RESEARCH NOTE

BY RICHARD WINTER

WinterCorp
www.wintercorp.com

WINTER CORPORATION

Autonomous Data Warehouse

Oracle's Self-Driving Database

Conclusion: Oracle's Autonomous Data Warehouse (ADW) provides unique capabilities. It is remarkably simple to get started with ADW, to load data and query it. ADW promises a major cost savings for many data warehouse workloads, both in comparison to on-premises operation and in comparison to the widely promoted "cloud first" and "cloud only" data ware-house offerings. Oracle ADW has functionality, uptime and elasticity features not matched by Redshift and similar products.

ADW is a new offering, which will only become fully autonomous for the most demanding requirements over a period of time. But, the author believes that 70% or more of data warehouse workloads can be beneficially moved to — or implemented on — ADW in the near term. Customers should start with simpler data mart requirements and only gradually work up to more demanding applications. Not all workloads and databases will be better off in the cloud, so it will be important to monitor cost and performance tradeoffs as the requirements increase.

WinterCorp recommends that customers begin experimenting with Oracle ADW without delay to experience the simplicity, speed to market and cost savings available. This recommendation is based on an analysis of ADW capabilities and supported by hands on experimentation with ADW by the author. ●

ORACLE HAS ANNOUNCED Autonomous Data Warehouse Cloud (ADW), a service whereby a user can rapidly create a data warehouse, load it with data and put it into use, typically without specialized skills. The author has attended the announcement, analyzed the capabilities and experimented with them hands on.

What ADW Actually Is

Oracle ADW is the combination of three things:

1. Oracle Database 19c, a new version of the product, featuring expanded database automation;
2. Oracle Cloud, a database-optimized infrastructure-as-a-service, cloud offering; and,
3. Policy-driven automation and machine learning.

Previously, creating a data warehouse required skilled customer personnel to perform a complex series of tasks such as provisioning and configuring a server; installing database software; allocating storage; creating a database; selecting indexes; choosing partitioning methods; making other physical design decisions; and, correctly setting a good number of parameters.

About WinterCorp

WinterCorp is an independent consulting firm focusing on analytic data management at scale.

Since our founding in 1992, we have helped customers meet the largest and most demanding data requirements.

Our expertise encompasses leading commercial products and open source products, on-premise and in the cloud.

Our services help customers define their data strategies, architect and engineer their solutions, select their platforms and manage the growth of their databases and systems.

WinterCorp
www.wintercorp.com
TYNGSBORO, MA
617-695-1800

Oracle's goal with ADW is to have the system do all this automatically. To create a data warehouse, the customer provides a logical schema; loads the data; and, defines who may access it. At that point the data warehouse is ready for use — and the DBA has been saved a great deal work.

This is dramatically simpler than the equivalent process with past versions of Oracle Database — and with other widely used on-premise database products.

More significantly, Oracle promises that ADW, via automation and machine learning, will continue to provide the physical administration of the database over its lifetime. ADW will automatically deal with such issues as growth and changes in usage patterns; growth in data; and, the simpler forms of growth in the schema, via the addition of new tables.

Significance of ADW

The most significant aspects of ADW for most customers will be cost reduction, simplicity of operation, increased data security and increased uptime. For many data warehouse requirements there will also be a large reduction in the skills required.

The single largest cost in most data warehouse programs — often 30% to 50% of total cost — is the labor required for physical database design, administration, operation and software maintenance. These costs either disappear or are greatly reduced with ADW. People performing these duties will be freed up to focus on logical database design and administration, data modeling and similar activities that add more business value. In most cases, users will require less IT help to use the data for query, reporting and analytics.

Security and Maintenance. Oracle does all required maintenance on ADW systems. In comparison to on-premises operation for most Oracle customers, this is a large advantage. Oracle data shows that most customers do not apply patches in a timely way to databases running in their own facilities. These late or absent patches create security vulnerabilities, operational problems and performance problems. For ADW, Oracle applies all patches at the optimal time, without interrupting the customer's access to data. Both the timeliness of software maintenance and the rigorous security practices that Oracle follows in its cloud contribute to a data security environment that few, if any, customers can match.

Exadata Hardware. Customers run the Oracle database on a wide range of hardware configurations, leveraging its ability to run on almost any server. But Oracle data warehouses run best on Exadata, the hardware/software combination that was engineered specifically for Oracle database. ADW runs exclusively on Exadata hardware, which provides efficiency and highly parallel data storage I/O

• Mature Enterprise Data Warehouse Utility not Matched by "Cloud First" or "Cloud Only" Databases
• Full Oracle SQL Including Stored Procedures
• Continuous Availability
• Real Elasticity Without Interruption of Service
• Independent Scaling of Storage and Server Capacity
• Autonomous Operation
• Automatic Software Maintenance without Interruption of Service
• Autonomous Statistics Maintenance & Tuning

Table 1:
Key Capabilities of Oracle ADW

that many customers and cloud providers cannot match. This is unique to the Oracle cloud.

Statistics. Database optimizers make use of statistics about the data when creating a query plan. These statistics are invisible to the user. Unfortunately, most customers do not keep the statistics up to date, which hurts query response time and wastes system resources. But, on ADW the policy-driven automation makes sure that statistics are updated consistently. That alone has a major impact on query performance, by ensuring that the plan used to execute the query is nearly always the most optimal plan that the Oracle database can generate.

Self-Tuning. The up-to-date statistics coupled with machine learning provide the opportunity for automated tuning with ADW. Thus, if queries begin to run more slowly because the workload or the data have changed, ADW is intended to make changes automatically.

Elasticity. AAll cloud vendors promote the elasticity of resources in the cloud: the idea that you only pay for resources when you need them. When you face peak demand, you can expand the configuration on which you run your workload.

But, on most cloud offerings this elasticity is available only job by job. If you have a data warehouse that must always be available for queries, then the data warehouse runs all the time as a single job. To scale up or down, typically you must take the data warehouse down; reconfigure; then bring it back up. This interrupts access to the data and service to the users of the data warehouse.

With ADW, you can have continuous uptime with elasticity. That means you can expand your configuration to accommodate peak demand without interrupting service. You can also contract the configuration after the peak has past. This provides meaningful elasticity with the intended cost savings, even for a continuously busy data warehouse. Finally, with ADW, you can scale compute resources up or down independently of data storage. This can provide a considerable cost advantage, for example, when the data volume is large but the workload is light.

Complete, mature data warehouse function. Complete, mature data warehouse function. Unlike the more widely advertised cloud data warehouse products, Oracle enjoys an advantage due to its maturity. ADW is the same Oracle database that customers run on-premise. It includes the complete Oracle code base that has been in use on a very large number of production data warehouses for years. It has a robust complement of functions and features that only mature from widespread use and years of incremental development. Thus ADW includes many features of SQL, including stored procedures, which are not available on RedShift and other cloud database products. In addition, via its maturity, ADW protects users from the maddening flaws that make real projects difficult to implement on many recently created cloud products.

An ADW Test Drive with Real Data

Having attended the announcement of ADW, I wondered what it would be like to actually use the service. From direct experience, I know that it takes considerable work and expertise to set up even a modest data warehouse on-premise on any major data warehouse platform. But with ADW, anyone can open an Oracle cloud account and get $300 of services free in order to try the service. I decided to try it myself.

Getting Set Up. It took me a few minutes to open the free account and create an instance of ADW in the Oracle cloud. I took all the defaults, so I got an Oracle database server with one processor and one TB of data storage. I now had an Oracle database into which I could load data, having done everything to this point with the Chrome browser on my Macbook.

I had some previous experience with Oracle SQL Developer, a tool with a graphical user interface for the knowledgeable SQL user. I downloaded a free copy and followed Oracle's instructions on how to connect it to my autonomous data warehouse in the cloud.

Loading Real Data. Wanting to use real data that I selected myself for my experiment, I downloaded 1.8 million

• Easy to Sign up for the Service and Provision a Data Warehouse
• Easy to Load Data
• Continuous Availability
• Data is Immediately Usable — Easy to Create and Submit SQL Queries
• Subsecond Response to Simple Queries (single user)
• Good, Scalable Response to More Demanding Queries
• Easy to Scale # of Processors
• 32x Linear Scaling Measured on One Large Join

Table 2:
Key Points about Oracle ADW Validated in Hands On Use

provider records and 11 million payment records from www.cms.gov, the website of CMS, the US federal agency that administers Medicare. This was real Medicare data, albeit sanitized to eliminate patient names.

The data provided by CMS was text data in the form of "csv" files (each field value is separated by a comma from the next value) with headers. If SQL data definitions were available, I didn't find them. But, no matter, by clicking on "Import Data," I was able to get Oracle SQL Developer to load data in this format directly into the data warehouse. It uses the column names in the header line of the input file and scans enough of the data to guess a data type and length, then loads the data.

I did need to navigate around a couple of problems at this point. The first was that the payments data had some text fields with embedded commas, a violation of the usual csv format. Ordinarily embedded commas should be quoted or preceded by an escape character. But since CMS had not done so, Oracle could not parse the records. I replaced the commas with hyphens using the TextEdit program on my MacBook. The text fields were now still readable and Oracle was able to load these records.

The second problem was that Oracle could not always tell how long a field was going to be from its automatic examination of the first 2000 records. So the data definition it automatically generated sometimes specified an insufficient maximum field length. To overcome this, I increased the maximum field lengths in the table definition manually. This second problem would not occur in practice if a schema for the input data was available, the more common situation for a substantial data warehouse input file.

After making these two adjustments — one to the input data and the other to the definition of the table to be loaded — the loading concluded smoothly. The provider file, which was smaller in total data size, loaded in a few minutes. For the larger payments file, totaling 6.6 GB uncompressed, I loaded in increments of 100,000 records. These took a few minutes each. This was while using the least efficient option available for loading data: importing desktop files via SQL Developer. Without much more effort, I could have transferred the files to Oracle Cloud Storage and loaded from there with Oracle SQL Loader, which would have been much faster. Had I selected a larger configuration with multiple processors, the load would also have been faster. For this small experiment, I opted for the ultimate in simplicity — a choice that will work fine for users with smaller data volumes.

Online Query. I then spent several hours making up SQL queries and running them. I cooked up a wide variety of queries of simple-to-moderate complexity. These included SELECT statements with aggregates such as SUM and COUNT; grouping queries; sorts; joins and other constructs. As one would expect, Oracle handled them all without difficulty.

I did expect Oracle to complete all my queries, but I didn't expect the very fast response I got. Even with the single processor configuration that I took by default — the smallest available — response time for my experiment was excellent. For the simplest queries, it was a under a second. For others it took a few seconds but was always pleasingly quick.

Query at Terabyte Scale. Oracle has pre-loaded a terabyte of data into ADW, so that customers can have the experience of experimenting with the new service at that terabyte scale.

The data preloaded by Oracle is of the sort that is often generated for benchmarks: there are five linked tables

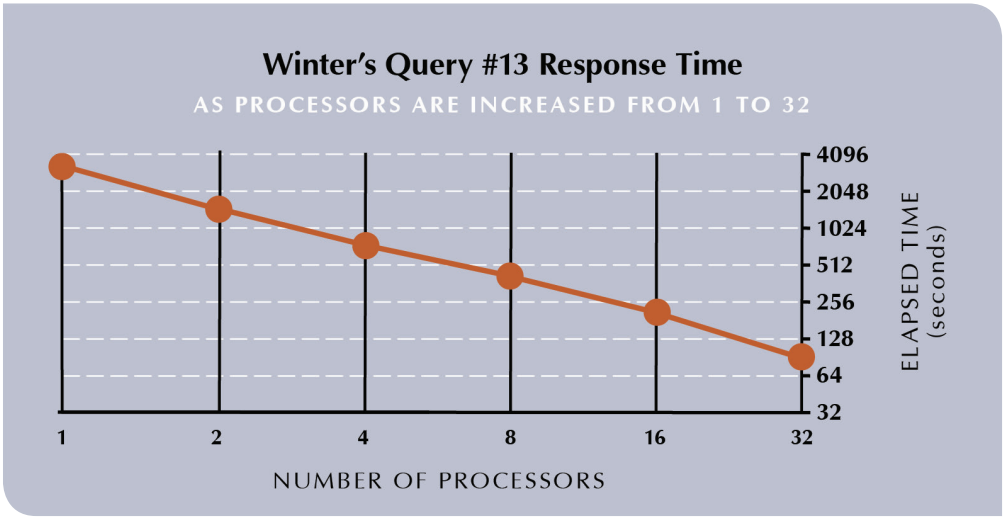

Chart 1:

ADW response time for my batch-type query #13, a large join, decreases rapidly as processors are added — a better-than-linear speedup as processors are increased from 1 to 32. Chart is log-log.

patterned after data that occurs in many real businesses. The tables are line items (from orders), customers, parts, suppliers and dates. The largest table contains about six billion line items and itself contains just under a terabyte of data.

To experiment at larger scale, I created and ran some 35 SQL queries, doing more or less anything that occurred to me. I ran most of these queries with a single processor. I started with simple queries, such as asking how many records were there in the lineitem table? It took 8 seconds to get an answer. How many were there by date for each of the 2556 dates in the six billion row table? I got an answer in 56 seconds. What was the total revenue of all the six billion line items. This took 162 seconds. These were fast, I thought, for queries employing a single processor to touch in sequence the equivalent of six billion facts. Probably this last query benefitted from Oracle's hybrid columnar compression.

I then played around with more demanding queries using more processors. For my thirteenth query, I asked ADW to build a new table containing all the columns of the lineitem table and the date table, joined, for all the orders that were received on the 15th day of any month in the 7-year period covered by the data. This resulted in a table of 197 million rows of 34 columns each. From ADW statistics, I could see that the average row was 200 bytes long. Therefore the new table consisted of about 40 GB of data extracted non-contiguously from the terabyte lineitem table and joined to data from the date table.

I want to stress that this was not an interactive query. Rather it is a workhorse query of the sort used to create the inputs for a large analysis or perhaps for machine learning. This is the kind of query that you start — and then go off to do something else.

With one processor, it took ADW 3,524 seconds to complete this sizable operation. I then repeated the query six times, each time doubling the number of processors. The result was a slightly better than linear speedup, as shown in *Chart 1*. At 32 processors, the response time was down to 87 seconds, about 40 times faster than with one processor: better-than-linear scaling over this range.

Since I was simulating building a table to be used in say, in a large analytic process, I thought that 87 seconds was fast enough — and stopped. I could have continued increasing the number of processors from 32 to as many as 128, had it been important to seek yet faster response. So, here I had some evidence, at least for one sizable join query, of the proposition that it is easy to add processor power and get faster query performance as a result. I then went on to do larger joins (for example, joining all five tables for a billion of the lineitems, creating a result table of about 100GB). In this case, too, I was able to obtain linear speedup by increasing the number of processors.

A Spin Around Town. So this test was like taking my friend's new car — ADW — out for a spin around town. You don't learn everything about a car in such a trip, and it is not systematic testing, but you get a sense of what the car is like. And I liked ADW. Overall, the experience was dramatically simpler than I expected. I had created and loaded two data warehouse tables with real data and begun to query them within a few hours. If I was doing this for a real customer, I have little doubt that I could rapidly get a realistic data warehouse of small-to-moderate proportions into use and delivering value. In a real data warehouse with many tables, there would be substantially more data preparation, but tools are available for this.

In addition, I created and ran 35 queries against the terabyte of data Oracle provides with the ADW trial. Simple queries were handled with good response. More demanding queries took time, of course, but were readily sped up by scaling up the ADW configuration. In the two cases where I did this, the speedup was at least linear. This gave me confidence that ADW will prove usable for data well into the terabyte range. A word of caution: much more rigorous and testing and experimentation will be required to assess ADW for larger scale and heavier use. It is unlikely that ADW can presently handle autonomously the demands of the largest, most complex or most heavily used data warehouses. Not every query can be sped up by allocating more processors. In my testing I did not go far into the issue of high query complexity at scale.

Here is what I learned from my hands-on experiment: it was easy to create real data warehouse tables with ADW, load data and run queries. Nothing was difficult and using the product felt good. Will ADW live up to all that Oracle is promising for it? You can't tell from the tests I ran. But ADW is off to a good start. **Customers ought to start working with ADW without delay.** ●