Cloud
Standards
Customer
Council

Migrating Applications to the Cloud: Assessing Performance and Response Time Requirements

October, 2014

Contents

© 2014 Cloud Standards Customer Council.

All rights reserved. You may download, store, display on your computer, view, print, and link to the *Migrating Applications to the Cloud: Assessing Performance and Response Time Requirements* white paper at the Cloud Standards Customer Council Web site subject to the following: (a) the document may be used solely for your personal, informational, non-commercial use; (b) the document may not be modified or altered in any way; (c) the document may not be redistributed; and (d) the trademark, copyright or other notices may not be removed. You may quote portions of the document as permitted by the Fair Use provisions of the United States Copyright Act, provided that you attribute the portions to the Cloud Standards Customer Council *Migrating Applications to the Cloud: Assessing Performance and Response Time Requirements (2014)*.

Acknowledgements

The major contributors to this whitepaper are: Mark Houghtlin (IBM) and Mike Edwards (IBM). Several members of the CSCC Practical Guide to Cloud Computing working group provided valuable feedback.

Executive Overview

In the CSCC white paper *Migrating Applications to Public Cloud Services: Roadmap to Success [1],* steps are outlined and highlighted that end users should take to ensure successful migration to cloud computing:

1. ***Assess your Applications and Workloads***
2. Build the Business Case
3. Develop the Technical Approach
4. ***Adopt a Flexible Integration Model***
5. Address Security and Privacy Requirements
6. Manage the Migration

In this supplement to that white paper, we expand upon the importance of thoroughly assessing performance and response time requirements and the potential effects of increased network latency as it relates to both step 1 and step 4. A best practice perspective of how to perform this type of analysis is highlighted. Emphasis is placed on mapping business requirements to the underlying technology to help improve decisions regarding the suitability of cloud computing for a particular workload[1].

By more fully testing and quantifying performance and response time implications early on in the migration process, performance issues that could compromise successful cloud migration can be avoided or mitigated.

To get the most out of this supplement it is recommended that you read the original white paper [1], or review the paper if you have already read it.

Considerations and Motivations

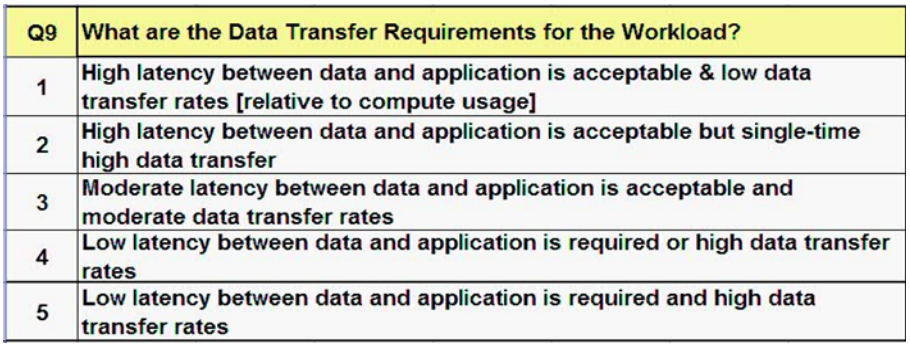
Assessing applications and workloads for readiness for migration to cloud computing allows organizations to determine which applications and data can – and which cannot – be readily moved to a cloud computing environment and which delivery models (public, private, or hybrid) can be supported. Generally speaking, some applications and workloads are more suitable for cloud computing than others. A cost-benefit model indicates the relative difficulty in moving an application to a cloud service versus the anticipated business value in making that move. Factors affecting this model include the response time implications; integration with other applications and data sources and the complexity of the integration (each integration is also referred to as a ***connection***).

It is common for an application being migrated to a cloud service to have connections of various kinds with other applications and systems. Application owners need to understand and address the impact of these connections. Consider what happens if an application is migrated to a cloud service and the other applications it has connections to remain in-house. What protocols do these applications use to talk to

[1] In this paper, the term *workload* refers to a business application, which is a unit of interacting software that will typically be moved as a single component, independently of other business applications.

each other? Can the same connection continue working, that is, is the protocol supported by the cloud service, and if it is, does the response time remain adequate, given the frequency of transactions, the size of the payload, the bandwidth and latency of the connection to and from the cloud service? To answer these questions accurately and with confidence requires a detailed assessment of the performance and response time requirements the connection must support.

When considering workloads for cloud suitability, different cost-benefit analysis methods can be employed. What method is chosen will dictate the level of confidence in the results. For example, a high-level workload suitability assessment may derive any concerns regarding potential response time impact from generalized questionnaire driven data collection methods, such as shown in Table 1. The answers to such a questionnaire impact suitability decisions – both for and against. It may be that the person completing the questionnaire is just providing their best guess, which may not be accurate enough for good decision making. These types of assessments may also deploy auto discovery tools like Configuration Management Database (CMDB) for uncovering application connections[2]. Amongst other considerations, typically the more connections discovered or required, the less suitable the application is for cloud migration. In other words, the cost to migrate to cloud computing may outweigh the benefit.



Q9	What are the Data Transfer Requirements for the Workload?
1	High latency between data and application is acceptable & low data transfer rates [relative to compute usage]
2	High latency between data and application is acceptable but single-time high data transfer
3	Moderate latency between data and application is acceptable and moderate data transfer rates
4	Low latency between data and application is required or high data transfer rates
5	Low latency between data and application is required and high data transfer rates

Table 1: Data Transfer Requirements

The shortcoming of either of these approaches (questionnaire, discovery tool driven) is that they only identify the high-level technological considerations of the application and its associated connections, and they do not assess the business requirements and end-to-end transaction flow. In today's highly distributed environments, each user request can flow through a large number of components (possibly over 50 different servers, application containers and databases) – see Figure 1. This makes it extremely difficult to understand how changes to individual elements affect performance. Managing and tracking transactions end-to-end is necessary to link underlying infrastructure and network components to actual user experience and business requirements. Without considering business requirements and end-to-end flow, an application can be prematurely deemed suitable or unsuitable for cloud migration if assessed by the questionnaire approach or through discovery tools alone.

[2] For a description of CMDB see the Additional References section at the end of this document.

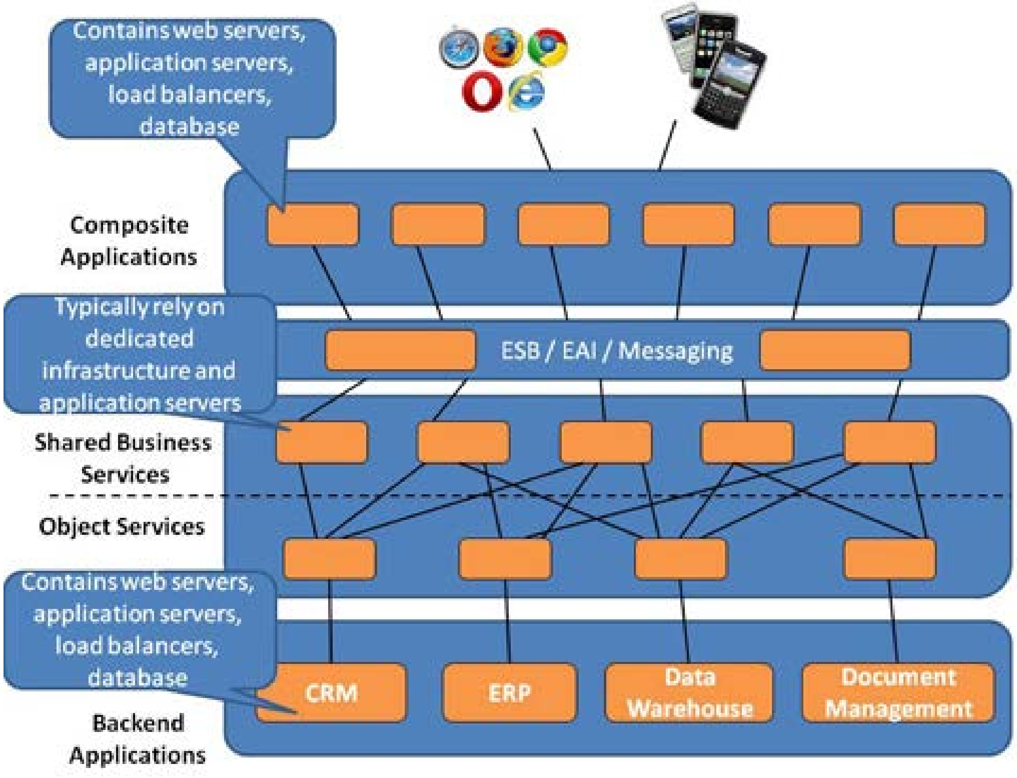

Contains web servers, application servers, load balancers, database
Composite Applications
Typically rely on dedicated infrastructure and application servers
ESB / EAI / Messaging
Shared Business Services
Object Services
Contains web servers, application servers, load balancers, database
Backend Applications
CRM
ERP
Data Warehouse
Document Management

Figure 1: Example environments and layers

Migration Roadmap

This section provides a prescriptive series of steps end users should take when considering migrating existing applications to cloud computing to ensure workload performance and response time requirements are met:

1. Identify business transactions and document their end-to-end application data flow
2. Perform a response time impact risk assessment
3. Perform response time impact testing

Requirements and best practices are highlighted for each step in the sections that follow.

Step 1: Identify Business Transactions and Document their End-to-End Application Data Flow

A better approach than the questionnaire and / or automated discovery method is one which consults with the business to identify individual business transactions, documents their end-to-end flow through the IT infrastructure, and maps the business requirement and importance to the technological considerations supporting it. This includes identifying the end user (or whoever starts the transaction), the transaction characteristics (synchronous real-time versus asynchronous batch, frequency, amount of data moved, network protocol(s) used), the business importance, and the response time sensitivity. This needs to be done for each transaction type which has a unique flow through the supporting IT infrastructure. The difficulty of migration depends on how many interfaces there are, how complex they are, the non-functional requirements which apply and whether integration standards are adopted.

Identifying and understanding specific business transactions is the best way to assess performance requirements. A business transaction is essentially a user request. For example, an eCommerce application might include “Check out” and “Add to Cart” as two important business transactions. Each business transaction includes all of the downstream activities until the end user receives a response and perhaps more, if the application does some additional asynchronous processing not directly part of the response to the user.

As shown in Figure 2, the application may have an implementation of "Check out" which performs request validation, stores the order data into a database, and then starts some asynchronous processing relating to the order by means of publishing a request to a topic in a messaging system, before responding to the user that the order is placed, without waiting for the asynchronous processing to complete. The asynchronous processing may then involve a back-end process which listens to the messaging system topic. This process may request the dispatch of items from the warehouse to the customer via invocation of an external service, and then may store data relating to the order in a Hadoop data store in preparation for deep analysis of customer buying behavior. All of these activities are grouped together into a single business transaction (“Check out”) so that one can understand how every part of the system affects the end users and the response time they receive.

For the sake of our discussion, it is important to understand the effect of separating any elements of the business transaction resulting from the migration to cloud computing. In the eCommerce example above, initially, the front-end application might be migrated to a cloud service while the messaging system and its downstream processing remain on-premises. What would be the response time impact to this business transaction caused by the migration, if any? How would the end users be affected? The best way to answer these questions with confidence is to perform a deeper analysis of the transaction and its response time characteristics, especially at all connection points. There is a strong likelihood in the case of the “Check out” transaction that a longer latency in the processing of the published message by the asynchronous back-end process is likely to have almost no effect on end users - but this can only be understood by careful examination of all the connections.

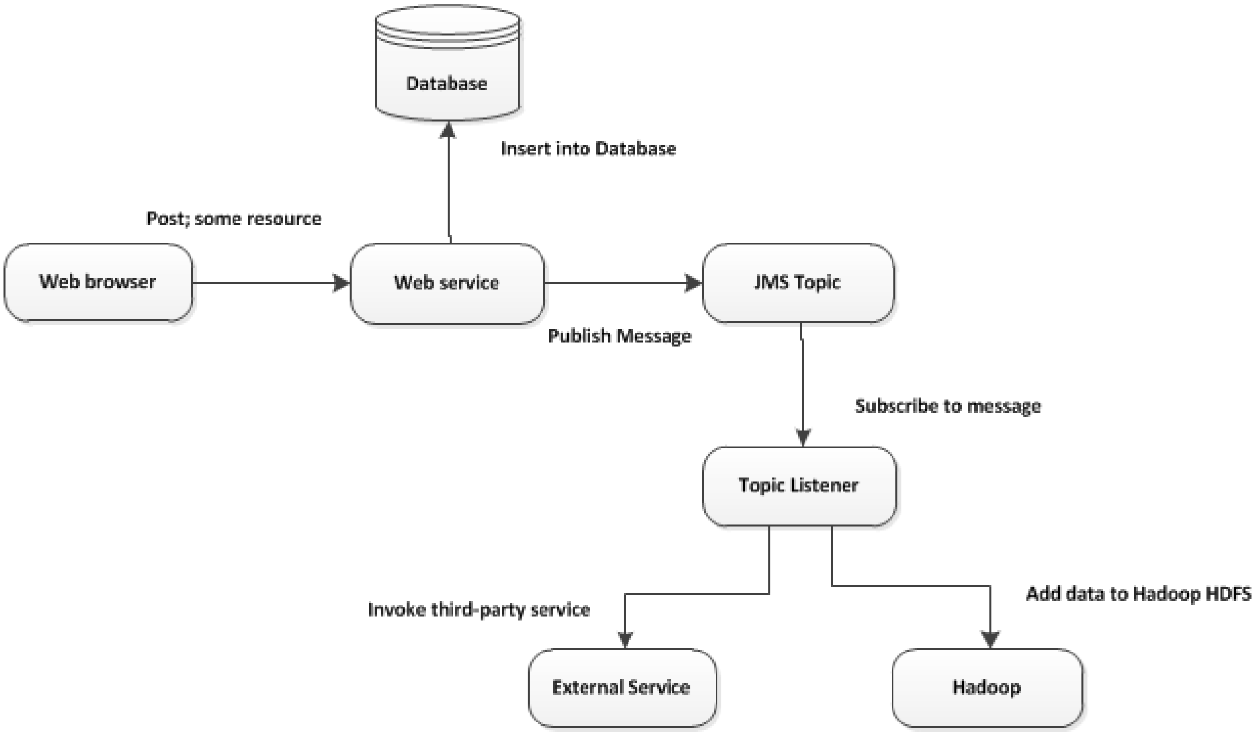


Figure 2: Example business transaction end-to-end flow

At its core, an application and workload are comprised of various types of transactions which may or may not span multiple applications. Not all transaction types have the same performance and response time requirements, and therefore a separate assessment of each transaction is usually necessary. After the transactions have been identified and documented, a risk assessment should be performed in order to evaluate the latency impact for each and to determine if migration to cloud computing is appropriate.

Step 2: Perform a Response Time Impact Risk Assessment

A risk assessment evaluates the business requirement and potential impacts of migrating a business transaction to cloud computing including sensitivity to delay, overall transaction characteristics, the end-to-end flow, and the response characteristics of any integration / connection points of concern. Using established criteria, each transaction should be assigned a rating based on both business importance and sensitivity to delay. Transactions rated as high importance to the business and sensitive to delay are assessed as 'high risk of impact' and should be considered for response time impact testing – see Step 3 below.

Assessments which use questionnaire based and / or automated discovery tools provide little to no business context regarding specific transaction types. The approach described here discovers the business requirement, business importance, and nature of each transaction – ***delivering the insights to make better strategic decisions about cloud migration.*** Another inherent benefit is that the process requires business team participation, bringing them to the table and helping to establish their buy-in to the project.

As an example, given an application with multiple connections to back-end services, one connection may support a synchronous real-time transaction that is response time sensitive and susceptible to an increase in network latency. Another connection point may support an asynchronous batch oriented transaction that is not response time sensitive and is not susceptible to increased network latency, only performed once a week, and only concerned with completing processing within a determined batch window. Considering response time impact and taking business importance into account, the first connection may not be suitable for splitting between cloud and in-house services whereas the second connection potentially could be split.

Most business transactions in current systems span multiple applications and/or shared services. At the connection points, factors such as response time requirements, communication characteristics and network protocol may be different. Using our example from Figure 2, request validation through publishing a request to a topic to the messaging system is synchronous and real-time in nature and therefore considered response time sensitive. However from that point forward the processing becomes asynchronous which is less sensitive to response time delays.

Therefore, a detailed understanding of a business transaction's end-to-end flow is required -- this potentially includes conducting performance testing to truly determine which services are feasible to migrate to cloud computing, and which connections can work from a cloud service to an in-house application. Without considering business requirements and end-to-end flow, an application can be prematurely deemed unsuitable for cloud migration if purely assessed on questionnaire driven merits alone (having earlier established that for most high-level suitability assessments typically the more integrations discovered, the less suitable the application is deemed for cloud migration).

Step 3: Perform Response Time Impact Testing

Once the performance and response time requirements are better understood for any given application and associated connections, response time impact testing is warranted for any proposed changes impacting the more sensitive transactions. Response time impact testing helps to quantify the potential impact of changes and can also potentially identify mitigation opportunities. This form of testing involves first establishing a baseline of current transaction response time and then modeling the impact on response time of changing application or network conditions such as application message size, available network bandwidth, load, and latency.

Assuming like for like in regards to system performance and system type, and assuming that all components of an application would move together, when assessing potential response time impact of cloud migration, there are basically two areas of focus:

1. Presentation / tier 1 services migrate to a cloud service and become remote (WAN) from the end users who use them where they were local (LAN) before.
2. Application-to-application and or shared services connections which were LAN based become separated across a WAN after the migration to a cloud service.

If a workload passes the initial assessment for suitability, then in all likelihood from a presentation / tier 1 services perspective it is already architected to meet performance and response time objectives including those for both local and remote users. However for situations where the presentation / tier 1 services are legacy-based, such as a client server implementations, then application network modeling tools can be employed to establish a baseline. ‘What if’ analytical modeling scenarios can be performed depicting the impact on response times of a network change such as increased network latency – as shown in Figure 3. This includes testing and modeling any application connections of concern.

For response time impact testing, prioritize transactions that are real-time synchronous in nature, important to the business, and considered response time sensitive. Start by documenting the end-to-end flow, identifying areas of concern, and then create a plan for testing. Testing should focus on capturing and analyzing the network flows of concern. Testing also has the advantage of validating transaction end-to-end flow and characteristics. Without testing, an understanding of transaction flow and characteristics may be based on incomplete information and it can be challenging for developers to know how their code and the use of shared services affects performance.

Diagnosing performance issues in pre-production and production environments requires significant efforts from multiple teams and can result in delays in the rollout and delivery of new projects. Testing requires a much deeper review of how things work, and can lead to discoveries that can change opinions regarding the suitability of an application for migration to cloud computing. For example, a transaction that is considered real-time synchronous and response time sensitive, and deemed potentially unsuitable to migrate to cloud computing, may only begin that way – on the front end. However for a particular application connection, deeper inspection may find that the connection is asynchronous in nature, making it suitable for cloud migration. Conversely, the testing might reveal that response time impact is not acceptable and mitigation steps are necessary.

Response time impact testing is most critical for business applications that require a phased migration approach (i.e. not all elements of a business application can migrate at the same time). In today’s complex shared services environment, this is a common area of concern for any data center relocation or consolidation initiative. To minimize risk requires a well planned and well executed migration plan. A better understanding of response time requirements and transaction characteristics help to determine the best migration plan by identifying what must move together and what can be separated. Also, where a response time impact is unavoidable, end user expectations can be set accordingly. In some cases, the business can tolerate a response time impact for an interim period of time. But there is no way of knowing this if it is not quantified and shared.

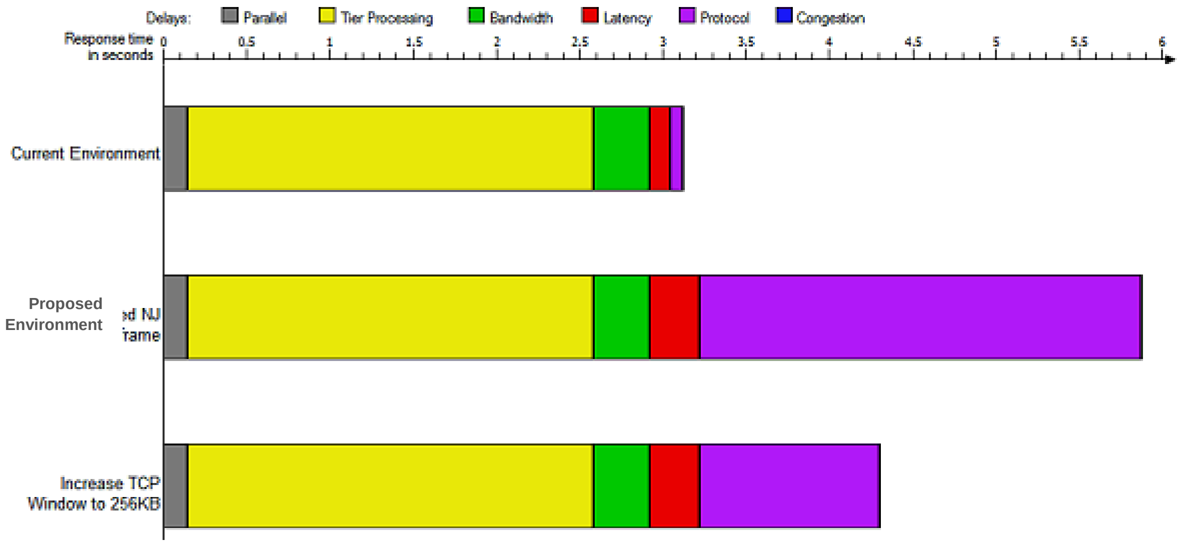


Figure 3: Performance analysis for a migration

Figure 3 illustrates the impact of increasing network latency (second and third bar) on baseline application performance in the current environment (top bar). The colors identify where overall transaction time is spent. The second bar shows the effects protocol overhead has on overall response time if a component of the business application is split across a WAN connection – doubling the response time from 3.1 seconds to 6 seconds. The third bar shows the effects of increasing the TCP window size from the default 64KB to 256KB in order to mitigate some of the effects of separating the application.

With the level and detail of data collected from testing, the business can decide whether the impact of the migration to cloud services will be acceptable or not. Tools meeting these requirements fall under the Application Performance Management (APM) suite, providing performance optimization along with capacity planning, performance monitoring and reporting. Often times, these APM tools will include analysis capabilities which help to identify areas of focus if mitigation steps are necessary (i.e. where time is being spent, efficiency of application and network interaction, programmatic vs. systematic processing delays, etc.). By understanding the focus areas, businesses are in a better position to evaluate the cost of migrating to cloud versus the benefit.

The best time to perform a study of this type is after an initial return on investment study has taken place and shown that there may be business value in pursuing a cloud based strategy, but before final decisions have been made and migration planning has commenced. If mitigation steps are required, then time must be allocated to address these. Customers want to exploit their cloud computing strategy to the fullest extent, but must avoid unwanted surprises which can have both tangible and intangible ramifications. Tools providing the APM capabilities described in the paragraphs above are available from several vendors.[3]

As shown in Figure 4, the different analysis methods deliver different levels of confidence - more due diligence produces more accurate results and lower risk. Customer requirements, project budget, and level of cloud adoption will dictate which method makes the most sense. However, no business wants to suffer damage to their corporate brands by making a poor transition to cloud computing which can impact revenue and customer retention. The tools, processes, and information used in the planning, execution and ongoing management of cloud applications can make the difference between success and failure.

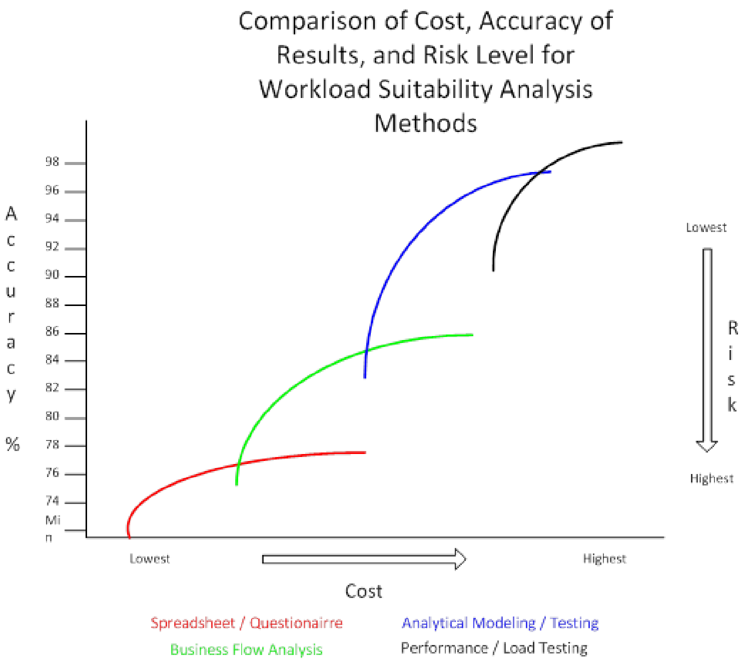


Figure 4: Accuracy against cost of workload suitability analysis

Performance Optimization: Application Performance Management (APM) and Big Data Analytics

Managing the boundary between on-premises applications and applications running in cloud services, including active monitoring, is necessary to ensure the success of a cloud computing initiative. A cloud-based application that fails to meet service level agreements (SLA) and provides poor performance benefits no-one.

[3] Vendors providing APM tools include Riverbed - http://www.riverbed.com/, Compuware - http://www.compuware.com/, and Computer Associates - http://www.ca.com/ to name a few.

Highly virtualized and dynamic cloud architectures may render obsolete the standard tools and processes used to manage a static IT environment. Transactions are the only threads that can provide the visibility needed to manage and monitor a cloud environment and associated cloud based applications. Managing and tracking transactions is the best way to understand what is going on in such sophisticated environments.

Developing business applications today is more complex than when a single application ran on dedicated hardware with its traffic running over a corporate backbone network. Now, applications typically are comprised of multiple components, each running on a different system and interconnected using a variety of company-owned, Internet-based or even cellular networks. To build such applications, developers can use some help. That is where application performance management (APM) solutions come in.

Today's APM technologies combine end user experience monitoring with end-to-end transaction tracking and big data analytics for processing and analyzing the large volume of transaction data collected. APM combined with big data analytics have the potential to substantially improve mean time to resolution (MTTR) for any problems encountered. Whether installed in a cloud computing or on premise production or development environment, the current generation of APM tools and technology offer low overhead instrumentation and highly scalable analytics to account for the volume of data to be processed. Throughout all the phases of a cloud computing project - planning, testing, migration of existing applications, operating and managing - APM can provide visibility into business service / IT dependencies and transaction behavior.

APM tools analyze the performance of applications during load, stress, volume, and scalability scenarios. Performance testing ensures that applications meet the behavioral requirements of prospective users at the appropriate phase of the application development lifecycle. Here are 9 ways developers can use APM solutions to improve the quality and performance of their software:

1. **End-User Experience Monitoring:** End-user experience monitoring is one step to ensuring applications will be well-received by employees, clients, and customers. APM helps developers understand the end-user experience by capturing data on an application's end-to-end performance. This information can be used to identify problems and isolate potential performance bottlenecks.

2. **Deep-Dive Performance Monitoring of Application Components:** Common sense dictates that in order to ensure suitable performance levels for an application comprised of many components, a developer needs data on the performance of each component. An APM solution should offer this functionality to help understand where improvements can be made or problems eliminated. Traditional monitoring metrics (CPU, Memory, Disk I/O, Network I/O, etc…) will help you size your cloud environment but tell you nothing about the actual application performance. It is best to understand performance in terms of business transactions. The important performance metrics encompass end user response time, business transaction

response time, external service response time, error and exception rates, and transaction throughput, with baselines for each.

The best time to optimize the performance and scalability of critical business transactions is early in the application development process, long before the application goes live. Pre-production performance testing is a good time to make sure there are no glaring performance issues that will be introduced into the cloud environment. Cloud auto-scaling decisions are often made based on infrastructure metrics such as CPU utilization. However, in a cloud or virtualized environment, infrastructure metrics may not be reliable enough for making auto-scaling decisions. Auto-scaling decisions based on application metrics, such as request-queue depth or requests per minute, are much more relevant to user experience. If application metrics are not already being monitored by the cloud service provider, then an APM solution can be used to provide them. Once an application is migrated, the resources the application is consuming in the cloud environment must be monitored. New instances of applications can quickly add up to a large expense if the application code is inefficient. Understanding how well an application scales under load and fixing resource hogs is required to drive better value out of the application as usage increases.

Many of the APM vendors offer cloud-ready monitoring capabilities (transaction tracking) that automatically detect and map application flow in real time and depict the environment as a logical topology / flow diagram via a dashboard interface, so you always have an up-to-date view of the application structure. Monitoring the performance of application components can be used to baseline overall performance and proactively detect unusual behavior before it impacts end users. In particular, a solution should provide code-level visibility into components such as Java or .NET applications, web servers, middleware, portals, and commercial application components.

3. **Analysis of Multi-Step Transactions:** Many applications are dependent on the smooth execution of several operations and transactions. From the end user's perspective, it is the cumulative time of all of these transactions that matter. From a developer's perspective, the only way to gain insight into how a multi-step transaction performs is to be able to see the interdependencies of the various components. APM solutions which provide such granular monitoring and management can help a developer better understand where delays might occur. This information can then be used to help improve business transaction response times.

4. **Transaction-Based Troubleshooting:** As noted in the point above, APM solutions that can analyze multi-step transactions can be useful. Even better are solutions that leverage that information and aid in troubleshooting. An ideal solution would help quickly identify the slowest transactions and detect such things as memory leaks which may be negatively impacting transaction performance. A developer can use these details to resolve problems in a shorter time and ideally be more proactive and prevent performance problems from happening in the first place.

5. **Root Cause Analysis:** A key to ensuring optimal performance is the ability to do root cause analysis of end-to-end application sessions. Today, root cause analysis is more complex and harder to conduct because of the dynamic nature of corporate infrastructures. Developers need the ability to isolate performance bottlenecks to individual instances in order to fix issues related to the network, databases, or the application code itself.

6. **SLAs and Metrics for Business Managers:** Developing a great, high-performing application is one thing. Getting recognition for that and convincing business managers is another. APM can be used to provide metrics to show that an application meets the business needs it was created for. For example, APM can help demonstrate that an application meets its agreed service level objectives. Simply put: an SLA, its objectives, and the metrics are an effective way to convey performance information to business managers.

Note: depending upon the cloud deployment model, the degree to which service level objectives can be defined, measured and reported will vary. Straightforward objectives such as availability ("up time") are one thing, but measuring and reporting end-to-end transaction response times are very different. As cloud offerings mature and enterprise acceptance expands, cloud service customers may demand that cloud service providers have SLAs containing service level objectives which cover aspects of application transaction response times.

7. **Analytics and Reporting:** Runtime environments and performance requirements change over time. Instances of application components might be moved from one physical server to another or to a cloud service. A customer service department might experience substantial growth, placing new demands on infrastructure to meet consistent end-user performance levels. APM can help in these areas by gathering performance data over time. Collection of performance information and presentation of that data can help inform the business of future issues that might impact the end-user allowing time to address them before they become a problem.

8. **Addressing the Role of the Internet in an Application's Infrastructure:** The growing use of cloud computing for business applications introduces an interesting performance issue that must be taken into consideration. Cloud-based application component traffic will take various paths over the Internet to the end user. This can introduce unpredictable performance characteristics when trying to measure end-to-end application performance. A performance measurement taken one second can yield a very different value the next second, day, or week. Developers must take this variability and unpredictability into account. One way to do that is to rely on APM to measure or monitor performance over a large period of time and form an average of the discrete measurements.

9. **Mobile User Issues:** Many applications today are accessed by users on their mobile devices. Beyond screen formatting issues, developers must factor in the bandwidth a user with mobile service will have compared to, for example, a desktop user on a wired LAN. APM can help assess the end-to-end performance of a given application for mobile users. In some cases, the mobile

connection might be the limiting factor for performance. If this can be ascertained, and if access by mobile users is essential, APM could help guide design approaches to accommodate these users.

Works Cited

[1] Cloud Standards Customer Council (2011). *Migrating Applications to Public Cloud Services: Roadmap to Success*. http://www.cloudstandardscustomercouncil.org/Migrating-Apps-to-the-Cloud-Final.pdf

Additional References

IBM's Workload Transformation Analysis for Cloud
http://www-935.ibm.com/services/us/en/it-services/cloud-services/workload-transformation-analysis-for-cloud/

Riverbed - Building Better Code: 10 ways to use APM
http://www.riverbed.com/about/document-repository/Building-Better-Code-10-Ways-to-Use-APM.html

Correlsense - Business Need: Cloud computing
http://www.correlsense.com/solutions/business-need/cloud-computing/

Correlsense - Solutions: SOA and Shared Services
http://www.correlsense.com/managing-performance-in-service-oriented-architectures-soa-and-shared-services-environments/

AppDynamics – Managing the Performance of Cloud based Applications-
http://www.appdynamics.com/blog/cloud/managing-performance-cloud-based-applications/

Configuration Management Database (CMDB) - A configuration management database (CMDB) is a repository that acts as a data warehouse for information technology (IT) organizations. Its contents are intended to hold a collection of IT assets that are commonly referred to as Configuration Items (CIs), as well as descriptive relationships between such assets. When populated, the repository becomes a means of understanding how critical assets such as information systems are composed, what their upstream sources or dependencies are, and what their downstream targets are.
http://en.wikipedia.org/wiki/Configuration_management_database

Copyright © 2014 Cloud Standards Customer Council