Why an Active/Passive Business Continuity Solution is Not Good Enough

Keith Evans >> Shadowbase Product Management >> Gravic, Inc.

The costs of prolonged downtime of critical business IT systems are significant (potentially to the point of shuttering the company). These potential costs are compounded by the fact that the many events which can lead to such outages are not rare; it is a case of when, not if. This likelihood of outage events is only acceptable if you have a complete, documented, and well-tested business continuity plan in place. Maybe you think that you do, but the data does not support this idea. Many companies are operating with the mistaken belief that their business continuity plan will work when the time comes, or even if it does work, that the plan is good enough to prevent significant consequences to the company. Read further to find out whether this false sense of security applies to you.

Business Continuity Architectures: Pros and Cons

The chosen availability architecture is the primary factor in determining how effective your business continuity plan will be when the time comes. To discuss the typical ones, we first need to understand a couple of terms:

- *Recovery Point Objective* (RPO) is the maximum acceptable amount of data loss arising from an outage of an active system. In practice, it is the data updated in the period between the last time the data was saved to (remote) recoverable media, and the point of failure.
- *Recovery Time Objective* (RTO) is the maximum acceptable time for recovery from an outage. In practice, it is the period between the time of failure and the point at which services are restored to an acceptable level.

Different business continuity architectures have different attributes with respect to RPO and RTO (Figure 1). Let us briefly review each of these major architectures.[1]

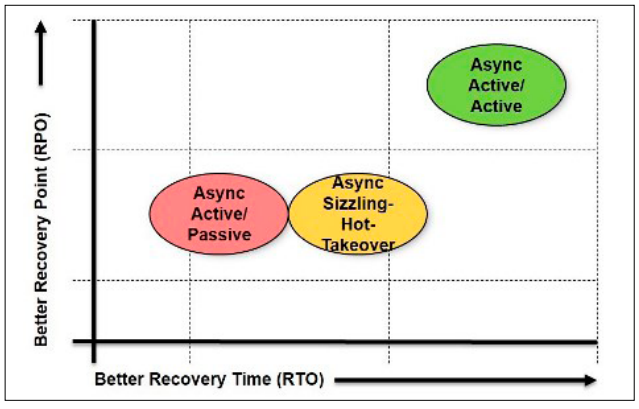

Figure 1 – RPO and RTO for the Various Business Continuity Replication Architectures

Active/Passive – Classic Disaster Recovery

In this architecture, all transactions are executed on a single system (the active node), and the database updates are replicated by Shadowbase software[2] to a backup system (the passive or standby node). In the event of a failure of the active node, a failover to the backup node is executed, the applications are brought up with the local (synchronized) database opened for read/write access, users are switched to the backup node, and processing resumes. This architecture and failover sequence are by far the most common, but are also the most flawed.

The key issue with this architecture is that it is *very difficult to test the backup node and failover procedures*. Proper testing requires an outage of the primary node and may take a long time. Therefore, failover testing is very often not performed at all or not to completion when it is attempted (because it may take longer to fully test than the available outage window).

It is also possible that restarting the production system after the test has completed may not work, which is another reason why testing may be avoided. Because of this lack of testing and the resulting uncertainties surrounding the state of the backup system and the takeover procedures, when a real outage occurs, management is often slow to initiate a failover in the first place, further delaying recovery. Hence, this architecture is risky, the state of the backup system (and procedures to failover) are not really known, failover faults are likely to occur causing the failover to be unsuccessful, or at least take a long time. For all these reasons, this architecture has the probability of a high RTO, often several hours or even days. While a basic active/passive architecture offers some protection, it is by no means the best solution. It should really only be considered as a starting point, or used for non-mission-critical applications.

Active/Almost-Active – The Sizzling-Hot-Takeover (SZT) Architecture

While it looks almost the same as a classic active/passive architecture, *sizzling-hot-takeover* (SZT) has one major difference which makes it a much improved solution. The difference is that while all transactions are still routed to a single active node, the backup node has the applications already up-and-running, with the local database open for read/write access.[3] The key benefit of this ability (versus an active/passive architecture) is to ensure the backup system is ready to go when

[1] Note that each of these architectures use asynchronous replication, where there is a slight delay between when the data is updated on one system, and is safely replicated/stored on another system. This delay accounts for the data loss in the event of an outage.

[2] For a much more detailed description of the various business continuity architectures and their total cost of ownership, see the Gravic Shadowbase white papers, Choosing a Business Continuity Architecture to Meet Your Availability Requirements and Fingers Crossed? Or What is Your Business Continuity Plan for the Inevitable?

[3] Not all data replication products allow the backup database to be open for application read/write access during replication, but Shadowbase software (from www.gravic.com/shadowbase) has no such restriction.

you actually need it. Since the applications are up-and-running on the backup node with the local database open read/write, it is easy to send test/verification transactions against test/verification accounts to validate the backup system at any time, with no impact to the active system. Hence, the backup system can be regularly validated, and becomes a known-working system. (For all intents and purposes, it is a fully active system, with the exception that it is not processing online transactions.)

When an outage occurs of the primary node, the decision to fail over can be made immediately, with confidence that failover faults will not arise, and the failover will succeed quickly. As a matter of fact, as a best practice, failovers should be performed regularly (e.g., weekly or monthly) to test the process and build the confidence of the staff in performing them. Businesses running active/passive architectures simply do not have the same confidence level as those running SZT architectures. Therefore, this architecture gives a much better and repeatable RTO versus classic active/passive architectures. SZT is the minimum level of business continuity solution which should be employed for mission-critical applications.

Active/Active – Partitioned

In a partitioned active/active architecture, the applications are active on all nodes, transactions are routed to all nodes, and each node has a copy of the database, which is kept synchronized by bi-directional data replication. To avoid data collisions[4], for example, the data (or requests) are partitioned so that transactions are routed to a specific node based on some key in the data, or from which user the transaction originated. The database may be split by customer name, and all transactions for customers A-M are executed on one node, and customers N-Z on the other, with their changes being replicated to the other node to keep the databases synchronized. This architecture provides the key benefits of active/active while avoiding data collisions.

The benefits of this architecture compared to classic active/passive and SZT are:

- On failure/outage, only half the users (in a two node configuration, fewer if more nodes are used) are affected and have to be switched. The other half of users see no outage at all, i.e., better RTO.
- There is about half as much data loss (in a two node configuration, fewer if more nodes are used), i.e., better RPO, because only the updates in the replication stream on the failed node are lost. The updates in the replication stream on the remaining node(s) are unaffected, and will be replayed once the failed node is recovered.
- There are little to no testing costs/issues, and no failover faults. All systems in the configuration are known to be working at all times (which is also true for SZT).
- Better system capacity utilization as all nodes are performing productive work.

Active/Active – Route Anywhere

This architecture is the same as the active/active partitioned model described above except that the partitioning aspect is removed. Any transaction can be executed by any node (hence the name, "route anywhere"). This architecture has all the benefits of the active/active–partitioned model, but in addition, eliminates two of the issues with that model. It does not require partitioning (which may not be possible in all cases), and since transaction routing is unconstrained, workload can be evenly load-balanced across nodes.

There is always a price to pay, and in this case it is the possibility of data collisions. For some applications, data collisions may be practically impossible. For example, it is highly unlikely the same credit/debit/ATM card would be used simultaneously for multiple transactions. But if collisions are possible, they must be identified and dealt with immediately. Data replication should include functionality to automatically detect, report, and resolve data collisions. User exits may also be provided to enable more sophisticated processing of data collisions if necessary.

All business continuity architectures are not created equal. Figure 1 helps to visualize the differences between these various architectures with respect to the parameters of RPO and RTO.[5] As far as RPO is concerned, active/passive and SZT solutions are similar. For RTO, an SZT architecture trumps an active/passive architecture. But an active/active implementation beats active/passive and SZT architecture on all counts.

Business Continuity Architectures: Total Cost of Ownership

There is another way of looking at the various business continuity architectures, which provides an even more striking view of the differences between them and their relative benefits, and that is to look at the *total cost of ownership* (TCO). Active/passive configurations are cheaper and less complex to implement, however, when looked at through the lens of TCO, they have a (very) false economy.

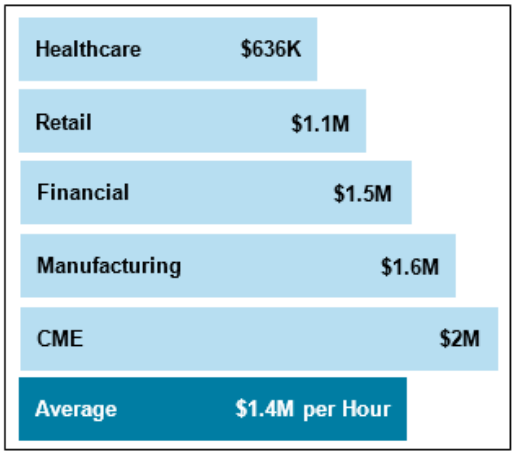

Figure 2 – Average costs per hour of downtime across various industries

Architecture	RTO	Outage Cost
Active/Passive[1]	~ 3 hours (if at all)	~ $4.5M
Active/Passive[2]	~ 10 minutes	~ $250K
Sizzling-Hot	~ 30 seconds[3]	~ $12.5K
Active/Active	~ 30 seconds	~ $6.25K[4]

[1] Worst case: with failover faults, management indecision, etc.
[2] Best case: with no failover faults, prompt management action, etc.
[3] Possibly slightly longer depending on network switching.
[4] Half of users see no outage at all (less than half if > 2 replicated nodes)

Figure 3 – Estimated Service Unavailability Costs for a Financial Application

[4] A data collision occurs when the same data record is updated simultaneously on two active systems, which after replication to the other system results in both copies of the data record being incorrect.
[5] Note that with respect to RTO and RPO, there is no difference between asynchronous active/active – partitioned and asynchronous active/active – route anywhere.
[6] Sources: Network Computing, the Meta Group, Contingency Planning Research.

First, to emphasize this point, let us put some dollar values on the cost of downtime (Figure 2).[6] As shown, these monetary costs are non-trivial, to say the least. Now, using the average cost per hour of downtime for a financial application of $1.5M/hour as an example, and using reported industry averages for typical periods of recovery time (RTO), we can estimate actual outage costs for the various business continuity architectures (Figure 3).

Obviously, basic active/passive architectures are very expensive when looked at in terms of TCO. They may be easier and cheaper to implement, but when outages do occur, they are likely to cost you much, much, more in the long run. Even in the best case scenario, with a well-tested system and a trouble-free failover, a basic active/passive configuration is still going to be about 20 times higher in outage costs when compared to an SZT configuration. For a worst case scenario (much more likely given the difficulties of testing and probability of failover faults as previously discussed), it is about 36 times more costly at about $4.5 million *per outage* (assuming the recovery *only* takes three hours, but it could take much longer).

The cost differences become even more apparent when viewed graphically (Figure 4). Given the marginal incremental cost and complexity, coupled with the significant decrease in potential outage costs, there is really no reason why anyone would run the risk and not move immediately from an active/passive to at least an SZT architecture.

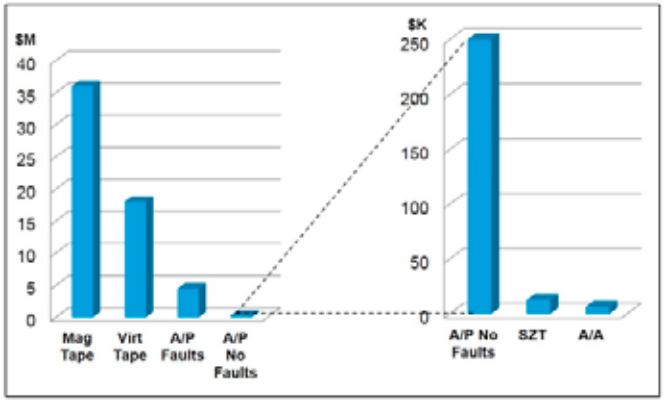

Figure 4 – Estimated Service Unavailability Costs for a Financial Application

As well as considering the cost of downtime, the cost of lost data needs to be considered. Again, using industry averages for the amount of data lost in the event of an outage (the amount of changed data which has not yet been safe-stored on a backup system), we can estimate the cost of lost data for the various business continuity architectures, across various industries (Figure 5).

Technology	RPO[1]	Retail[2]	CC/Debit[3]	EFT[4]	Stock Trade[5]
A/P + S/H[6]	~ 1 sec	$47.5K	$35.6K	$688K	$31.6M
A/A[6]	~ 0.5 sec	$23.8K	$17.8K	$344K	$15.8M

[1] Example assumes rate of 500 transactions per second
[2] Retail average transaction ~ $95 (US online) (Source: Monetate 2012)
[3] CC/Debit average transaction ~ $71 (UK) (Source: European Central Bank 2011)
[4] EFT average transaction ~ $1,376 (Source: Canadian Payments Association 2011)
[5] Stock trade average transaction ~ $63,284 (Source: London Stock Exchange 2012)
[6] Asynchronous replication

Figure 5 – Estimated Costs of Lost Data Across Various Industries

In this case, active/passive and SZT are the same since they both lose the same amount of data, but active/active is much better since it only loses half as much data. Again, this difference is more dramatically illustrated graphically (Figure 6).

But even if data loss (RPO) goals based on the average value of a transaction may appear acceptable, some data transactions are much more valuable than others and absolutely cannot be lost:

- Healthcare – lost dosage records can result in patient overdose on medication
- Manufacturing – car manufacturer can tolerate short production line outage, but cannot lose data regarding bolt torque settings, etc., in case of lawsuits from accidents
- Electronic Funds Transfer (EFT) – some transactions are worth millions, even if the average transaction is much lower
- Stock Trades – like EFT, some transactions are worth millions, and stock price is based on previous trades, so none can be lost

Therefore, RPO goals must be set based not on the value of an average transaction, but on the value of the most expensive/critical transaction. If the cost of losing the most valuable/critical data is very high, then an active/active configuration is the only solution, since it has the best RPO characteristics (least data loss).[7]

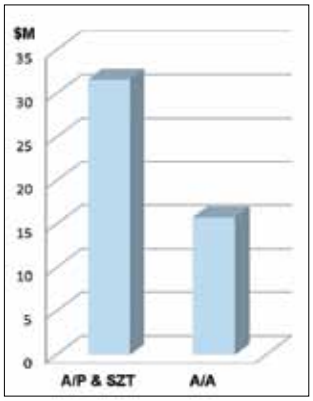

Figure 6 – Estimated Costs of Lost Data for Average Stock Trade Transaction

To summarize, overall TCO decreases by orders of magnitude more than the cost that the business continuity solution increases, as illustrated by Figure 7:

- The better the availability, the greater the complexity and implementation cost
- The better the availability, the lower the outage cost (by orders of magnitude)
- Net result, as implementation cost increases, overall TCO decreases

By this measure, the cost and complexity of an active/active solution is clearly more than outweighed by its superior overall TCO. It also illustrates how much better an SZT solution is in terms of TCO compared with a basic active/passive architecture.

Conclusion

To implement a business continuity plan, the IT architecture to be employed in order to maintain services in the event of an outage (planned or unplanned) must be selected. Many users select, and never get beyond, a basic active/passive architecture, but it has many issues, which can prevent a successful and timely failover. This model is reactive, risky, and provides a false sense of security. The likelihood of an extended outage is high; consequently, the likelihood of a very expensive outage is high. ***Active/passive architectures are simply not good enough for mission-critical applications.***

Though the more sophisticated business continuity solutions (SZT and active/active) are more complex and somewhat more expensive to implement, they are in fact far more cost-effective when looked at in terms of TCO. If you are running an active/

[7] If your application absolutely cannot tolerate any data loss, then contact Gravic, Inc. for more information about a new Shadowbase technology, synchronous replication, which will eliminate data loss entirely.

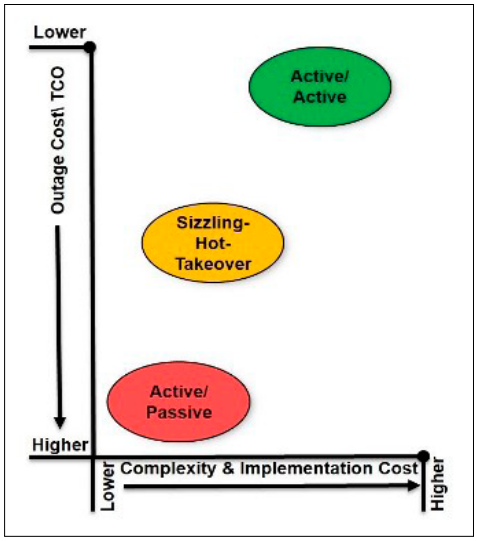

Figure 7 – TCO Versus Complexity/Cost of Implementation

passive architecture, it will probably only take one outage during peak processing hours to realize this fact the hard way. Because SZT is only marginally more complex than active/passive to implement, yet the benefits are significant, ***SZT should be the absolute minimum architecture chosen for applications which must remain available.***

SZT itself should only be seen as a stepping stone to a fully active/active architecture. Active/active cuts in half outage and data loss costs, and significantly improves the utilization of system capacity (i.e., there is no idle backup system). ***An active/active architecture provides the only acceptable solution for applications with high-value transactions where data loss must be minimized, and/or applications which must be continuously available.***

The solution is in your hands. The attention-grabbing outage headlines and long meetings with senior management explaining what happened need not be applicable to your company. With Shadowbase data replication, the solutions are available today to make extended outages a thing of the past. The potential outage costs described above are sobering; you do not want to validate them the hard way. If you are currently using an active/passive configuration, move as quickly as you can to an SZT architecture. If you are already running in SZT mode, congratulations, but also consider whether you should be taking the next step and moving to a fully active/active implementation. You do not want to make the right decision in hindsight, after it is too late to protect the availability of your mission-critical applications and data.

Sales and Support for Shadowbase data replication technology is available from select resellers in certain regions and from HP globally. Contact either Gravic or your HP account team for more information.

Keith B. Evans works on Shadowbase business development and product management for Shadowbase synchronous replication products, a significant and unique differentiating technology. Asynchronous data replication suffers from certain limitations such as data loss when outages occur, and data collisions in an active/active architecture. Synchronous replication removes these limitations, resulting in zero data loss when outages occur, and no possibility of data collisions in an active/active environment. Shadowbase synchronous replication can therefore be used for the most demanding of mission-critical applications, where the costs associated with any amount of downtime or lost data cannot be tolerated.