

Licensing distributed DB2 10.5 servers in a high availability (HA) environment

[Amyris Rada](#)

Senior Information Developer, DB2 for Linux, UNIX, and Windows
IBM

04 November 2013

[Roman Melnyk, B.](#)

DB2 Information Development
IBM

Are you trying to license your IBM® DB2® Version 10.5 for Linux®, UNIX®, and Windows® servers correctly in a high availability environment? Do you need help interpreting the announcement letters and licenses? This article explains it all in plain English for the DB2 10.5 release that became generally available on June 14, 2013.

Introduction

Customers choose DB2 because of its incredible time to value, its ability to scale and integrate across disparate environments, its robustness, and minimized down time (both planned and unplanned). In this article, we focus on the high availability (HA) aspects of DB2, especially from the licensing point of view.

We receive a lot of questions about licensing DB2 in an HA environment. One major source of confusion is the fact that vendors differ greatly in how they price their database products in HA environments.

Another source of confusion is terminology. For example, the IT industry sometimes refers to HA environments as *clusters*. We don't like using this term by itself anymore, because it has become somewhat overloaded. Clusters can refer to *clustering for scalability* (such as the partitioned database environments that are available when you use DB2 Advanced Enterprise Edition), *clustering for availability* (available through software such as Tivoli System Automation for Multiplatforms, which is deeply integrated with DB2), or both (as is the case with a DB2 pureScale cluster or the IBM Smart Analytics System). In this article, when using the term *cluster*, we mean clustering for HA (unless otherwise noted).

For clarity, you should refer to *high availability clusters* or *scalability clusters* when discussing clusters with your clients or colleagues. Of course, some solutions address both scalability and high availability in relation to clusters, so ensure that you draw the right distinctions.

The terms that are used to describe the server that acts as the failover server after an outage are sometimes confusing too. For example, *standby* or *secondary* server. If you've been around long enough, you've likely encountered a variety of terms for the function that a standby server performs. Terms such as *idle*, *active*, *cold*, *warm*, *hot*, and *passive* have all been used in availability discussions.

For the most part, IBM Software Group (IBM SWG) literature uses the temperature taxonomy (*cold*, *warm*, and *hot*) to describe standby servers, and DB2 has conformed to the IBM SWG taxonomy since DB2 9.5.

We've updated this article for the DB2 10.5 release (generally available on June 14, 2013) to help you sort out DB2 high availability licensing rules and put you *in-the-know*. This article also covers the DB2 pureScale technology, which has been greatly enhanced for the DB2 10.5 release, with online fix pack updates, backup and restore between DB2 pureScale instances and regular DB2 instances, and HADR. In DB2 10.5, DB2 pureScale is included with DB2 Developer Edition, DB2 Advanced Workgroup Edition, and DB2 Advanced Enterprise Edition.

[Figure 1](#) summarizes the DB2 10.5 high availability (HA) taxonomy that is based data "temperature" and gives examples of the types of configuration that fall into each temperature category.

Figure 1. DB2 10.5 HA taxonomy with multiple temperatures

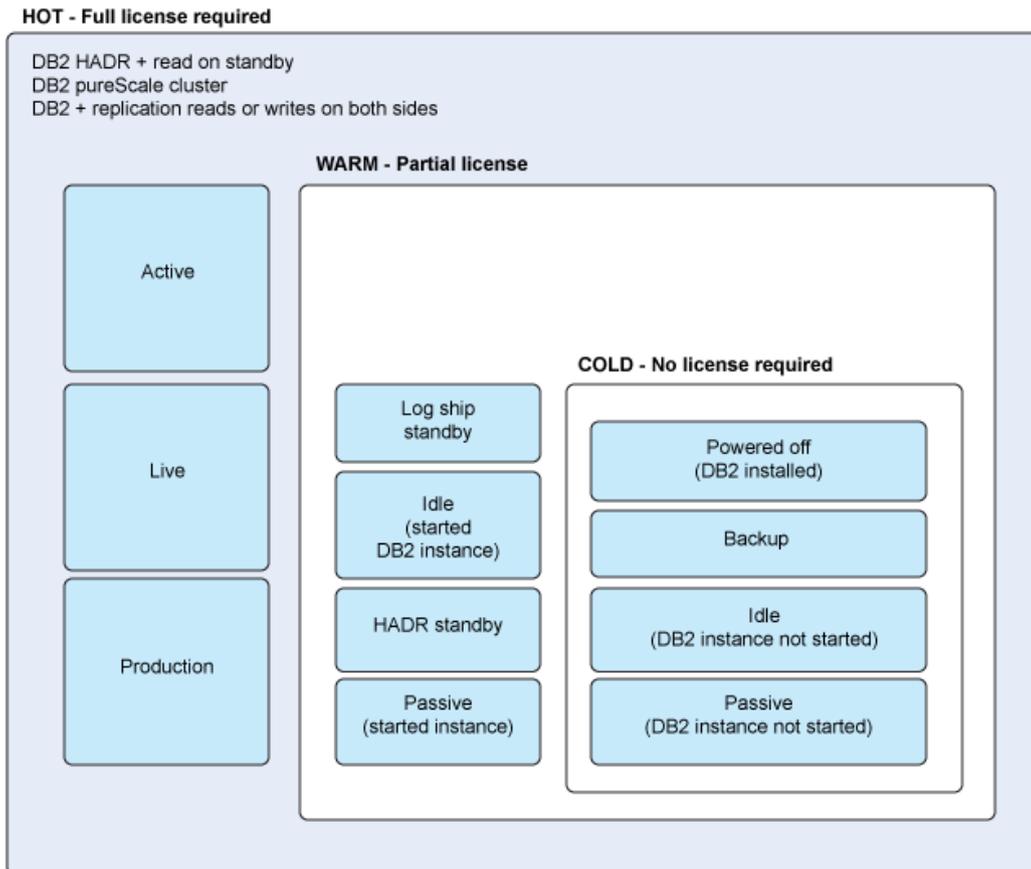


Table 1 shows the most common terms that are used to describe HA environments being mapped to the DB2 taxonomy and licensing terms.

Table 1. Industry high availability terms mapped to DB2 licensing terms

Hot	Warm	Cold
Database software is installed on another server intended for failover purposes.	Database software is installed on another server intended for failover purposes.	Database software is installed on another server intended for failover purposes.
At the same time that this server is maintaining its partner high availability scenario, it is servicing other applications as a primary data server. There is end-user access to this standby database even when no failure has occurred.	The DB2 instance is started, and it might be receiving updates from the primary database for high availability purposes only, such as applying logs or doing recovery work. There is no end-user access to this standby database.	The DB2 instance is not started during normal operations, and it will be started only when a failover occurs. The database can also be started briefly for maintenance operations (such as applying a fix pack or performing a backup) following which it must be stopped.
Typically used in twin-failover HADR, HADR with read on standby, DB2 pureScale, HADR in a DB2 pureScale environment, or replication scenarios.	Typically used in a "vanilla" HADR (no reads on standby), Q-replication, or log shipping scenario.	Typically used in clustering scenarios where DB2 pureScale, HADR, or log shipping is <i>not</i> deployed and the DB2 instance is not started, such as a PowerHA for AIX cluster (formerly known as HACMP).

How you license a DB2 server in an HA environment depends on your answers to the following key questions:

What DB2 edition do you have installed?

Is it DB2 Express-C, DB2 Express, DB2 Workgroup, DB2 Enterprise, or DB2 Advanced Enterprise? If it's the free DB2 Express-C, you should be aware that you are not licensed to use DB2 Express-C in any kind of HA configuration. If you need HA, you need at least a DB2 Express license. All editions of DB2 10.5 include full support for hot, warm, or cold standby server configurations (more on that in a bit). This is especially good news for DB2 Express customers using HADR, who had to separately license the DB2 High Availability Feature for Express Edition with DB2 9.7 (if they licensed DB2 Express under either the PVU or AU pricing metrics) to set the HADR failover.

How did you license your primary DB2 server, which you want to ensure is highly available?

The DB2 edition and licensing metric chosen for the standby server must match the primary server. For example, if you license your DB2 Express server by using the Limited Use Virtual Server (LUVS) or Fixed Term Server (FTL) licensing options, you'd have to buy an additional LUVS or FTL license for the standby server, whether it's hot or warm. If the standby is cold, an additional LUVS or FTL license is not required. If you license your DB2 Express server by using the Authorized User Single Install (AUSI) model, you'd license the standby server in a warm state for 5 AUSIs, and you wouldn't need any licenses for a cold standby machine. If your DB2 Express server is licensed by using the [Processor Value Unit \(PVU\)](#) model, you'd license a warm standby server for 100 PVUs (no matter what processor architecture the server is using), and you wouldn't need a license for a cold standby machine either.

How is the standby machine being used when a failure *has not* occurred?

Is it being used as a production server for DB2 transaction and query work? Is the DB2 instance on this server running? Perhaps the instance is performing work, but only to help prime a recovery database in the event of a failure, such as in an HADR scenario. Are you managing a DB2 pureScale cluster? What the standby server is doing when *everything* is running just fine has pretty much *everything* to do with how your DB2 server needs to be licensed.

If you're looking for an overview of all the distributed DB2 10.5 servers and how to license them, see ["Which distributed edition of DB2 10.5 is right for you?"](#). For comparisons of the features, functions, and benefits that are available with the different DB2 10.5 server editions, see ["Compare the distributed DB2 10.5 data servers"](#).

There are a number of licensing enhancements that have significantly lowered the cost of HA for your DB2 servers through the years. Starting with DB2 10.5, you no longer have to license selected tools on a DB2 server that is acting as a warm or cold standby server. This applies to the following tools:

- InfoSphere Optim High Performance Unload for Linux, UNIX, and Windows
- InfoSphere Optim Performance Manager Extended Insight for Linux, UNIX, and Windows
- InfoSphere Optim Performance Manager Extended Edition for Linux, UNIX, and Windows
- InfoSphere Optim Performance Manager Enterprise Edition for Linux, UNIX, and Windows
- InfoSphere Optim Performance Manager Workgroup Edition for Linux, UNIX, and Windows
- InfoSphere Optim Performance Manager Content Manager Edition for Linux, UNIX, and Windows
- InfoSphere Optim Configuration Manager for Linux, UNIX, and Windows

- InfoSphere Optim Query Workload Tuner for Linux, UNIX, and Windows

In this article, we refer to *licensing a server*; however, all DB2 editions support subcapacity licensing, in which you license only the capacity that the DB2 server is using. If we use a phrase such as *license the PVU rating of the server*, this means the PVU rating of a VMWare session or an LPAR if you're using these virtualization technologies. There are some [prerequisites](#) and special rules for both PVU and non-PVU-based subcapacity licensing, so ensure that you know all the details before deploying DB2 in this kind of environment.

DB2 Licensing for HA environments

As of DB2 10.5, there are five different licensing metrics:

- [Limited Use Virtual Server \(LUVS\)](#) is available only in DB2 Express.
- [Fixed Term License \(FTL\)](#) is available only in DB2 Express.
- [Limited Use Socket \(LU Socket\)](#) is available only in DB2 Workgroup.
- [Terabyte License \(TB\)](#) is available only in DB2 Advanced Workgroup and DB2 Advanced Enterprise.
- [Processor Value Unit \(PVU\)](#) is available in all editions.
- [Authorized User Single Install \(AUSI\)](#) is available in all editions.

The following information details the HA-licensing considerations that you should be aware of for each licensing option.

LUVS license

The LUVS license is *only* available for DB2 Express servers. To license a DB2 Express server by using a LUVS license, you simply buy a license for *each physical or virtual DB2 server* in the cluster that is *hot or warm*. A *cold DB2 standby* requires no license. Identifying the activity level of a DB2 Express server with a LUVS license is not necessary when it comes to HA licensing. There are no user minimums, and you don't have to figure out the PVU rating of the underlying server or anything else: simple! For an example scenario, see [Figure 2](#).

Fixed Term License (FTL)

A DB2 Express FTL gives you access to the DB2 Express software for a one year. It is a term license, not a perpetual license, which is the case with the other DB2 licensing options. To license a DB2 Express server with an FTL, you simply buy a license for *each hot or warm physical server* in the cluster, just like the DB2 Express LUVS licenses. Cold standby servers require no license. As with LUVS licenses, FTLs don't have user minimums and you don't have to figure out the PVU rating.

LU Socket license

The LU Socket license was introduced in DB2 9.7 just for DB2 Workgroup servers. To license a hot DB2 Workgroup standby server by using an LU Socket license, you simply license the number of sockets on the *physical or virtual server*, the same as you would for the primary server, because all servers are fully utilized for regular operations. To license a warm DB2 Workgroup standby on a *physical or virtual server* by using an LU Socket license, you need only one LU Socket license.

For example, let's assume that you have a cluster consisting of two unpartitioned, 4-way dual core IBM POWER7 servers, and the standby server is identical to the primary server.

- For a primary DB2 server, you would buy four DB2 Workgroup LU Socket licenses.
- For a hot DB2 standby server, you would buy four DB2 Workgroup LU Socket licenses. Incidentally, this server would be rated at 960 PVUs, and you would still buy only four DB2 Workgroup LU Socket licenses. You would buy four DB2 Workgroup LU Socket licenses for the primary server.
- For a warm DB2 standby server, you simply buy one additional LU Socket license, no matter what.
- For a cold DB2 standby server, no license is required.

LU Socket licenses should be used with DB2 Workgroup servers because of the additional CPU capacity that this option offers to your environment.

Terabyte License (TB)

The Terabyte (TB) licensing option is available for DB2 Advanced Workgroup and DB2 Advanced Enterprise. These editions include a script tool that helps you to calculate the required number of TB licenses for each database. For a *hot standby server*, including reads on standby servers, the number of TB licenses required is the same as the number of licenses that are required for the primary server. For a *warm standby server*, one TB license is required for each database on the server. Cold standby servers require no license. HADR capabilities are not available with a TB license. However, DB2 Advanced editions include the homogeneous Q replication capability that you can use to set up a standby environment. If the servers in [Figure 4](#) use homogeneous Q replication instead of HADR, you could use licenses with the TB metric. Data server A and Data server B require the same number of TB licenses.

If the servers in [Figure 6](#) use homogeneous Q replication instead of HADR and the database is 7.5 TB, Data server 2 requires eight TB licenses, and Data server 1 requires only one TB license. If Data server 2 had a second database of 5 TB, Data server 2 would require 13 TB licenses, and Data server 1 would require two TB licenses.

Processor Value Unit (PVU) license

Any DB2 physical or virtual server can be licensed by using the PVU licensing model. For a *hot standby server*, including reads on standby servers, the entire PVU rating of the server must be licensed. Of course, you would use this same approach to license your DB2 server even if it weren't part of an HA cluster, because it is servicing production work anyway, so there shouldn't be any surprises here.

If the servers in [Figure 3](#) used 64 processor cores of a POWER6 server, and assuming that each machine were running DB2 Enterprise, you would have to purchase a total of 15360 PVUs for this solution: 7680 PVUs for Data server 1 and 7680 PVUs for Data server 2.

For a *warm standby server*, a license of 100 PVUs is required, regardless of the server PVU rating. [Figure 6](#) shows Data server 2 with a license of 7680 PVUs and Data server 1 as a

warm standby server with a license of 100 PVUs. A cold standby server does not require a license.

Authorized User Single Install (AUSI) license

An *authorized user* (AU) is a single individual (in some cases, it can be an application or appliance, as long as it doesn't act on behalf of other users) with a specific identity who resides inside or outside of your company. If the AUSI metric is being used, the following licenses are required:

- For a *primary DB2 server*, an AUSI license with the total number of AUs who will access the server is required. You need to account for everyone who accesses the server, and meet the minimum limits per edition. The DB2 Express and DB2 Workgroup editions require a minimum of five AUSI licenses. DB2 Enterprise, DB2 Advanced Workgroup, and DB2 Advanced Enterprise require a minimum of 25 AUSI licenses for every 100 PVUs on the server.
- For a *warm standby server*, a DB2 Express or DB2 Workgroup AUSI license for five users is required. For all other editions, an AUSI license for 25 users is required. If you use the AUSI metric in [Figure 6](#), a license for 25 users is required, because the minimum number of PUVs in a warm standby is 100.
- For a cold DB2 standby, no license is required.

There have been a lot of changes to flatten the total cost of ownership (TCO) that is associated with DB2 HA clusters, both from a licensing and administrative perspective. That's pretty refreshing, considering the kind of economy we are in. [Table 2](#) summarizes these changes.

Table 2. Summary of DB2 licensing terms for HA environments.

DB2 Edition	Cold	Warm	Hot
DB2 Express-C	<ul style="list-style-type: none"> • Not available 	<ul style="list-style-type: none"> • Not available 	<ul style="list-style-type: none"> • Not available
DB2 Express Edition	<ul style="list-style-type: none"> • Standby server does not require a license. 	<ul style="list-style-type: none"> • If the primary server has a PVU license, the standby server must be licensed for 100 PVUs. • If the primary server has an AUSI license, the standby server must be licensed for five authorized users. • If the primary server has a LUVS license, the standby server must be licensed with one LUVS license. 	<ul style="list-style-type: none"> • Standby server must be licensed with the same DB2 edition and licensing metric as the primary server, including the Advanced Recovery Feature.
DB2 Workgroup Edition	<ul style="list-style-type: none"> • Standby server does not require a license. 	<ul style="list-style-type: none"> • If the primary server has a PVU license, the standby server must be licensed for 100 PVUs. • If the primary server has a socket license, the standby server must be licensed for one socket. • If the primary server has an AUSI license, the standby server must be licensed for five authorized users. 	<ul style="list-style-type: none"> • Standby server must be licensed with the same DB2 edition and licensing metric as the primary server, including the Advanced Recovery Feature.
DB2 Advanced Workgroup Edition	<ul style="list-style-type: none"> • Standby server does not require a license. 	<ul style="list-style-type: none"> • If the primary server has a PVU license, the standby 	<ul style="list-style-type: none"> • Standby server must be licensed with the

		<p>server must be licensed for 100 PVUs.</p> <ul style="list-style-type: none"> • If the primary server has an AUSI license, the standby server must be licensed for 25 authorized users. • If the primary server has a TB license, a standby server that uses alternate technologies must be licensed for a separate 1-TB entitlement per database. 	<p>same DB2 edition and licensing metric as the primary server, including the Advanced Recovery Feature.</p>
DB2 Enterprise Server Edition	<ul style="list-style-type: none"> • Standby server does not require a license. 	<ul style="list-style-type: none"> • If the primary server has a PVU license, the standby server must be licensed for 100 PVUs. • If the primary server has an AUSI license, the standby server must be licensed for 25 authorized users. 	<ul style="list-style-type: none"> • Standby server must be licensed with the same DB2 edition and licensing metric as the primary server, including the Advanced Recovery Feature.
DB2 Advanced Enterprise Server Edition	<ul style="list-style-type: none"> • Standby server does not require a license. 	<ul style="list-style-type: none"> • If the primary server has a PVU license, the standby server must be licensed for 100 PVUs. • If the primary server has an AUSI license, the standby server must be licensed for 25 authorized users. • If the primary server has a TB license, a standby server that uses alternate technologies must be licensed for a separate 1-TB entitlement per database. 	<ul style="list-style-type: none"> • Standby server must be licensed with the same DB2 edition and licensing metric as the primary server, including the Advanced Recovery Feature.

Hot standby

A *hot standby* configuration is one in which all physical or virtual servers have operational DB2 databases that service user transactions and queries. This configuration is called a *hot/hot* cluster (or an *active/active* configuration, because all of the servers in the cluster are performing some level of business production work at all times). If one of the servers in the HA cluster fails, clustering software redistributes the failed server's workload to one or more surviving servers in the HA cluster.

If a server in a two-node cluster fails, the workload on the hot standby server doubles, because it now has to handle its original workload *plus* the failed server's workload. This scenario is something to consider when you are setting up any HA environment. If you are a database administrator (DBA) who's laying it all on the line for a tight SLA, you need to ensure that you properly vet your HA topology and any proposed HA solution.

To summarize, a hot standby server is any machine or virtual server being used to service user transactions and queries during normal operational periods, while at the same time acting as a standby for another server that is also servicing user transactions during normal operational periods. When a server failure occurs, the hot standby server takes on the load of the failed server. Whether or not the standby server continues to do the work that it was handling during normal

operational periods is immaterial. The fact that the standby server is being used to service user transactions or user queries during normal operations means that it must be fully licensed.

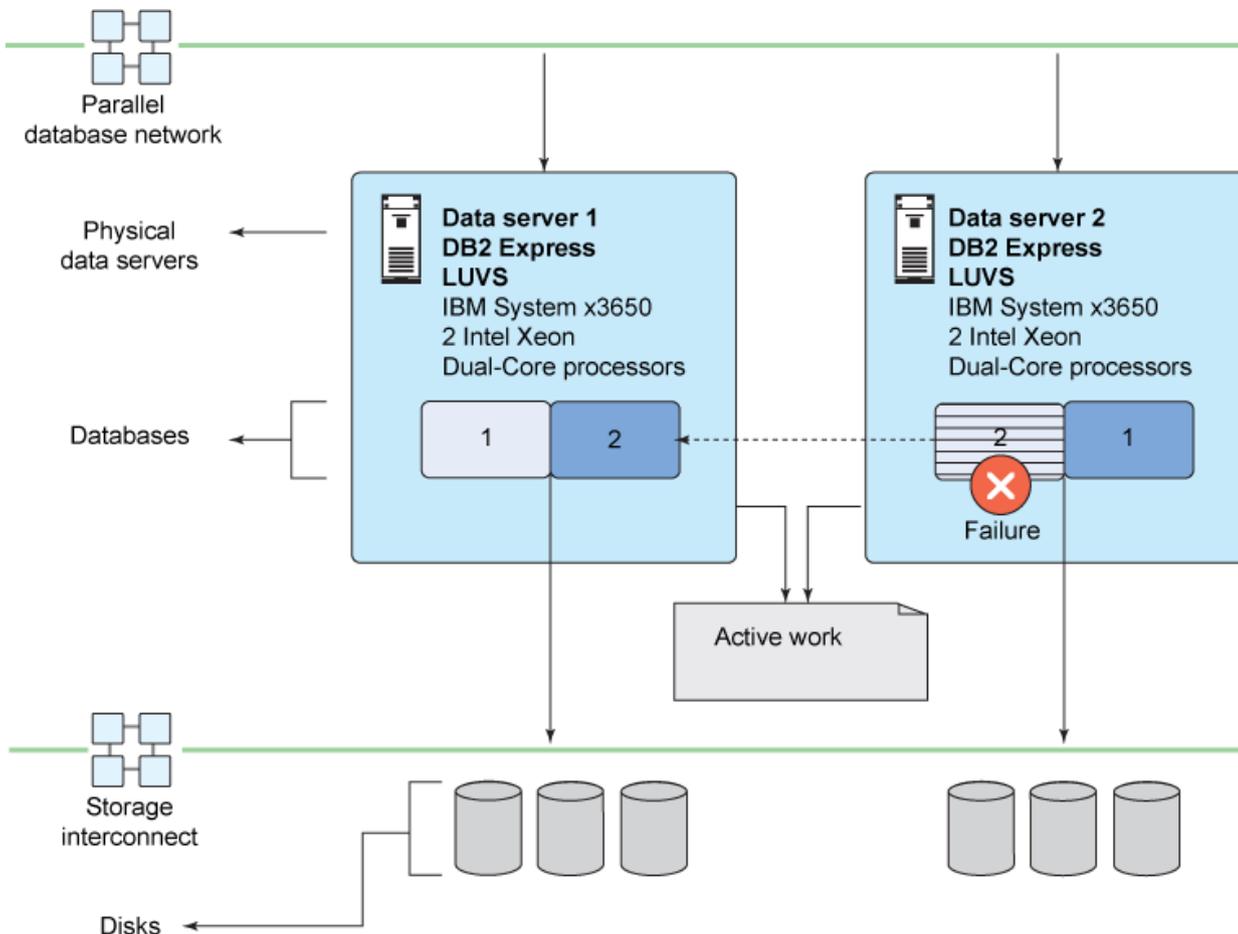
In the remaining sections, we discuss the following hot standby scenarios in detail:

- [Two-node hot standby HA cluster](#)
- [Hot/hot HADR cluster](#)
- [Hot/hot HADR cluster with "read on standby"](#)
- [DB2 pureScale environment with three members](#)

Two-node hot standby HA cluster

A scenario illustrating a two-node hot standby HA cluster is shown in [Figure 2](#), where Data server 1 is a hot standby for Data server 2, and Data server 2 is a hot standby for Data server 1. After a failure, Data server 2's workload fails over to Data server 1, and vice versa. Data server 1 has a [DB2 Express Limited Use Virtual Server \(LUVS\)](#) license. Therefore, Data server 2 must also have a DB2 Express LUVS license.

Figure 2. Two-node hot standby HA cluster scenario



In this example, Data server 1 and Data server 2 are a pair of unpartitioned servers that are both being used for transaction and query workload processing during periods of normal operation. The

solid boxes represent the database on each machine where the workloads are executing before a failure, and the cross-hatched boxes are the standby databases where the work will occur in the event of a failure on the failover partner machine.

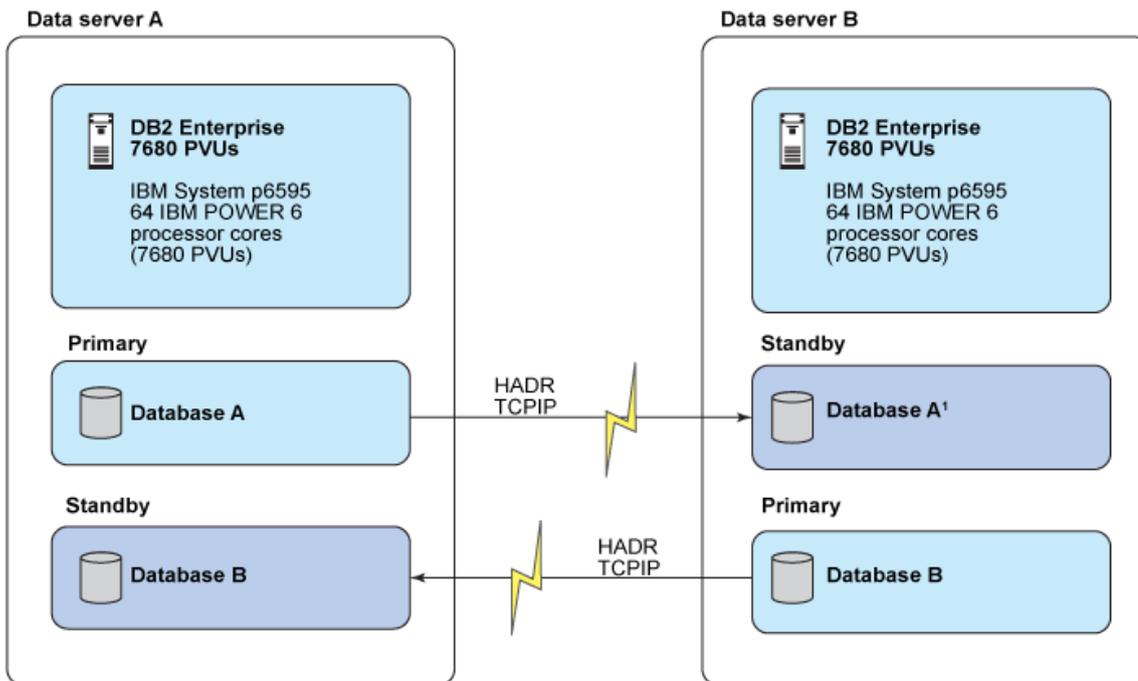
In this sample scenario, Data server 2 fails, and its workload is transferred to its failover partner, Data server 1. Data server 1 is a hot standby server to Data server 2 (and Data server 2 is a hot standby server to Data server 1). This type of configuration is often referred to as a *high-availability clustering pair*, *twin failover pair*, *hot/hot pair*, or *active/active pair*, and is common in today's IT landscape.

There are many ways in which to implement a hot/hot high availability cluster in DB2, and depending on what you need from the solution, you can use PowerHA for AIX, HADR in conjunction with a database on each server failing over to the other, HADR (with reads on standby activated), HADR in a DB2 pureScale environment, Q-replication, hot standby for reporting that uses snapshot technology or disk image copies, or DB2 pureScale, which is the pinnacle of scalability and availability combined.

Hot/hot HADR cluster

An HADR cluster can function as a hot/hot cluster in multiple ways. Consider the environment that is shown in [Figure 3](#). Both servers are licensed for [DB2 Enterprise Processor Value Unit License \(PVU\)](#).

Figure 3. A hot/hot HADR cluster scenario



[Figure 3](#) shows that an unpartitioned data server A runs a production database called Database A, and a standby database in an HADR cluster called Database B¹. At the same time, unpartitioned data server B runs a production database called Database B, and a standby database in the same

HADR cluster called Database A¹. Under normal operating conditions, both Server A and Server B are busy servicing production work on Database A and Database B, respectively (a hot/hot configuration). Both servers must be *fully licensed with the same metric* of 7680 PVUs.

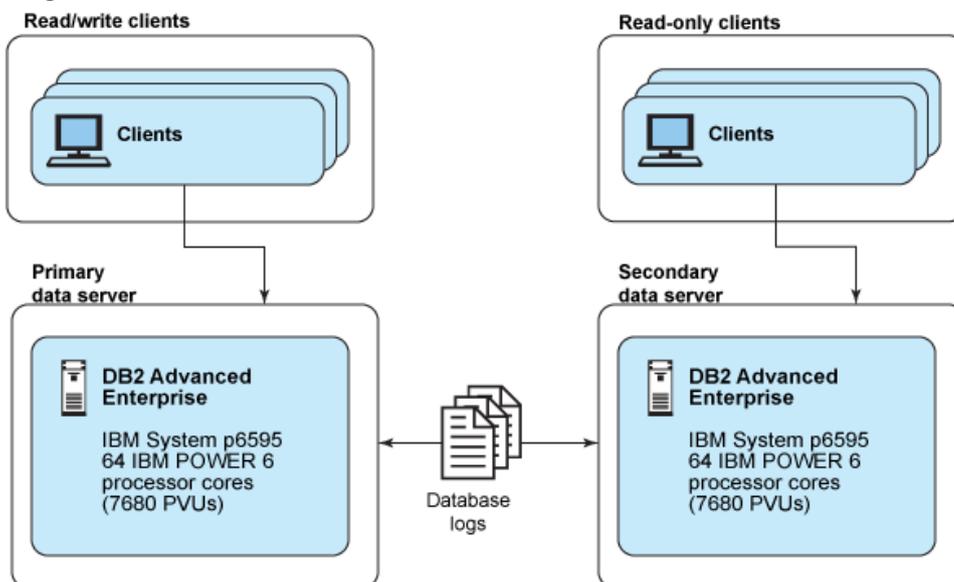
In this case, you would need to fully license both servers even if there were no standby databases in the environment. This example highlights an important point: If a physical or virtual server running an edition of DB2 is already fully licensed, no additional licenses are required to run a standby database of the same edition on that physical or virtual server. This rule applies regardless of licensing metric or whether the standby database is warm or hot.

- If all databases shown in [Figure 3](#) were DB2 Express licensed under the LUVS metric, you would need only two LUVS licenses in total. However, if you were running the primary and standby databases in separate LPARs, you would need four LUVS licenses.
- If you had two DB2 Workgroup servers accessed by 100 users, you would need to purchase a DB2 Workgroup AUSI license for 200 users: 2 servers x 100 AUs per server. Even if fewer than 100 users were ever connected to either server at any one time, all 100 users would still have to be licensed *for each* server.
- If you were using DB2 Express or DB2 Workgroup and only three users were accessing the servers, you would need a total of 10 DB2 Express or DB2 Workgroup AUSI licenses: (2 servers x 5 minimum AUs) to satisfy the minimum AUSI requirements that are associated with these DB2 editions.

Hot/hot HADR cluster with "read on standby"

[Figure 4](#) shows an example of an HADR environment that uses the read on standby capability, which has been available since DB2 9.7 Fix Pack 1. Both servers are licensed for [DB2 Advanced Enterprise Authorized User License Single Install \(AUSI\)](#).

Figure 4. A hot/hot HADR cluster with read on standby scenario



Clients can read from or write to the primary server (making it hot); however, because clients are also reading from the secondary server during normal operations, it is also hot, and therefore *both*

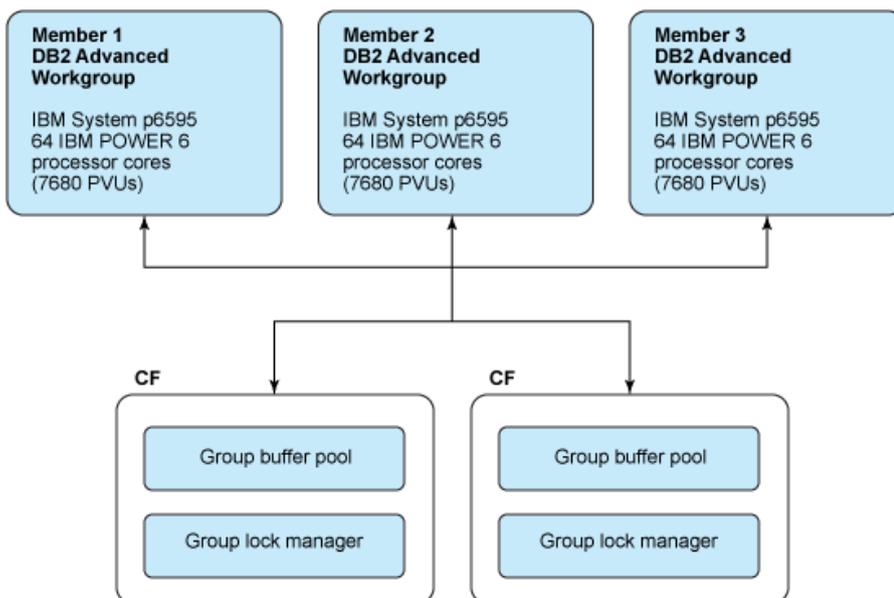
servers must be fully licensed. If you read data from a server for business purposes, then it's a hot server. In this scenario, both servers would have to be licensed for at least 1925 users, because a minimum of 25 users per 100 PVUs is required.

If homogeneous Q replication is used instead of HADR and the primary has a [DB2 Advanced Enterprise 10-TB license](#), the secondary must also have a DB2 Advanced Enterprise 10TB license.

DB2 pureScale environment with three members

Figure 5 shows a DB2 pureScale HA and scaling cluster with three members and two cluster caching facility (CF) servers. The members have a [DB2 Advanced Workgroup PVU license](#). DB2 pureScale is included in DB2 Advanced Workgroup PVU and AUSI licenses. You cannot use a TB license, because DB2 pureScale is not available with that license.

Figure 5. A DB2 pureScale cluster scenario



In this scenario, because transactions are being seamlessly load balanced across all three members, the members are all hot, and each member must have a license for 7680 PVUs. CF servers are typically fully duplexed and do not require any license.

Warm standby

A *warm standby* configuration is one in which at least one physical or virtual server in the HA cluster with DB2 installed and running is not servicing user transactions or query workloads during periods of normal operation. It is warm in the sense that the server is not performing "useful" work for end users. Work that is classified as "not useful" from an end user point of view includes administrative actions that assist in failover scenarios, such as having a database in rollforward pending state to support log shipping, supporting transaction-level log shipping for an HADR environment, performing a backup, and so on. If one of the servers in the HA cluster fails, the workload is redistributed to the warm standby server.

One common misconception is that a warm standby server is a waste of resource when it is solely dedicated to recovery operations. In fact, you can leverage a warm standby server for many uses well beyond the standby role. For example, you could create a separate DB2 instance on the warm standby server (with possible licensing implications) and use it as a test server, or perhaps offload other workloads and functions to it. In the event of a failure, you could scale back these workloads (or resources allocated to a virtualized partition where they run) and allow the warm standby server to use all of its resources to handle the load from the failed server.

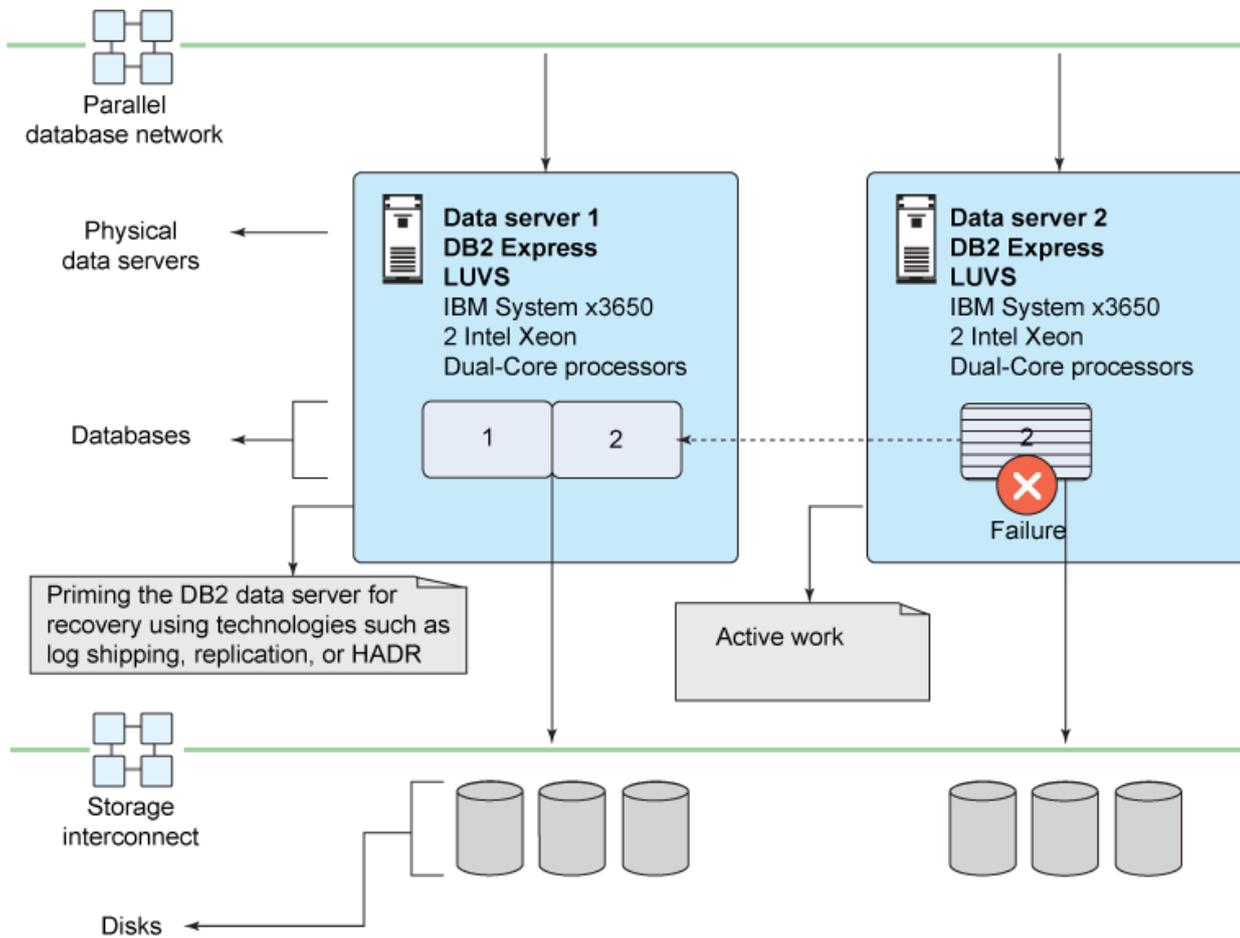
You might choose to use the standby database as a somewhat current read-only version of the production database, thereby spreading out cost over more resources. "Read on standby" has been available since DB2 9.7 Fix Pack 1. Be aware that many vendors limit this model to "either/or", meaning that while you are reading from the database, you can't roll forward through the logs to keep it current. And even if it isn't "either/or", there might be other compromises, such as the speed at which redo processing can run.

Hot/warm HADR cluster

A scenario illustrating a two-node warm standby (sometimes referred to as *active/idle*) HA cluster is shown in [Figure 6](#). This is a "vanilla" HADR configuration (the read on standby capability is not being used) that was created for a production OLTP environment. In this example, during periods of normal operation, Data server 2 is being used for transaction and query workloads (shown as Active Work in the figure), and Data server 1 is acting as a warm standby. Data server 2 fails, and its DB2 workload is passed to Data server 1. Data server 2 is licensed for DB2 Enterprise 7680 PVUs, and Data server 1 is licensed for DB2 Enterprise 100 PVUs.

If work of any kind were being performed on Data server 1 before the failure, it would be scaled back to handle the additional workload from Data server 2.

Figure 6. A two-node hot/warm HADR cluster scenario



If Data server 2 had four DB2 Workgroup LU Socket licenses, Data server 1 would require only one additional LU Socket license for a warm standby. The number of LU Socket licenses is independent of the 7680 PVUs rating.

If you had wanted to use the AUSI metric in this scenario because you have only 100 users, you would still need a license for 1950 users, because a minimum of 25 users per 100 PVUs is required ($7680 / 100 = 77 \times 25 = 1925$ AUs) for the primary server in addition to 25 users for the warm standby server.

Cold standby

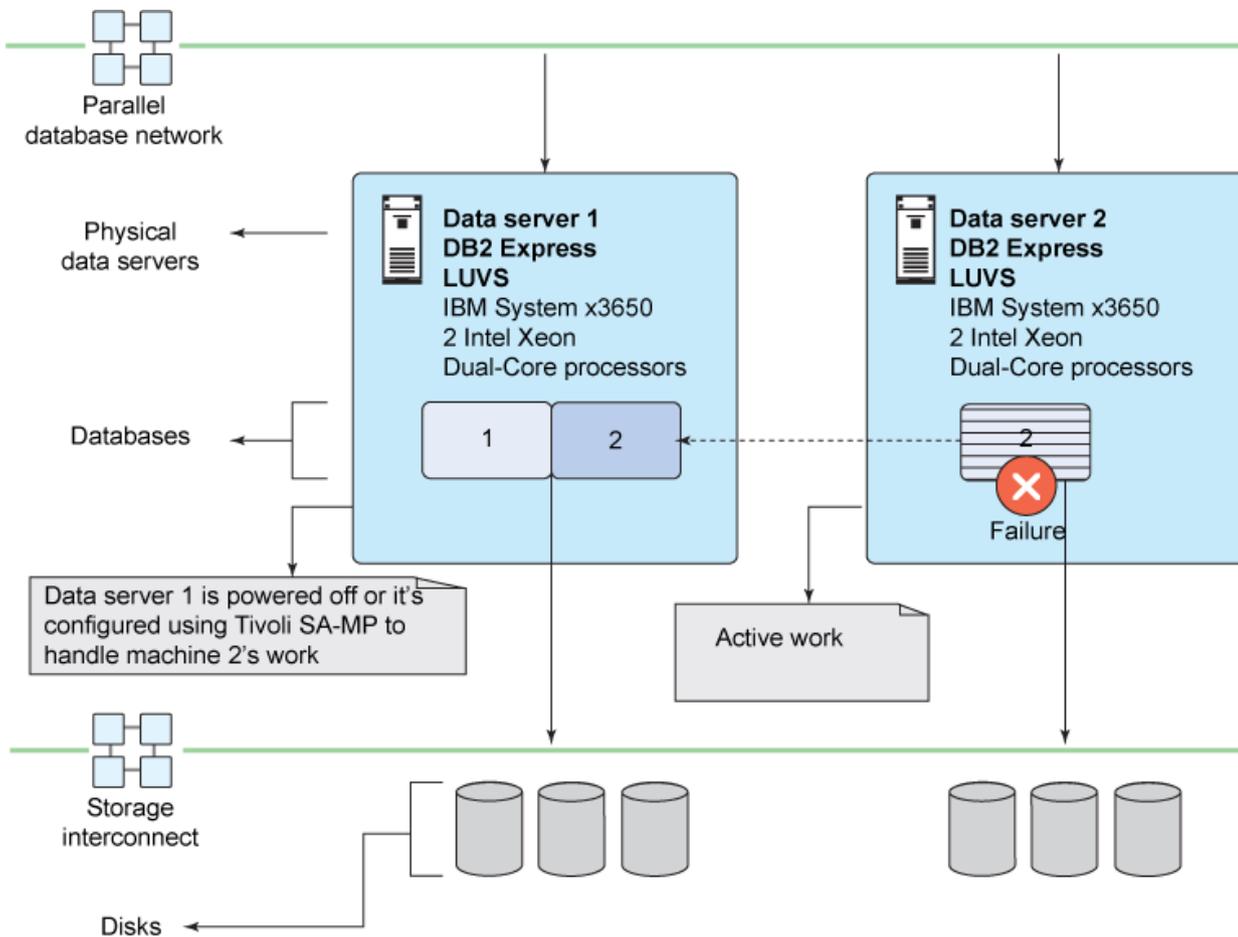
A *cold standby* configuration is one for which the following conditions are true:

- At least one physical or virtual server in the cluster does not have a DB2 database that is servicing user transactions or query workloads during periods of normal operation.
- At least one physical or virtual server in the cluster is not performing any administrative actions that assist in failover scenarios, such as having a database in rollforward pending state to support HADR. In fact, a cold standby server cannot even be powered on except for brief periods to conduct maintenance, such as performing a backup operation or applying a fix pack.

Hot/cold HADR cluster

A scenario illustrating a two-node cold standby HA cluster is shown in [Figure 7](#).

Figure 7. A two-node hot/cold HADR cluster scenario



In this example, during periods of normal operation, Data server 2 is being used for transaction and query workloads (shown as Active Work in the figure), but Data server 1 doesn't have a running DB2 instance. When Data server 2 fails, Data server 1 is booted up, recovered to a certain point in time from a backup image, and made available for user transactions. It could also be the case that Data server 1 is configured to be part of a Tivoli SA MP cluster (among other HA clustering options, including PowerHA for AIX), but that the database instance isn't started. As you can imagine, a cold standby configuration isn't much of an availability solution, because down time is typically going to be a lot longer than with a hot or warm standby configuration. Nevertheless, a cold standby configuration might meet the needs of some of your applications.

Mixed standby environments

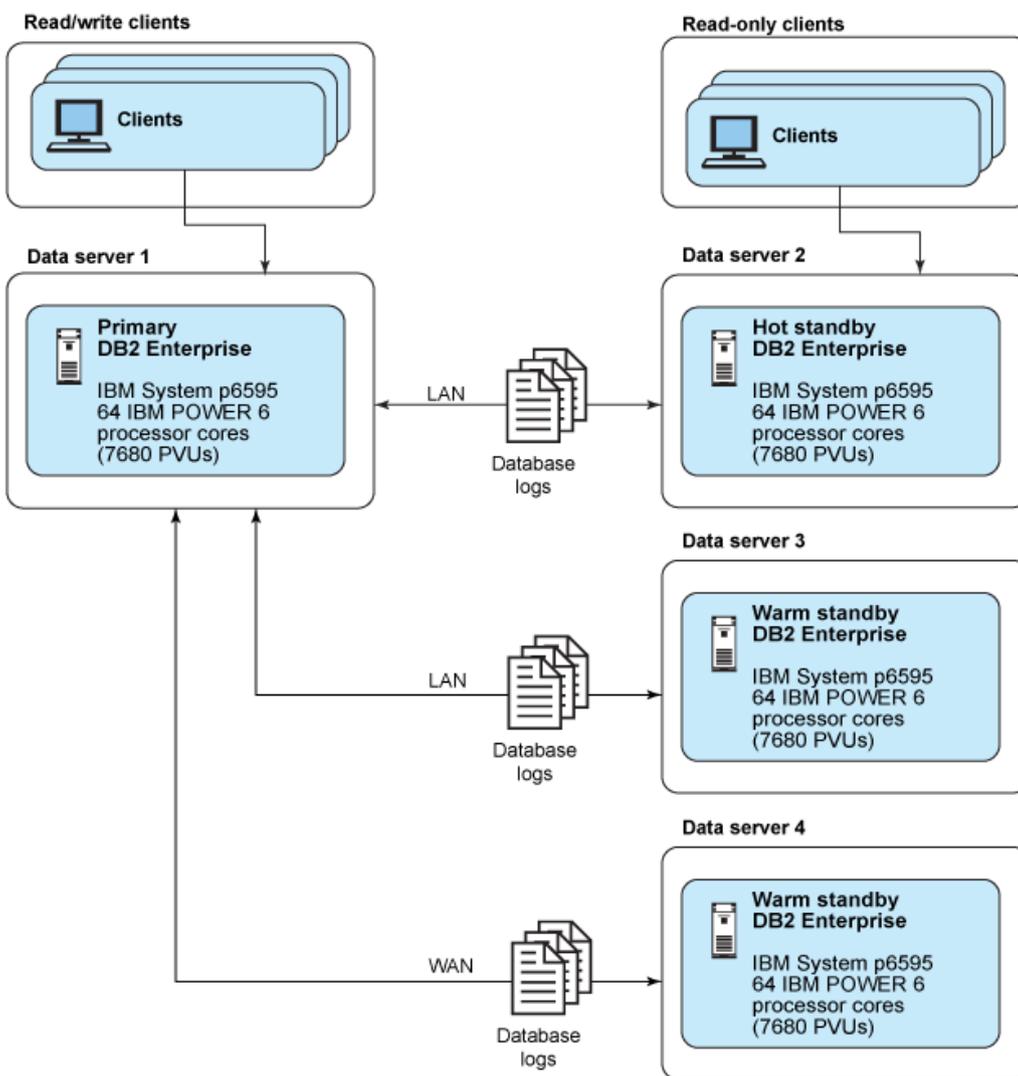
So far, we've only considered HA scenarios where the standbys were all hot, all warm, or all cold. That will always be the case if you have only a single standby server in your cluster or if you are using DB2 pureScale, which supports only hot standbys. However, it's also possible to have

multiple standbys in an HA environment that are not the same "temperature", for redundancy and disaster recovery purposes.

HA cluster with hot and warm standby

The following scenario requires redundant HA for the primary data center, as well as disaster recovery at a second remote data center. Furthermore, it needs the ability to process read workloads without adding any extra load to the primary server during normal operations. One architecture that minimally meets those requirements is shown in [Figure 8](#).

Figure 8. HA cluster with hot and warm standby scenario



In this example, the primary database on Data server 1 uses log shipping or Q-replication to replicate data to two local standby servers, Data server 2 and Machine 3, as well as a third remote standby server, Machine 4. If Data server 1 fails, Data server 2 takes over the workload. If Data server 2 also fails, Machine 3 takes over. During normal operations, Data server 2 is also used to process read workloads and must therefore be fully licensed. Machines 3 and 4 don't process any

end-user workloads during normal operations, but must be fully functional to be able to receive and apply the logs from the primary. For this reason, they must both be licensed as warm servers.

The license considerations for each standby in a mixed standby configuration are exactly as described earlier, in the sections on hot and warm standbys. For example, if you were running DB2 Enterprise ([Figure 8](#)), and each machine is an unpartitioned two-socket POWER7 dual core server rated at 100 PVU per core, you would need $(4 \times 100) = 400$ PVU entitlements for the primary server and another 400 PVU entitlements for the hot standby server. For each of the warm standby servers, you would need an additional $(1 \times 100) = 100$ PVU entitlements. Therefore, you would need 1000 DB2 Enterprise PVU entitlements in total for this configuration.

If, instead, you were licensing DB2 Enterprise under the AUSI metric and had 88 authorized users who would be accessing Data server 1, and 10 authorized users who would be accessing Data server 2 under normal conditions, you would need $(25 \times 4) = 100$ AUSI entitlements for the primary server and another $(25 \times 4) = 100$ AUSI entitlements for the hot standby server. For each of the warm standby servers, you would need an additional $(25 \times 1) = 25$ AUSI entitlements. Therefore, you would need 250 DB2 Enterprise AUSI entitlements in total for this configuration.

There are various ways to implement this configuration. However, the most straightforward way would be to use the new HADR multiple standby capability that was introduced in all DB2 10.1 editions. This function enables you to create a cluster with a primary server and up to three standby servers, all using the same proven and easy to use HADR functionality. (Prior to DB2 10.1, a HADR cluster could have only one primary and one standby server.) Moreover, each of the standby servers can have a different "temperature", independently of the other standby servers in the cluster.

With HADR, the primary server (Data server 1 in [Figure 8](#)) ships logs at transaction boundaries to Machines 2, 3, and 4. Data server 2 also uses the read on standby capability to handle read workloads. If Data server 1 fails, the TSA functionality included with HADR automatically reroutes Data server 1's workload to Data server 2. Upon taking over this work, Data server 2 stops processing its original read workload so that it can focus exclusively on its new workload. If Data server 2 also fails, your DBA manually fails over the workload to Machine 3. In the worst case scenario, where you lose your primary site, including everything that was on Machines 1, 2, and 3, your remote standby Machine 4 would be up-to-date, except perhaps for any logs that were in flight when your primary site went down.

Reach higher - availability, that is

You have a lot of HA options with DB2. We recommend that you create multi-tier classifications for applications that require availability; for example, Bronze, Silver, and Gold ratings. Perhaps it's the case that anything rated Bronze would leverage a cold standby and a DB2 integrated cluster; a Silver rated solution would leverage DB2 HADR technology; and a Gold rated solution would be a hot standby configuration such as DB2 pureScale. We've put together a "cheat sheet" of some of the possible HA solutions and their respective categorization in [Table 3](#).

Table 3. Addressing your HA requirements with a tiered solution

Bronze	Silver	Gold
DB2 integrated clustering	HADR	DB2 pureScale
<ul style="list-style-type: none"> Hot/cold Basic availability solution Free in most cases Provides failover, typically in less than 5 minutes 	<ul style="list-style-type: none"> Hot/warm (optional hot with read on standby) Provides very fast failover, typically in less than 1 minute Easy to setup with turnkey availability Minimal licensing on the standby server Potential zero data loss option 	<ul style="list-style-type: none"> Hot/hot Online failover Transactions on surviving nodes are not impacted Provides the fastest failover, typically in less than 30 seconds Online scaleout capability to address workload spikes

Everything doesn't always need to be a gold-level solution. People think that they need 24/7 availability for all applications, but that's just not the case.

For example, one client had a mission-critical application that needed 24x7 availability, but their other applications could absorb as much as a two-day outage. Would it make sense to treat them all the same from an availability perspective? If you have an unlimited budget, perhaps. Choose the availability solution that matches the service level agreement (SLA) that's backing your job: HA is a linear cost curve, which means that the more redundancy and availability you want, the more it will cost from a hardware and software perspective. Beyond the software, the more HA you want, the more you typically end up paying (think redundancy, licensing, power, and more). The bottom line is not everything needs to be ultra-available; be smart and definitely leverage the right technology for the right HA requirements.

Table 4 contains a "cheat sheet" for HA's "cousin", disaster recovery (DR). Use the same considerations that you would use to determine appropriate licensing for HA and apply them to DR, as we show in the discussion around the scenario in Figure 8.

Table 4. Addressing your DR requirements with a tiered solution

Bronze	Silver	Gold
Q replication	HADR physical replication	DB2 pureScale with HADR
<ul style="list-style-type: none"> Hot/hot Unrestricted access to the target database Supports different versions of DB2 on source and target Supports sub-setting of data Multiple target support Online failover Can replicate to a different topology Async only Can be complex to set up No DDL support 	<ul style="list-style-type: none"> Hot/warm Zero data loss option Provides very fast failover, typically in under 1 minute Easy to set up with turnkey availability DDL and DML replication support Limited use of the standby Multiple standby support Full database replication only Can be configured with time delay apply 	<ul style="list-style-type: none"> Hot/hot Zero data loss option Online failover Provides the fastest failover, typically in less than 30 seconds Online scaleout capability to address workload spikes Unrestricted access to the target Can be configured with time delay apply Single standby support DDL and DML replication support Online fix pack updates Can replicate to a different topology

Resources

Learn

- See [Compare the distributed DB2 10.5 servers](#) to understand the basic licensing rules, functions, and feature differences between the members of the distributed DB2 10.5 server family described in a side-by-side comparison table.
- See [Which distributed edition of DB2 10.5 is right for you?"](#) to learn the details on what makes each edition of DB2 10.5 unique and the changes made to DB2 editions.
- See [DB2 and IBM Processor Value Unit \(PVU\) pricing](#) for a description on how to license DB2 products using the new Value Unit metric as well as a number of helpful everyday scenarios.
- Visit the [developerWorks resource page for DB2 for Linux, UNIX, and Windows](#) to read articles and tutorials and connect to other resources to expand your DB2 skills.
- Learn about [DB2 Express-C](#), a no-charge version of DB2 Express Edition for the community.
- Visit the [developerWorks Information Management zone](#). Find more resources for DB2 developers and administrators.
- Stay current with [developerWorks technical events and webcasts](#) focused on a variety of IBM products and IT industry topics.
- Follow [developerWorks on Twitter](#).
- Watch [developerWorks on-demand demos](#) ranging from product installation and setup demos for beginners, to advanced functionality for experienced developers.

Get products and technologies

- Build your next development project with [IBM trial software](#), available for download directly from developerWorks.
- Download a free trial version of [DB2 for Linux, UNIX, and Windows](#).
- Download [DB2 Express-C](#), a no-charge version of DB2 Express Edition for the community that offers the same core data features as DB2 Express Edition and provides a solid base to build and deploy applications.
- [Evaluate IBM products](#) in the way that suits you best: Download a product trial, try a product online, or use a product in a cloud environment.

Discuss

- [Participate in the discussion forum for this content](#).
- Get involved in the [My developerWorks community](#). Connect with other developerWorks users while exploring the developer-driven blogs, forums, groups, and wikis.

About the authors

Amyris Rada



Amyris Rada is a senior writer with the DB2 for Linux, UNIX, and Windows product team at the IBM Canada Lab in Markham, Ontario. She has been part of the DB2 team since 1998, and has held different positions in partner enablement, quality assurance, and information development. Amyris is a Computer Engineer (Simon Bolivar University). She is currently responsible for several content areas for the DB2 Information Center and collaborates with DB2 best practices development. She recently co-authored [Best practices: Physical database design for online transaction processing \(OLTP\) environments](#) and [DB2 best practices: Physical database design for data warehouse environments](#). Before working for IBM, Amyris worked at KL Group and INTERGRAPH.

Roman Melnyk, B.



Roman B. Melnyk, Ph.D., is a senior member of the DB2 Information Development team. Roman edited *DB2 10.5 with BLU Acceleration: New Dynamic In-Memory Analytics for the Era of Big Data* (McGraw-Hill, 2013), *Harness the Power of Big Data: The IBM Big Data Platform* (McGraw-Hill, 2013), *Warp Speed, Time Travel, Big Data, and More: DB2 10 for Linux, UNIX, and Windows New Features* (McGraw-Hill, 2012), and *Apache Derby - Off to the Races* (Pearson Education, 2006). Roman co-authored *DB2 Version 8: The Official Guide* (Prentice Hall Professional Technical Reference, 2003), *DB2: The Complete Reference* (Osborne/McGraw-Hill, 2001), *DB2 Fundamentals Certification for Dummies* (Hungry Minds, 2001), and *DB2 for Dummies* (IDG Books, 2000).

© Copyright IBM Corporation 2013

(www.ibm.com/legal/copytrade.shtml)

Trademarks

(www.ibm.com/developerworks/ibm/trademarks/)