders traditioneel meestal naar drie verschillende methoden: verhoging van het aantal disks; verplaatsing van de meest bezochte files naar een eigen disk en de implementatie van een RAID-systeem. Sinds de komst van betaalbare SSD's is daar nu een vierde optie bijgekomen.

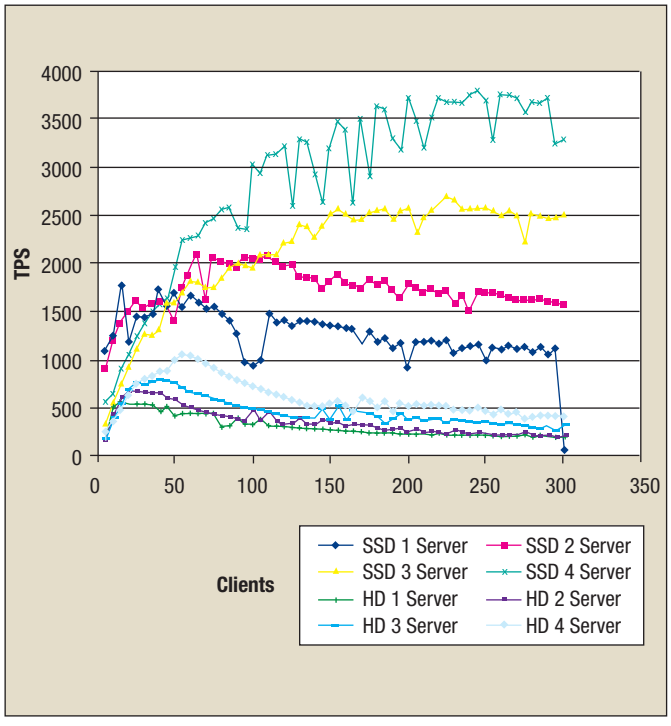
Welke componenten naar een SSD verplaatsen?

Het beste kan op operating systeemniveau de IO wait time worden geïdentificeerd. De daarvoor beschikbare tools verschillen per type operating system. Voor het Microsoft OS is de beste tool de 'Performance Monitor'. Alhoewel het niet de eigenlijke IO wait statistieken weergeeft, geeft '% Processor Time' een indicatie van de CPU-bezetting. TMS beveelt aan om op de fysieke disk de 'Average Disk Queue Length' en 'Disk Bytes per Second' te analyseren voor mogelijke bottlenecks in het disksubsysteem. Voor UNIX systemen zijn de commando's top, iostat, vmstat en sar bruikbaar. Het top commando laat de '% iowait' voor het systeem zien. Vanaf Oracle versie 8.1.7.2 wordt de Statspack utility meegeleverd om de databaseprestaties te meten.

Heeft men eenmaal vastgesteld dat het IO subsysteem de oorzaak is van de prestatieproblemen, dan is de volgende stap om te bepalen welke componenten van de Oracle database de hoogste IO genereren en IO wait time veroorzaken. Daarvoor komen de

volgende databasecomponenten in aanmerking: de complete database; Redo Logs; Indexen; Temporary Tablespace, Rollback Data en frequent opgevraagde tabellen.
Er zijn databases die al hun files op een SSD zouden moeten plaatsen. Dergelijke databases moeten tenminste een van de volgende karakteristieken bezitten: een hoge concurrent access; frequente random toegang tot alle tabellen; kleine tot middelgrote databases en grote leesintensieve databases. Bij zeer grote enterprise databases is deze optie economisch gezien natuurlijk niet haalbaar. Redo Logs zijn een van de meest bepalende factoren voor de write prestaties van Oracle databases en zijn een constante belasting voor het IO systeem. Het is daarom belangrijk om de redo logs op de snelste disk te zetten.
Vanzelfsprekend is het schrijven van een redo log naar SSD een logische manier om de algehele databaseprestaties te verbeteren.

Wanneer veel gebruikers gelijktijdig een index lezen dan vinden er op de disk drive frequente, kleine, random transacties plaats. Onder dergelijke omstandigheden kunnen HD's vaak niet aan de vraag voldoen en treedt er IO wait time op. Door de indexen op een SSD te plaatsen kunnen de prestaties van een applicatie verbeteren. Met name voor OLTP-systemen met een hoog aantal gelijktijdige gebruikers kan dit resulteren in een snellere toegang tot de database.
De Temporary Tablespace wordt voor complexe sorts, hash en bitmap indexoperaties gebruikt. Omdat tablespaces voor verschillende typen operaties worden gebruikt, kan de inhoud snel gefragmenteerd raken. TMS heeft in interne testen geconstateerd dat de prestaties van de Oracle database snel afnemen naar mate meer data worden gefragmenteerd. Alle reden om de Temporary Tablespace in een SSD op te nemen omdat fragmentatie daarbij geen rol speelt.
Bij databases met een hoog aantal gelijktijdige gebruikers kunnen rollback segmenten (undo tablespace in nieuwere versies) de oorzaak zijn van contention. Undo data worden gecreëerd elke keer dat in een Oracle transactie een record wordt gewijzigd. Omdat de undo tablespace bij elke wijziging wordt aangesproken, is het zinvol om ook deze op een SSD op te slaan. Het maakt een snelle write operatie mogelijk bij een update transactie en zal undo tablespace sneller beschikbaar maken voor de volgende operatie.
Tenslotte iets over frequent bezochte tabellen. De schatting is dat slechts 5 tot 10 procent van de opgeslagen data bij OLTP-systemen veelvuldig worden geraadpleegd. Deze tabellen dragen bij aan een hoog percentage database-activiteit en dus IO naar storage devices. Wanneer een tabel vaak wordt geraadpleegd dan treedt er een transactiequeue op zodat andere transacties moeten wachten en dit is dus een teken dat er sprake is van IO wait time. Het is dan verstandig om de vaak bezochte tabellen op een SSD te plaatsen.



Afbeelding 4:Vergelijking prestaties HD en SSD met TPC-C benchmark.

Test HD en SSD met TPC-C benchmark

Een door TMS destijds uitgevoerde Oracle TPC-H en TPC-C toont de SSD testresultaten van deze benchmarks. De test betrof een Oracle 11g die op vier Linux servers draait en via Fibre Channel verbonden is met twee RamSan-400's, een enkele RamSan-500 en twee Fibre Channel RAID 5 array's met elk veertien HD's. De totaal benodigde tijd voor de benchmarks bedroeg ruim 5 uur bij HD's, met SSD's slechts ruim 7 minuten. Daarmee bedroeg de algehele prestatiewinst een factor vijf. De belangrijkste winst werd gehaald in de query-gebaseerde transactiebelasting, een factor 25 of beter in vergelijking met HD's.

Een grote zorg voor alle managers, DBA's en gebruikers zijn de waarden waarbij een systeem zal pieken in het geleverde vermogen om transacties te verwerken versus het aantal ondersteunde gebruikers. Een methode om dit aan te tonen is de TPC-C benchmark. In afbeelding 4 zien we de testresultaten. Daarin valt af te lezen dat een HD maximaal 1.051 TP's levert bij 55 gebruikers en de SSD 3.775 TP's bij 245 gebruikers. De HD resultaten beginnen al bij 1.051 TP's met 55 gebruikers af te nemen, bij SSD pas bij respectievelijk 3.775 en 245. Zelfs bij een 1 node server presteert het SSD-gebaseerde systeem beter dan een HD-gebaseerd systeem met 4 nodes. En dan valt te bedenken dat de TPC-C benchmark op basis van de RamSan-500 werd uitgevoerd die 'slechts' 100.000 IOP's levert. De RamSan-630 levert het tienvoudige daarvan!
Het is duidelijk, de SSD is de beste keus om het IO wait probleem bij databases op te lossen. Het garandeert de hoogste databaseprestaties.
Informatie op Internet: www.ramsan.com

Bram Dons is onafhankelijk IT-consultant.