

# Real-time Recovery Architecture as a Service

by David Floyer | Friday, September 2nd, 2016-Wikibon.com

---



[Wikibon.com](http://Wikibon.com)

## Real-time Recovery Architecture as a Service

by David Floyer | Friday, September 2nd, 2016

<http://wikibon.com/real-time-recovery-architecture-as-a-service/>

# Real-time Recovery Architecture as a Service

by David Floyer | 2 September 2016 | Analysis, Blog, Infrastructure

Contributing Analyst: David Vellante

## Premise

Backup is one thing; recovery is everything. Today most backup & recovery environments are still on-site and batch orientated, and in general are built round a backup application (e.g., Oracle RMAN) and batch-orientated purpose built backup appliances (PBBAs, e.g., EMC Data Domain). The emphasis has been on speed of backup completion rather than speed of recovery.

In a cloud-centric world, flexible recovery mechanisms both onsite and in the cloud will be an imperative, and the backup & recovery processes will need to radically adapt to be real-time and offered as a service. Aggressive Recovery Point Objectives (RPOs\*, see Footnote 1 below for definition and discussion) are a prerequisite for rapid and flexible Recovery Time Objectives (RTOs\*, see Footnote 1 below for definition and discussion). Without very low (close to zero) RPOs, RTO recovery times are longer and uncertain.

Wikibon believes that batch backup appliances (PBBAs) based on the storage system will give way to real-time, continuous data protection systems that work at the application memory level and memory speed. This will allow very aggressively RPOs that are very close to zero, measured in milliseconds rather than hours. In turn, this will enable very rapid recovery either onsite or in the cloud. Wikibon believes such systems will radically simplify recovery and will be offered as a service in both private and public clouds.

Research note: there are two main systems of storage-level database protection in production today. The first is the backup appliance discussed above, and is the subject of this research. The second is storage replication technology, which will be the subject of future Wikibon research investigation into the shortcomings of synchronous replication in real-time low-latency environments.

## PBBAs and their Useful Role

For decades, backup processes have remained static as a serial batch job designed around tape-based systems. Virtual tape libraries leveraged spinning disks, but were too expensive to keep much backup data on disk. The genius of PBBAs, popularized by Data Domain, was they kept existing batch backup processes in place and hence the

migration was non-disruptive. They used data de-duplication techniques to dramatically reduce the cost of backup disk storage. PBBAs effectively eliminated tape as a primary backup medium and relegated tape to archiving, last resort disaster recovery uses and compliance requirements.

Current best practice is to exploit application consistent snapshot technology and push delta changes multiple times per day to PBBAs. The PBBA then allows asynchronous transmission of the reduced data to another PBBA in another location, improving RPO relative to the traditional once a day backup window. Claims have made that this could, in theory, become a continuous system, but this is both theoretically and practically impossible, as they are based on the application state at the storage level, rather than the application state in memory.

In summary, PBBAs provided a much better way of performing the traditional batch backup and recovery in the traditional data center. However, it has major deficiencies as enterprise IT becomes more cloud-centric and businesses require much more aggressive recovery SLAs.

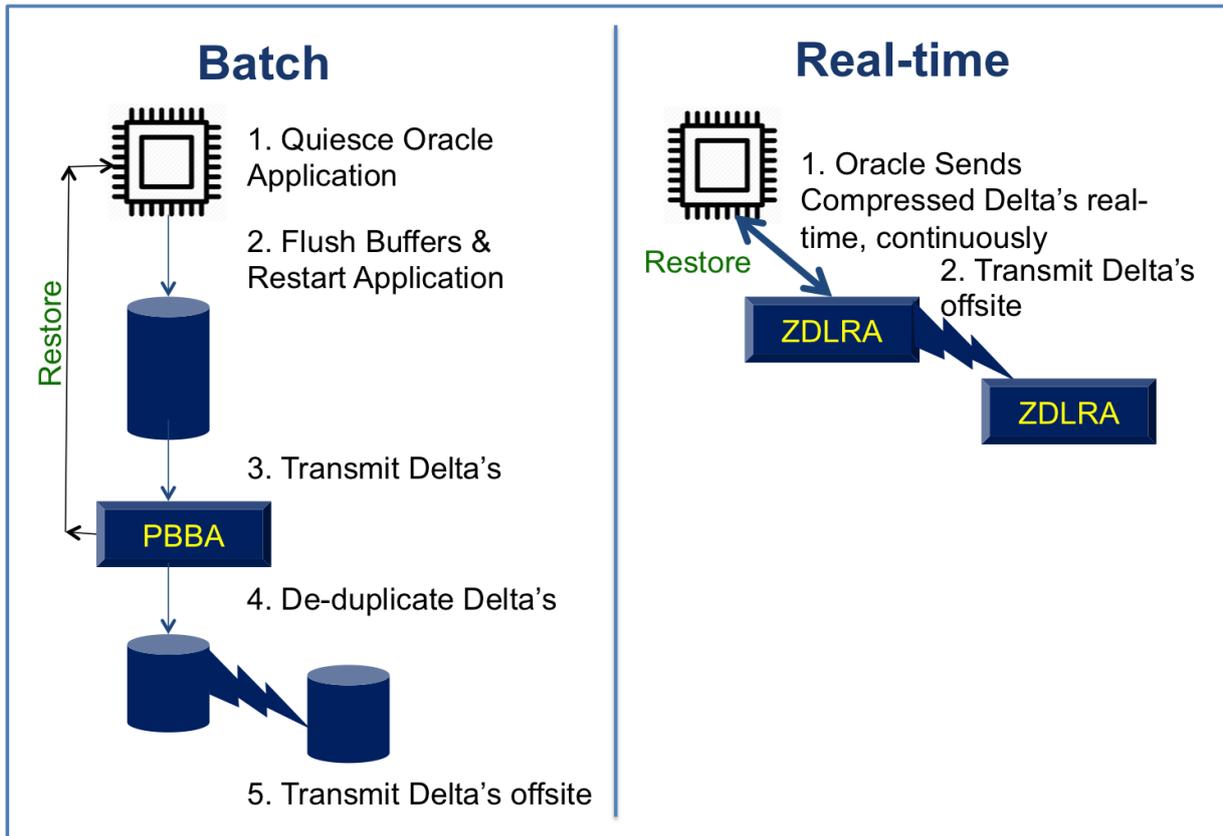
## The Deficiencies of PBBAs

The PBBAs still fundamentally rely on a batch process and practically speaking can't approach anywhere near RPO zero. PBBAs require a serial process that is both cumbersome and time-consuming. Relative to modern methods of backup, we believe these systems will increasingly become outdated for customers requiring close to RPO zero and wanting to include the cloud for recovery processes. Consider the following steps required to backup data using a PBBA:

1. The application must be quiesced;
2. The database buffers are flushed and a snapshot (or equivalent) of storage is taken;
3. The application is restarted and all of the delta changes are pushed to the PBBA;
4. These changes are de-duplicated and stored onsite within the PBBA;
5. The data is then replicated offsite to another PBBA in a second data center or in the cloud.

The left hand side (Batch) of Figure 1 below illustrates the steps required to protect the data using the PBBA techniques. This is sometimes implemented as a single command, but the steps within the command are still batch processes.

Other deficiencies of PBBAs are that individual applications and databases cannot change their RPO and RTO requirements easily. Real-time detailed knowledge of the backup and recovery status at an application and database level is not available within a storage-centric PBBA. Only the application and database or file system can create real-time protection and monitor the complete protection status in real-time.



**Figure 1: The Evolution of Data Protection from Batch Storage Paradigm to a Real-time Application-based Paradigm**

Source: © Wikibon 2016

### Real-time Recovery Architecture

Real-time protection systems has been in production for many years, from Oracle and other vendors (see section "Other Database Vendors" below). Oracle offers Active Data Guard for local and remote failover, and GoldenGate for full system active-active topologies. These are high-function but expensive features and configurations (e.g., Active Data Guard is \$11,500/core, plus an additional standby system).

The ZDLRA (Zero Data Loss Recovery Appliance) is an Oracle appliance and uses the same mechanism as Active Data Guard to capture compressed database log files in real-time directly from memory and send them to the recovery appliance. The ZDLRA operates with an RPO of milliseconds, and organizes the log files and metadata for fast restore by RMAN locally, remotely or in a cloud. By taking RMAN incremental forever backups, the time to recover from the ZDLRA is minimized. The ZDLRA offers a much lower price point and is competitive with PBBAs, with no requirement to pay for additional Oracle database licenses or features.

The right hand side of Figure 1 above shows how ZDLRA works. The real-time process is implemented at the memory level, and achieves RPOs measured in milliseconds.

1. The database (with real-time redo transport enabled) generates compressed redo changes in memory and transfers them directly to the ZDLRA in milliseconds. The changes are staged, validated and the RMAN recovery catalog updated.

2. Replicated changes are sent to a downstream ZDLRA, managed with OEM Cloud Manager.

The ZDLRA hold the metadata and the changes for optimal recovery by RMAN, without the DBA needing to know the physical location of any data. Compared with traditional PBBAs, ZDLRA offers RPOs measure in milliseconds, faster recovery and integrated management tracking of the recovery status of all databases. [More information on how the ZDLRA works can be found in this research note.](#)

As an example, if you run the described backup process on a PBBA twice per day (i.e. every twelve hours) the average RPO exposure will be about six hours. This compares to an real-time recovery architecture, which is continuous with a RPO of milliseconds.

Future applications are increasingly real-time, exploit memory-intensive computing, require low-latency characteristics and live in cloud-defined environments. Processor technology clock speeds are not improving, and the number of cores in a processor is increasing dramatically. All these factors are dramatically increasing the amount of data held in application state in memory. This increases the challenge of moving data from application state to storage state, and the challenge of refreshing memory in the case of a system restart. As applications become more real-time oriented and the amount of data in memory grows, the application state becomes more out of line with storage state. These future systems will require real-time recovery mechanisms as a service

## Other Databases

Systems from other companies such as [Dbvisit Standby](#) also use the Oracle real-time log files. IBM's DB2 has the High Availability Disaster Recovery (HADR) feature, which can use log shipping at the log buffer level. Microsoft SQL Server and MySQL have log shipping, but in both the logs are part of the storage system.

Oracle's Golden gate also supports IBM DB2, Microsoft SQL Server, and MySQL and sources and targets. Supporting databases other than just Oracle products would boost ZDLARA and Active Data Guard potential to take on PBBAs.

## Friction to Real-time Recovery

Practitioners should note that even with ZDLRA (and other methods touting zero data loss), the probability of some data loss, while minimized, still exists. For example, if a catastrophic disaster occurs before data is shipped off-site to a distance protected from the local disaster, some data will be lost. While there really is no such thing as zero data loss, in our view, the ZDLRA gets as close to "RPO Zero" than any other solution at a significantly lower price point.

As stated before, the genius of PBBAs was they could be implemented with very little change to the batch tape ecosystem. The migration to real-time log file transfer will require some changes to the existing processes and procedures. For Oracle databases, the change is not severe, and the benefits very real.

## Conclusions

Purpose-built backup appliances served to dramatically transform backup while at the same time allowing for non-disruptive adoption of a superior technology (relative to tape). Indeed the market continues to show growth today, primarily due to software that supports cloud and DR. However for customers demanding close to RPO zero, simplified recovery and improved recovery performance (especially protecting multiple databases with a single appliance),

Wikibon projects that PBBAs will increasingly give way to more integrated, memory-based approaches that are application-led. The ability to have different recovery SLAs by application, and monitor the exact level of protection at any time offer much greater flexibility than PBBAs. In time, Wikibon expects these Data Guard-like real-time approaches will subsume the classic storage-level database log files, to the joy of DBAs everywhere. Wikibon also expects that all high-performance high-availability databases and files systems will adopt a similar approach to data protection.

As was said at the beginning of this research: backup is one thing; recovery is everything. Approaches to protecting data are being driven by new application requirements at cloud scale that demand new real-time methods and new levels of availability. Wikibon expects that ISVs and cloud providers will be moving aggressively into real-time recovery architectures and recovery as a service. Wikibon also expects that all high-performance high-availability databases and files systems will adopt a similar approach to data protection.

Oracle is well positioned to lead this migration to real-time recovery architectures as a service, with RPO in milliseconds, a full range of RTO options and price points, and support for private and public hybrid clouds.

### Action Item

***Practitioners requiring close to RPO zero and aggressive RTO SLAs should plan for an integrated data protection approach that effectively eliminates the concept of storage-led backup and shifts thinking to a recovery as a service model. The database and file system vendors and some cloud vendors will be the initial predominant suppliers of this technology, and understanding their roadmaps and commitment to support real-time recovery as a service strategies is crucial to both database and data protection technology selection.***

### Footnotes

#### ***\*Footnote 1: Defining RPO and RTO***

*Recovery point objectives and recovery time objectives are IT terms used to describe (respectively), the amount of data you're willing to expose to data loss prior to your next recovery point and the time it takes to bring your applications back online. Specifically:*

*The **Recovery Point Objective** (RPO) for an application describes the point in time to*

*which data must be restored to successfully resume processing (often thought of as time between last backup and when a data loss “event” occurred). A key part of the process is to ensure that the data is also sent offsite, to eliminate an onsite “event” destroying data. RPO defines the amount of data that’s at risk of being lost and the business must accept whatever risk is present or make a change to reduce data loss exposure. The theoretical (and impossible) goal for mission critical applications is RPO zero—i.e. zero data loss, or at least zero the majority of the time.*

*The **Recovery Time Objective** (RTO) for an application is the goal for how quickly an organization needs to have an application’s data accurately restored and available for use after an “event” has occurred, where the event is one that restricts access to application data. A key component of RTO is dealing with lost data, and low or near-zero RPO is a prerequisite for low RTOs. The event may necessitate recovery onsite or in the cloud.*

*RPO and RTO are useful measures to determine what technologies, products, processes and procedures are required to meet those targets, which should be set through a business impact analysis (BIA). The BIA ties application unavailability to the consequences of data loss and the budget available to meet these objectives.*

*The ROI impact of reducing RPO and RTO equals the reduction in expected loss from an event (i.e. the probability of an event times its business impact) divided by the cost of achieving that benefit.*

*ROI (Value) = Reduction in expected loss / the cost of achieving that reduction.*