Disaster Recovery as a Cloud Service: Economic Benefits & Deployment Challenges

Timothy Wood, Emmanuel Cecchet, K.K. Ramakrishnan[†],
Prashant Shenoy, Jacobus van der Merwe[†], and Arun Venkataramani

University of Massachusetts Amherst
{*twood,cecchet,shenoy,arun*}*@cs.umass.edu*

[†] *AT&T Labs - Research*
{*kkrama,kobus*}*@research.att.com*

Abstract

Many businesses rely on Disaster Recovery (DR) services to prevent either manmade or natural disasters from causing expensive service disruptions. Unfortunately, current DR services come either at very high cost, or with only weak guarantees about the amount of data lost or time required to restart operation after a failure. In this work, we argue that cloud computing platforms are well suited for offering DR as a service due to their pay-as-you-go pricing model that can lower costs, and their use of automated virtual platforms that can minimize the recovery time after a failure. To this end, we perform a pricing analysis to estimate the cost of running a public cloud based DR service and show significant cost reductions compared to using privately owned resources. Further, we explore what additional functionality must be exposed by current cloud platforms and describe what challenges remain in order to minimize cost, data loss, and recovery time in cloud based DR services.

1 Introduction

Our society's growing reliance on crucial computer systems means that even short periods of downtime can result in significant financial loss, or in some cases even put human lives at risk. Many business and government services utilize Disaster Recovery (DR) systems to minimize the downtime incurred by catastrophic system failures. Current Disaster Recovery mechanisms range from periodic tape backups that are trucked offsite, to continuous synchronous replication of data between geographically separated sites.

A key challenge in providing DR services is to support *Business Continuity*[1] (BC), allowing applications to rapidly come back online after a failure occurs. By minimizing the recovery time and the data lost due to disaster, a DR service can also provide BC, but typically at high cost. In this paper we explore how virtualized cloud platforms can be used to provide *low cost* DR solutions that are *better* at enabling Business Continuity.

Virtualized cloud platforms are well matched to providing DR. The "pay-as-you-go" model of cloud platforms can lower the cost of DR since different amounts of resources are needed before and after a disaster occurs. Under normal operating conditions, a cloud based DR service may only need a small share of resources to synchronize state from the primary site to the cloud; the full amount of resources required to run the application only needs to be provisioned (and paid for) if a disaster actually happens. The use of automated virtualization platforms means that these additional resources can be rapidly brought online once the disaster is detected. This can dramatically reduce the recovery time after a failure, a key component in enabling business continuity.

To explore the potential for using cloud computing as a DR solution, we perform a basic pricing analysis to understand the cost of running cloud-based DR for different application types and backup mechanisms. Our results indicate that some applications can see substantial economic benefits due to the on demand nature of cloud computing platforms. We discuss under what scenarios clouds provide the greatest benefits for DR, and present the limitations of current cloud platform features and pricing schemes. Our end goal is to show how cloud platforms can provide low cost DR services and can be optimized to minimize data loss and recovery time in order to provide both efficient disaster recovery and business continuity.

2 How is DR Done Today?

A typical DR service works by replicating application state between two data centers; if the primary data center becomes unavailable, then the backup site can take over and will activate a new copy of the application using the most recently replicated data. In this work we focus on DR systems with the goal of providing *business continuity*–allowing applications to fail over to a backup site while minimizing service disruptions.

[1] In this work we consider BC to be a stringent form of DR that requires applications to resume full or partial operation shortly after a disaster occurs, and we focus on the software and IT infrastructure needed to support this. In addition, a full BC plan must cover issues related to physical facilities and personnel management.

2.1 DR Requirements

This section discusses the key requirements for an effective DR service. Some of these requirements may be based on business decisions such as the monetary cost of system downtime or data loss, while others are directly tied to application performance and correctness.

Recovery Point Objective (RPO): The RPO of a DR system represents the point in time of the most recent backup prior to any failure. The necessary RPO is generally a business decision—for some applications absolutely no data can be lost (RPO=0), requiring continuous *synchronous replication* to be used, while for other applications, the acceptable data loss could range from a few seconds to hours or even days.

Recovery Time Objective (RTO): The RTO is an orthogonal business decision that specifies a bound on how long it can take for an application to come back online after a failure occurs. This includes the time to detect the failure, prepare any required servers in the backup site (virtual or physical), initialize the failed application, and perform the network reconfiguration required to reroute requests from the original site to the backup site so the application can be used. Depending on the application type and backup technique, this may involve additional manual steps such as verifying the integrity of state or performing application specific data restore operations, and can require careful scheduling of recovery tasks to be done efficiently [7]. Having a very low RTO can enable business continuity, allowing an application to seamlessly continue operating despite a site wide disaster.

Performance: For a DR service to be useful it must have a minimal impact on the performance of each application being protected *under failure-free operation*. DR can impact performance either directly such as in the synchronous replication case where an application write will not return until it is committed remotely, or indirectly by simply consuming disk and network bandwidth resources which otherwise the application could use.

Consistency: The DR service must ensure that after a failure occurs the application can be restored to a consistent state. This may require the DR mechanism to be application specific to ensure that all relevant state is properly replicated to the backup site. In other cases, the DR system may assume that the application will keep a consistent copy of its important state on disk, and use a disk replication scheme to create consistent copies at the backup site.

Geographic Separation: It is important that the primary and backup sites are geographically separated in order to ensure that a single disaster will not impact both sites. This geographic separation adds its own challenges since increased distance leads to higher WAN bandwidth costs and will incur greater network latency. Increased round trip latency directly impacts application response time when using synchronous replication. As round trip delays are limited by the speed of light, synchronous replication is feasible only when the backup site is within 10s of kilometers of the primary. Asynchronous techniques can improve performance over longer distances, but can lead to greater data loss during a disaster. Distance can especially be a challenge in cloud based DR services as a business might have only coarse control over where resources will be physically located.

2.2 DR Mechanisms

Disaster Recovery is primarily a form of long distance state replication combined with the ability to start up applications at the backup site after a failure is detected. The amount and type of state that is sent to the backup site can vary depending on the application's needs. State replication can be done at one of these layers: (i) within an application, (ii) per disk or within a file system, or (iii) for the full system context. Replication at the application layer can be the most optimized, only transferring the crucial state of a specific application. For example, some high-end database systems replicate state by transferring only the database transaction logs, which can be more efficient than sending the full state modified by each query [8]. Backup mechanisms operating at the file system or disk layer replicate all or a portion of the file system tree to the remote site without requiring specific application knowledge [6]. The use of virtualization makes it possible to not only transparently replicate the complete disk, but also the memory context of a virtual machine, allowing it to seamlessly resume operation after a failure; however, such techniques are typically designed only for LAN environments due to significant bandwidth and latency requirements [4, 9].

The level of data protection and speed of recovery depends on the type of backup mechanism used and the nature of resources available at the backup site. In general, DR services fall under one of the following categories:

Hot Backup Site: A hot backup site typically provides a set of mirrored stand-by servers that are always available to run the application once a disaster occurs, providing minimal RTO and RPO. Hot standbys typically use synchronous replication to prevent any data loss due to a disaster. This form of backup is the most expensive since fully powered servers must be available at all times to run the application, plus extra licensing fees may apply for some applications. It can also have the largest impact on normal application performance since network latency between the two sites increases response times.

Warm Backup Site: A warm backup site may keep state up to date with either synchronous or asynchronous replication schemes depending on the necessary RPO. Standby servers to run the application after failure are available, but are only kept in a “warm” state where it may take minutes to bring them online. This slows recovery, but also reduces cost; the server resources to run the application need to be available at all times, but active costs such as electricity and network bandwidth are lower during normal operation.

Cold Backup Site: In a cold backup site, data is often only replicated on a periodic basis, leading to an RPO of hours or days. In addition, servers to run the application after failure are not readily available, and there may be a delay of hours or days as hardware is brought out of storage or repurposed from test and development systems, resulting in a high RTO. It can be difficult to support business continuity with cold backup sites, but they are a very low cost option for applications that do not require strong protection or availability guarantees.

The on-demand nature of cloud computing means that it provides the greatest cost benefit when peak resource demands are much higher than average case demands. This suggests that cloud platforms can provide the greatest benefit to DR services that require warm stand-by replicas. In this case, the cloud can be used to cheaply maintain the state of an application using low cost resources under ordinary operating conditions. Only after a disaster occurs must a cloud based DR service pay for the more powerful–and expensive–resources required to run the full application, and it can add these resources in a matter of seconds or minutes. In contrast, an enterprise using its own private resources for DR must always have servers available to meet the resource needs of the full disaster case, resulting in a much higher cost during normal operation.

2.3 Failover and Failback

In addition to managing state replication, a DR solution must be able to detect when a disaster has occurred, perform a failover procedure to activate the backup site, as well as run the failback steps necessary to revert control back to the primary data center once the disaster has been dealt with. Detecting when a disaster has occurred is a challenging problem since transient failures or network segmentation can trigger false alarms. In practice, most DR techniques rely on manual detection and failover mechanisms. Cloud based systems can simplify this problem by monitoring the primary data center from cloud nodes distributed across different geographic regions, making it simpler to determine the extent of a network failure and react accordingly.

In most cases, a disaster will eventually pass, and a business will want to revert control of its applications back to the original site. To do this, the DR software must support bidirectional state replication so that any new data that was created at the backup site during the disaster can be transferred back to the primary. Doing this efficiently can be a major challenge: the primary site may have lost an arbitrary amount of data due to the disaster, so the replication software must be able to determine what new and old state must be resynchronized to the original site. In addition, the failback procedure must be scheduled and implemented in order to minimize the level of application downtime.

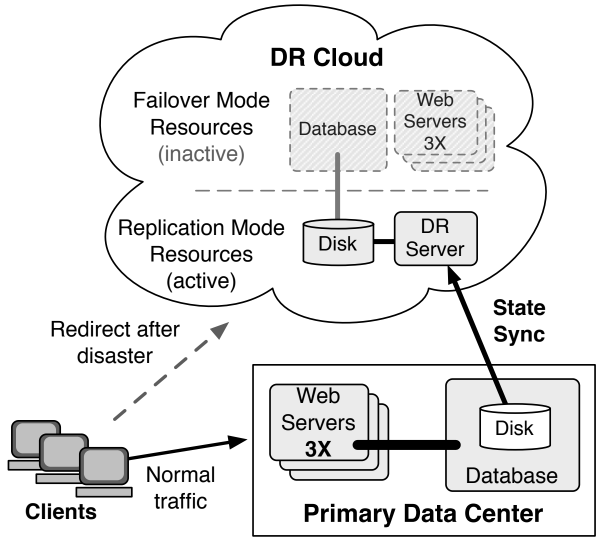


Figure 1: RUBiS is configured with 3 web servers and 1 database at the primary site. In ordinary operation the cloud only requires a single DR server to maintain database state, and only initializes the full application resources once a disaster occurs. After the failure, client traffic must be redirected to the cloud site.

3 DR as a Cloud Service

While there are many types of DR that can be provided using cloud resources, we focus on a warm standby system where important application state is continuously replicated into the cloud. Figure 1 illustrates this setup for a web application that requires four servers (one database and three web servers) in the primary site. Within the cloud providing DR, the level of resources required depends on whether it is in Replication Mode or Failover Mode. During normal operation, the system stays in *Replication Mode*, and requires only a single low cost VM to act as the DR Server that handles the state synchronization. When a disaster occurs, the system enters *Failover Mode*, which requires resources to support the full application. In this section we analyze the costs of this form of DR and discuss both the benefits and challenges remaining for DR in the cloud.

3.1 Are Clouds Cheaper for DR?

We first study the costs associated with disaster recovery services to understand if clouds can actually make DR cheaper. We compare the cost of running a DR service

	Public Cloud		**Colocation**	
RUBiS	Replication	Failover	Replication	Failover
Servers	$2.04	$32.64	$26.88	$26.88
Network	$0.54	$18.00	$1.16	$39.14
Storage	$1.22	$1.39	–	–
Total per day	$3.80	$52.03	$28.04	$66.01
Total per year	$1,386	$18,992	$10,234	$24,095
99% uptime cost	**$1,562 per year**		**$10,373 per year**	

(a)

Resource Consumption

	Replication	Failover
Servers	1 cloud / 4 colo	4
Network	5.4 GB/day	180 GB/day
Storage	30 GB	30 GB
IO	130 req/sec	150 req/sec

(b)

Figure 2: (a) Cost per day and year for providing DR services for RUBiS. Under normal operation, only the Replication Mode cost must be paid, leading to substantial savings when using a cloud platform. (b) Resources required during Replication and Failover Modes are the same for the cloud and colocation center except that the colo center must always have 4 servers available.

using public cloud resources against a "build your own" DR service using an enterprise's own private resources. To estimate the cost of the latter approach, we use the price of renting resources from a colocation facility. This is a reasonable estimate for small to medium size businesses which may own a single data center but cannot afford the additional expense of a second full data center as a DR site.

Our cost study is meant to be illustrative rather than definitive—we found a wide range of prices for both cloud and colo providers, and we do not include factors such as management costs which may not be equivalent in each case. While large enterprises that own multiple data centers may be able to obtain cheaper resources by running DR between their sites, they will still face the same cost model as the colocation facility. Past cost studies indicate that the primary costs of running a private data center are for purchasing servers and infrastructure—costs that do not change regardless of whether servers are actively used or not [5]. In contrast, the cloud's pay-as-you-go model benefits users who can turn resources on and off as needed, which is exactly the case in disaster recovery services that acquire resources on demand only after a failure occurs.

3.1.1 Case Study: Multi-tier Web Application

To understand the cost of providing DR in the cloud, we first consider a common multi-tier web application architecture composed of several web front ends connected to a database server containing the persistent state for the application. This scenario illustrates how some components of an application may have different DR requirements. The web servers in this example contain only transient state (e.g., session cookies that can be lost without significantly disrupting the application) and only require a weak backup policy; we assume that all the front ends can be recreated from a template image stored in the backup site and do not require any other form of synchronization. The database node, however, requires stronger consistency and uses a disk based replication scheme to send all writes to a VM in the backup site. Applications such as this are a natural fit for a cloud based DR service because fewer resources are required to replicate the important state than to run the full application.

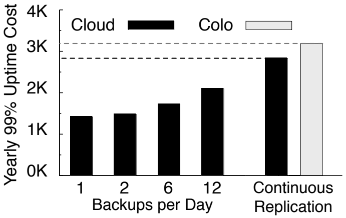
To analyze the cost of providing DR for such an application, we calculate the Replication Mode and Failover Mode costs of running DR for the RUBiS web benchmark. RUBiS is an e-commerce web application that can be run using multiple Tomcat servers and a MySQL database [3]. Figure 1 shows RUBiS's structure and how it replicates state to the cloud. We calculate costs based on resource usage traces recorded from running RUBiS with 300 clients, and prices gathered from Amazon's Cost Comparison Calculator [1]; we have validated that the colocation pricing information is competitive with offerings from other providers.

Cost Breakdown: Figure 2(a) shows the yearly cost for running the DR service with a public cloud or a private colocation facility. The server cost only requires one "small" VM to run the DR server in Replication mode in the cloud whereas the colocation DR approach must always be provisioned with the four "large" servers needed to run the application during failover. Figure 2(b) shows the resource requirements for both modes. The network and IO consumption during failover mode includes the web traffic of the live application with clients whereas the replication mode only includes the replicated state persisted to the database. The storage cost for EC2 is based on EBS volumes (Amazon's persistent storage product) and IO costs, whereas the colocation center storage cost is included as part of the server hardware costs.

99% Uptime Cost: Since disasters are rare, most of the time only the Replication Mode cost must be paid. The best way to compare total costs is thus to calculate the yearly cost of each approach based on a certain level of downtime caused by disasters. Assuming a 99% uptime model where a total of 3.6 days of downtime is handled by transitioning from Replication to Failover Mode, the yearly cost of the cloud DR service comes to only $1,562, compared to $10,373 with the colocation provider—an 85% reduction (Figure 2a). This illustrates the benefit of the cloud's pay-as-you-go pricing model—

	Public Cloud		**Colocation**	
Data Warehouse	Replication	Failover	Replication	Failover
Servers	$4.08	$12.00	$8.51	$8.51
Network	$0.10	$0.12	$0.22	$0.26
Storage	$3.50	$3.92	–	–
Total per day	$7.68	$16.04	$8.73	$8.77
Total per year	$2,802	$5,853	$3,186	$3,202
99% uptime cost	**$2,832 per year**		**$3,186 per year**	

(a)



(b)

Figure 3: (a) Cost for providing DR services for the data warehouse application. The cloud provides only moderate savings due to high storage costs. (b) Using periodic backups can significantly lower the price of DR in the cloud by reducing the cost of VMs.

substantial savings can be achieved if the cost to synchronize state to a backup site is lower than the cost of running the full application.

Cost of Adding DR: Our analysis so far considered the primary site to run on the user's own private resources, but they could also be run in the cloud. However, simply using cloud resources does not eliminate the need for DR—it is still critical to run a DR service to ensure continued operation if the primary cloud provider is disrupted. Running the whole application in the cloud costs $18,992 per year and using cloud DR in addition only adds 8%. Running the application in a colo center costs more in the first place ($24,095 per year) but adding DR in a second colo facility increases the total cost by almost 42%. Finally, if a colocation center is used for the primary site but a cloud is used for DR, then the incremental cost of having DR is only 6.5%.

3.1.2 Case Study: Data Warehouse

Our second case study analyzes the cost of providing DR for a Data Warehouse application. A data warehouse records data such as a stream of website clicks or sales information produced by other applications. Data is typically appended to the warehouse at regular intervals, and reports are generated based on the incoming and existing data sets. We consider a small sized Data Warehouse with a 1TB capacity that adds 1 GB of new data per day. To run the full application, a powerful server is required– we estimate costs based on a "High-Memory Extra Large Instance" from EC2.

Cost Breakdown: Figure 3(a) shows the cost for running the data warehouse application. We assume that the cloud based DR system requires a "medium" size VM as a backup server due to its IO intensive nature, resulting in a relatively high server cost even under normal operation. Additionally, the cloud must pay a large storage cost to support the 1TB capacity of the data warehouse. As a result, the cloud based DR service provides a smaller benefit because its Replication Mode cost is only slightly lower than the cost in a colocation facility, and its Failover Mode cost is significantly higher.

99% Uptime Cost: By comparing the Failover Mode costs, it is clear that it is cheapest to use a colocation center as the primary site of the data warehouse ($5,853 per year in the cloud versus $3,202 per year in a colocation center). However, since the replication cost for the cloud is lower and is incurred for 99% of the time, the total costs is still lower for the cloud. Despite having a higher Failover Mode price, the cloud based DR system still lowers the total DR cost from $3,186 to $2,832 over a one year period assuming 99% uptime.

Periodic Backups: The data warehouse application obtains a smaller economic benefit from the cloud than seen in the multi-tier web application case study due to its increased server and storage requirements during ordinary operation. However, the flexibility of cloud resources can help reduce this cost if the application can tolerate a weaker RPO. For example, it may be sufficient to only send periodic backups to the cloud site once every few hours or after each bulk load, rather than running the DR service continuously. Assuming that one hour of VM time is charged per backup, Figure 3(b) shows how the cost of DR can be substantially lowered by reducing the backup frequency. While a similar approach could be used in a private data center to reduce energy consumption, it would have a much smaller effect on overall cost since power usage of individual servers is a minor fraction compared to the cost of hardware and space that must be paid regardless of whether a machine is in use or not.

3.2 Benefits of the Cloud

Under current pricing schemes, cloud based DR services will not see much benefit when used for applications that require true "hot" standby servers since this can significantly raise the cost during normal operation. However, for applications that can tolerate recovery times on the order of 200 seconds (a typical VM startup time in the EC2 cloud), substantial savings can be found by utilizing low cost servers while replicating state in ordinary conditions and powerful ones only after a disaster occurs. Cloud DR services may be able to obtain additional economic benefits by multiplexing a single replication server for multiple applications, further lowering the cost of resources under normal operation. For applications with a loose RPO, the cloud can provide even greater benefits by only initiating the replication service a few times a day to create periodic backups.

Cloud computing can facilitate disaster recovery by significantly lowering costs:

- The cloud's pay-as-you go pricing model significantly lowers costs due to the different level of resources required before and during a disaster.
- Cloud resources can quickly be added with fine granularity and have costs that scale smoothly without requiring large upfront investments.
- The cloud platform manages and maintains the DR servers and storage devices, lowering IT costs and reducing the impact of failures at the disaster site.

The benefits of virtualization, while not necessarily specific to cloud platforms, still provide important features for disaster recovery:

- VM startup can be easily automated, lowering recovery times after a disaster.
- Virtualization eliminates hardware dependencies, potentially lowering hardware requirements at the backup site.
- Application agnostic state replication software can be run outside of the VM, treating it as a black box.

These characteristics can simplify the replication and deployment of resources in a cloud DR site, and enable business continuity by reducing recovery times.

4 Challenges for the Cloud Provider

Although cloud-based DR can provide economic benefits for a customer, such a service raises numerous challenges for a cloud provider, as discussed next.

4.1 Handling Correlated Failures

Typically a cloud provider will attempt to statistically multiplex its DR customers onto its server pool. Such statistical multiplexing assumes that not all of its customers will experience simultaneous failures, and hence the number of free servers that the cloud providers must have available is smaller than the peak needs of all its customers. However, correlated failures across customers is not uncommon—for instance, an electric grid failure or a natural disaster such as a flood can cause a large number of customer from a geographic area to simultaneously failover to their DR sites. To prevent such correlated failures from stressing any one data center, the cloud provider should attempt to distribute its DR customers across multiple data centers in a way that minimizes potential conflicts—e.g. multiple customers from the same geographic region should be backed up to different cloud data centers. This placement problem is further complicated by constraints such as limits on latency between the customer and cloud site. To intelligently address this issue, the cloud provider must employ risk models—not unlike ones used by insurance companies—to (i) estimate how many servers should be available in a data center for a certain group of customers and (ii) how to distribute customers from a region across different data center sites to "spread the risk". In the event of stress on any single data center due to correlated failures, dynamic migration of a group of customers to another site can be employed.

To achieve all of these tasks seamlessly, the cloud provider should be able to treat all of its data centers as a single pool of resources available to its DR customers [10, 2]. In practice, current data centers act as isolated entities and it is non-trivial to move or replicate storage and computation resources between data centers. We believe that future cloud architectures will rely on network virtualization to provide seamless connectivity between data centers, and wide-area VM and storage migration to allow for resource management across data center sites.

4.2 Revenue Maximization

The DR strategies we have discussed assume that customers only pay for the majority of their DR resources *after* some kind of failure actually occurs, and that sufficient resources are always available when needed. The cloud service provider must maintain these resources and pay for their upkeep at all times, regardless of whether a customer has experienced a failure. Since disasters are typically rare, there will be little or no revenue from the server farm in the normal case when there are no failures. Hence, a cloud provider must find ways to generate revenue from such idling resources in order to make its capital investments viable.

We assume that a cloud DR provider will also offer traditional cloud computing services and rent its resources to customers for non-DR purposes. In this case, the cloud may be able to "double book" its servers for both regular and DR customers. Public clouds generally only offer best effort service when new VM or network resources are requested. While this is sufficient for general cloud computing, in disaster recovery it is imperative that additional resources be available within the specified RTO. One existing pricing mechanism that would facilitate this on demand access to resources is the use of "spot instances". Spot instances allow the service provider to rent resources, typically at a lower price, without guarantees about how long they will be available. A cloud service could generate revenue from idling DR servers by offering them as spot instances to non-DR customers and reclaim them on-demand when these servers are needed for high priority DR customers.

Currently, cloud platforms often provide few guarantees about server and bandwidth availability and network quality of service, which are important for ensuring an application can fully operate after failover. EC2 currently supports “reserved” VM instances that are guaranteed to be available, but they are primarily designed for users who know that they will be actively running a VM for a long period of time, and their pricing is designed to reflect this with a moderate yearly fee but cheaper hourly costs. For disaster recovery, it may be desirable to allow for “priority resources” which are guaranteed to be available on demand, although perhaps at a higher hourly cost than ordinary VM instances or network bandwidth (which also increases the revenue for the cloud provider while providing better assurances to a customer).

4.3 Mechanisms for Cloud DR

While cloud computing platforms already contain many useful features for supporting disaster recovery, there are additional requirements they must meet before they can provide DR as a cloud service.

Network Reconfiguration: For a cloud DR service to provide true business continuity, it must facilitate reconfiguring the network setup for an application after it is brought online in the backup site. We have previously proposed how a cloud infrastructure can be combined with virtual private networks (VPNs) to support this kind of rapid reconfiguration for applications that only communicate within a private business environment [10]. Public Internet facing applications would require additional forms of network reconfiguration through either modifying DNS or updating routes to redirect traffic to the failover site. To support any of these features, cloud platforms need greater coordination with network service providers.

Security & Isolation: The public nature of cloud computing platforms remains a concern for some businesses. In order for an enterprise to be willing to fail over from its private data center to a cloud during a disaster it will require strong guarantees about the privacy of storage, network, and the virtual machine resources it uses. Likewise, clouds must guarantee that the performance of applications running in the cloud will not be impacted by disasters affecting other businesses.

VM Migration & Cloning: Current cloud computing platforms do not support VM migration in or out of the cloud. VM migration or cloning would simplify the failback procedure for moving an application back to its original site after a disaster has been dealt with. This would also be a useful mechanism for facilitating planned maintenance downtime. The Remus system [4] has demonstrated how a continuous form of VM migration can be used to synchronize both memory and disk state of a virtual machine to a backup server. This could potentially allow for full system DR mechanisms that allow completely transparent failover during a disaster. To support this, clouds must expose additional hypervisor level functionality to their customers, and migration techniques must be optimized for WAN environments.

5 Ongoing Work and Conclusions

We have argued that cloud computing platforms are an excellent match for providing disaster recovery services due to their pay-as-you-go pricing model and ability to rapidly bring resources online after a disaster. The flexibility of cloud resources also allows enterprises to make a trade off between data protection and price to an extent not possible when using private resources that must be statically provisioned. We have compared the costs of running DR services using public cloud or privately owned resources, and shown cost reductions of up to 85% by taking advantage of cloud resources.

In our ongoing work, we are developing Dr. Cloud, a prototype DR system that we can use to understand the potential for using existing cloud platforms to provide DR. This will allow us to better understand what features and optimizations must be included within the cloud platform itself, and to explore the tradeoffs between cost, RPO, and RTO in a cloud DR service.

Acknowledgements: This work was supported in part by NSF grants CNS-0720271, CNS-0720616, CNS-09169172, and CNS-0834243, as well as by AT&T. We also thank our reviewers for their comments and suggestions.

References

[1] Aws economics center. `http://aws.amazon.com/economics/`.

[2] Rajkumar Buyya, Rajiv Ranjan, and Rodrigo N. Calheiros. InterCloud: Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services. In *The 10th International Conference on Algorithms and Architectures for Parallel Processing*, Busan, Korea, 2010.

[3] Emmanuel Cecchet, Anupam Chanda, Sameh Elnikety, Julie Marguerite, and Willy Zwaenepoel. Performance Comparison of Middleware Architectures for Generating Dynamic Web Content. In *4th ACM/IFIP/USENIX International Middleware Conference*, June 2003.

[4] Brendan Cully, Geoffrey Lefebvre, Dutch Meyer, Mike Feeley, Norm Hutchinson, and Andrew Warfield. Remus: High availability via asynchronous virtual machine replication. In *Proceedings of the Usenix Symposium on Networked System Design and Implementation*, 2008.

[5] Albert Greenberg, James Hamilton, David A. Maltz, and Parveen Patel. Cost of a cloud: Research problems in data center networks. In *ACM SIGCOMM Computer Communications Review*, Feb 2009.

[6] Kimberley Keeton, Cipriano Santos, Dirk Beyer, Jeffrey Chase, and John Wilkes. Designing for Disasters. *Conference On File And Storage Technologies*, 2004.

[7] Kimberly Keeton, Dirk Beyer, Ernesto Brau, Arif Merchant, Cipriano Santos, and Alex Zhang. On the road to recovery:restoring data after disasters. *European Conference on Computer Systems*, 40(4), 2006.

[8] Tirthankar Lahiri, Amit Ganesh, Ron Weiss, and Ashok Joshi. Fast-Start:quick fault recovery in oracle. *ACM SIGMOD Record*, 30(2), 2001.

[9] Vmware high availability. `http://www.vmware.com/products/high-availability/`.

[10] T. Wood, A. Gerber, K. Ramakrishnan, J. Van der Merwe, and P. Shenoy. The case for enterprise ready virtual private clouds. In *Proceedings of the Usenix Workshop on Hot Topicsin Cloud Computing (HotCloud), San Diego, CA*, June 2009.