Enabling Iterative Cloud ROI Analysis with Automated IT Discovery

Thomas E. Chefalas, Meir M. Laker, Chang-Shing Perng, Shu Tao
IBM T. J. Watson Research Center

Abstract—**During the sales engagement between Infrastructure as a Service (IaaS) cloud provider and its customers, analysis on Return On Investment (ROI) in cloud is critical for developing a service solution and making financial investment commitments. However, cloud providers are often challenged in obtaining information about customers' existing IT environments, so as to compare the cost of continuing the current IT environment with that incurred by migrating to a cloud environment. To address this challenge, we propose a methodology of using automated IT discovery tools, implemented as a service appliance, to collect data on customers' premises in a non-intrusive fashion. Realizing that even with automated discovery, such information collection can only be conducted incrementally, we also develop an approach to conducting ROI analysis based on partial IT environment data, while iteratively refining the results with more complete data provided by the discovery tool.**

I. Introduction

During the sales engagement between an IaaS cloud provider and its potential customers, the provider often needs to demonstrate how migrating to cloud is beneficial to the customer. An important step of this is the analysis on Return On Investment (ROI) in cloud. Typical ROI analysis involves the following steps. First, detailed information regarding customer's existing IT environment, such as hardware and software inventory, server configurations, utilization, etc., needs to be obtained to come up with a cost projection for a customer-specified time period. Then, cloud provider maps customer's current IT environment to an equivalent cloud environment configuration that would achieve the same or better application performance. Last, the cost of using the cloud environment is analyzed and compared to that of staying in the traditional IT environment to obtain a set of ROI metrics, including payback period, annual savings, total cost of ownership (TCO), return on investment, etc.

Even though the above steps seem straightforward, there are several challenges in practice that hinder a thorough ROI analysis. The most common challenge is the inability of a provider to obtain complete IT environment data from its customers. As the IT environment becomes increasingly complex and fast-evolving, many customers today do not have up-to-date IT inventory data to allow the basic ROI analysis. Meanwhile, discovery of customer's IT environment is time consuming and often raises security concerns.

To address these challenges, we develop a tool called ServiceStation, that can simplify and automate the IT discovery process. ServiceStation is a pre-loaded appliance that can be quickly deployed in customer environment with little installation and configuration. Once provided with necessary credentials, it automatically discovers and collects configuration information from customer's IT infrastructure, identifying deployed physical servers, network devices, VLANs, software components, etc., as well as the interdependencies among them. ServiceStation uses native management interfaces (i.e.,WMI, SSH, SNMP) to collect data, hence does not impose any overhead on customer hosts. It gives full control to the customer, with only the collected configuration data transferred back to the provider through a secure communication channel, hence greatly eases the security concerns.

The other challenge we need to address is that discovery often has to be conducted iteratively, because different levels of discovery require different levels of input and support from the customer. For instance, a lower level discovery will identify the number of hosts, network topology, etc., without accessing the end hosts. A higher level discovery, with host credentials, can collect information about system configurations, software components, etc. Since the discovery results are input to ROI analysis, this means our ROI model needs to be able to take partial input, yet still generate useful insights, and iteratively improve estimation accuracy when higher-level discovery data becomes available. Our ROI models are exactly developed to work in this fashion: when only partial data is available, the model will extrapolate the missing data based on (i) industry averages, or (ii) data obtained from historical engagements. Iteratively, the model will replace these estimates with more accurate input. As a result, our ROI model can provide a full spectrum of analyses that fit every stage of customer engagement.

Neither ROI analysis nor IT discovery is a new topic. For instance, ROI/TCO analysis for cloud computing has been generally discussed in [1], [2], [3]. A general methodology for building ROI model for IT transformation has been discussed in [4]. IT discovery tools such as IBM's TADDM [5] and their use in collecting IT configuration data and Configuration Management Database (CMDB) [6], [7] have also been a common industrial practice. Our paper's contributions are in developing a novel methodology to enable cloud ROI analysis with automated discovery data, while allowing iterative improvement of accuracy using incremental data input. In the rest of the paper, we will first discuss our discovery tool in Section II, then introduce the iterative ROI analysis in Section III, and finally conclude in Section IV.

978-3-901882-50-0

II. Automated IT Discovery

To enable automated IT discovery, we developed *ServiceStation*, an appliance that can be deployed to customer environment, either as a hardware box or as a virtual machine image. ServiceStation is based on IBM's TADDM technology [5], but customized to collect IT environment information specifically needed for cloud ROI analysis. ServiceStation is pre-loaded and pre-configured, hence requires minimum setup and configuration efforts from customer.

A. ServiceStation Architecture

ServiceStation has a multi-tier architecture, as shown in Fig. 1. It is typically deployed behind customer firewall. The customer provides to ServiceStation with a set of LAN segments, system credentials and a discovery policy that determines when and how deep it will search for end points. ServiceStation then discovers all end points on those LAN segments, communicates with them using common protocols (e.g., WMI for Windows based systems, SSH for Unix based systems and SNMP for network devices) and collects system configuration information. Note that its discovery is performed without requiring software agents to be installed on end hosts. Therefore, the discovery process imposes minimum overhead to customer environment.

The ServiceStation appliance includes the following components: a *discovery engine* that is based on IBM TADDM [5] for discovery, a *data manager* that stores and manages discovered data using IBM DB2 database, a *Web console* for communication with users, implemented using IBM WebSphere Application Server, IBM HTTP Server, Java and PHP; and a *data transfer module* that sends collected data to cloud provider via SSL. The data discovered is structured using IBM's Common Data Model [8], which can be exported in XML and used directly as input to our ROI model.

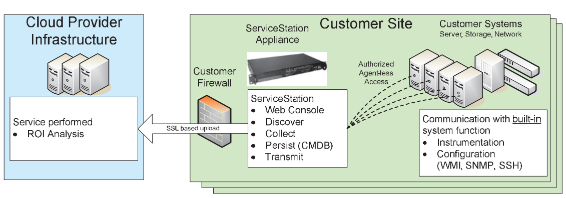

Fig. 1. ServiceStation Architecture

B. Common Data Model

ServiceStation stores data using the IBM Common Data Model (CDM) [8], a standard model for representing system resources and their relationships. It is a logical data model that consistently describes the details of system resources. ServiceStation can identify physical or virtual servers, hypervisors, network devices, VLANs, storage and software components that are active in a runtime environment. It also discovers the interdependencies between various hardware and software components. During the discovery process, ServiceStation uses the CDM to store device identification information (e.g., device type, model, serial number), system inventory information (e.g., firmware levels, operating system, installed software), topology information (i.e., system-level relationships) and application-level configuration and dependencies (e.g., WebSphere server configurations and its dependencies on back-end DB2 servers).

A sample CDM for a Windows server is visualized in Fig. 2. It represents the detailed system-level information, such as the OS, file system, storage volumes and network, as well as application-level information, such as installed DB2 software. The relationships between components are also fully captured in this model.

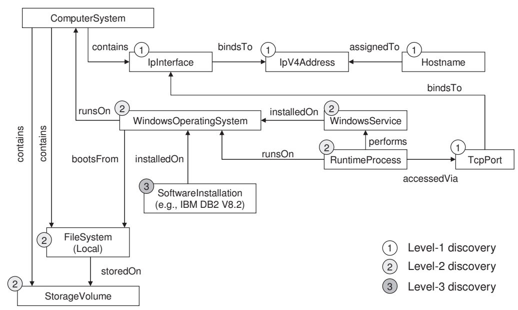

Fig. 2. A sample CDM for a Windows server

C. Iterative Discovery

ServiceStation requires different levels of inputs from customer's IT team to perform various discovery jobs. Depending on which engagement phase it is used for, customer may provide very little or very detailed input. Hence, we configure ServiceStation to run at the following three levels of discovery:

- **Level 1**: network interfaces, IP addresses, ports, hostnames, network topology;
- **Level 2**: hypervisor, hardware configurations, software installed;
- **Level 3**: application-level dependencies.

At level 1, the only input from customer is a range of IP addresses to be scanned in a given network segment. ServiceStation will automatically discover the active network interfaces, associated IP addresses, hostnames, ports, and IP-level network topology. At level 2, customer will provide administrator credentials to allow ServiceStation to log on to running OSes. Therefore, it can discover the OS types, as well as hardware and software configurations of these OSes. At level 3, application-level credentials will be provided for the ServiceStation to examine the dependencies between applications installed across different hosts (e.g., dependencies among HTTP servers, application servers, and database servers for a typical three-tier web application). For example, Fig. 2 shows for a Windows server, what specific information will be discovered at these three levels. As data is discovered by ServiceStation at different levels, it will be iteratively fed as input to ROI analysis.

Note that at level 2 and 3, the access credentials are kept by the customer, and need not be shared with the cloud

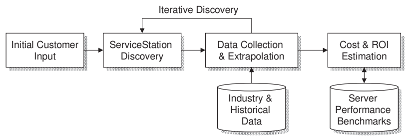

Fig. 3. Iterative IT discovery and ROI analysis

Information	Discovery Level
OS	2
Num of CPU cores and clock rate	2
Memory size and disk space	2
CPU and memory utilization	2
Workload type	3
Server age	other

TABLE I
INFORMATION NEEDED FROM EACH SERVER FOR ROI ANALYSIS

provider. The customer can either load the information into ServiceStation manually or through scripts, or configurate ServiceStation to automatically retrieve access information from an existing LDAP server.

III. ITERATIVE ROI ANALYSIS

The goal of ROI analysis is for a cloud provider to demonstrate to its customer the financial benefits of migrating from its current IT infrastructure to cloud. Depending on the various stages of the customer engagement, the information available for ROI analysis also varies. This in turn determines how the ROI model should be designed. In this section, we introduce a ROI model that can adapt to any engagement stage, with various levels of input available.

As shown in Fig. 3, iterative ROI analysis starts with some initial customer input. This could include the total number of data centers, servers, intended scope of migration to cloud, etc. Then, ServiceStation will perform its discovery procedures, starting from the lowest (easiest) level possible. After finishing each round of discovery, the data will be collected and consolidated for ROI analysis. Obviously, until ServiceStation finishes all levels of discovery, some of the IT environment data will be missing. Therefore, the ROI model also needs to extrapolate these missing values.

A. Data Collection and Extrapolation

As shown in Sec. II, each level of information collected by ServiceStation can be used directly as input to ROI analysis or for data extrapolation. For example, even with only level-1 scan, the obtained IP addresses can be used for estimating server/device count. If we know the number of IP addresses (or equivalently, the number of NICs), the average number of NICs each type of networked equipment has, and an estimated percentage of each type of equipments, then we can estimate the numbers of equipments in each type. Similarly, other levels of discovery can be used to infer further detailed input data for ROI analysis.

To accurately compare the cost of continuing existing IT environment and that of using cloud require very detailed information about IT inventory. To illustrate this, we list in Table I the details required by our analysis for each server. In practice, such detailed information is almost never fully available, regardless of the stage of engagement. Even with the help of ServiceStation, some information (e.g., server age) cannot be automatically discovered. Therefore, we need an extrapolation model to fill in the missing data before feeding it to ROI analysis.

Our data extrapolation model defines a server taxonomy table that stores comprehensive information about typical enterprise IT environments for customer from different industries, as well as data collected from historical cloud engagements. Due to the sensitivity of this data, we can only provide illustrative examples here. For instance, if we know the customer is running certain number of x86 servers, but not the ratio between 32-bit and 64-bit servers, we can estimate this value using the average value obtained from customers in the same industry. Another example is that if we do not know the age of customer's existing servers, we can also combine industry and historical engagement data to derive an estimate.

B. Cost Estimation

The core of ROI analysis is cost estimation and comparison between existing IT environment and cloud, which is conducted in the following three steps:

1) Project future costs of the existing IT environments, based on the input IT inventory data, considering business growth and other financial factors.
2) Map the existing environment to a cloud environment that can support the same workload. Then estimate the migration cost and project the future costs of the cloud environment.
3) Compare the costs of the two environments and calculate return on investment, payback period, internal rate of return, and other financial metrics.

Costs for running existing, non-cloud IT environment can be categorized in Table II. Our cost estimation model is supported by a database that contains detailed specifications about hardware and software from most vendors. Given a hardware's brand and model, we can obtain its typical purchase and maintenance cost. Given a software version, the licensing and service costs are also available from this database. The network cost can be inferred based on average market price, given customer's current bandwidth consumption. Administration labor cost is either provided by customer directly, or inferred from the number of servers and industrial average of server-to-FTE (full time employee) ratios. The facility cost is either from customer or industry average. The Energy cost is estimated based on energy consumption specifications associated with hardware equipments, which are also provided by the aforementioned database.

Cost	Description
Hardware	Acquisition and maintenance costs of servers, network equipments, and other devices.
Software	Licensing and service cost of OS and application software.
Network	Network bandwidth costs charged by ISPs
Admin	Labor costs for system administration
Facility	Floor space, water, etc.
Energy	Electricity

TABLE II
COST CATEGORIES FOR CURRENT NON-CLOUD IT ENVIRONMENT

Cost estimation for the target cloud environment is relatively straightforward. The provider should already have the prices set for the compute resources in its cloud, such as VMs with different CPU and memory configurations, storage, network bandwidth, etc. The other costs, such as administration, facility and energy costs, are already factored into these prices. Therefore, the only estimation needed is to map out equivalent cloud resources that can support the same workload running on the current IT infrastructure. This is done in the following steps. First, we identify all servers in the current environment that need to be replaced by VMs in cloud. Then, we compute for each existing server the performance index using server performance benchmarks, such as [9], [10]. Last, for each existing server, we adjust its performance index with observed utilization percentages, and map it to an equivalent VM configuration in the cloud environment (the performance index for the set of VM configurations in the cloud offerings should be predefined by the provider).

Another one-time cost for cloud environment we need to consider is the cost incurred by workload migration. This includes the labor costs (e.g., image conversion, server decommissioning, VM and application set up, etc.), as well as the loss of revenue or productivity due to any disruption caused by migrations. Note that here, the migration schedule, namely when and which existing servers will be migrated, is also a big contributing factor to migration costs. Here, the model needs to take into account the application-level dependencies among existing servers, when deciding the sets of servers that need to be migrated together.

For cost estimation in both non-cloud and cloud environments, we also project IT environment growth rate and technology refresh rate, based on either input from customer or industry averages. This will help us map out the number of servers and equipments to be retired, purchased, and maintained in the current environment, or the scaling speed in the cloud environment, in any future time frame. After all the estimation steps described above, we can then compare the cumulative cost of current IT environment with that of the cloud environment, for a customer-specified time frame. Fig. 4 shows a sample cost comparison we conducted in a real cloud engagement. As shown in the figure, the initial cost of cloud environment is slightly higher at the beginning of migration. However, in merely 6 months, the cost of continuing current IT environment will catch up. By the end of year three, the cost of using cloud will be only about 2/3 of that of continuing the current IT environment.

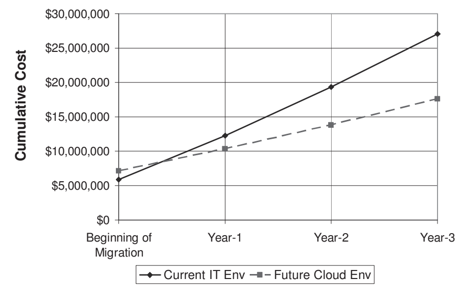

Fig. 4. A sample cost comparison between current IT and future Cloud environments

IV. CONCLUSION

In this paper, we proposed a method for conducting cloud ROI analysis using automatically discovered IT environment data. This proposed method is ideally suited for cloud provider to engage potential customers and to demonstrate the potential savings that can be achieved by adopting cloud services. To address the data availability challenge in this problem, we developed an iterative IT discovery and ROI analytics framework. This allows provider to conduct a wide range of ROI analysis at different stages of customer engagement, while continually refining the analysis with additional IT environment data being discovered. We demonstrated the feasibility and effectiveness of our method through real examples and case studies.

REFERENCES

[1] K. Beaty, V. Naik, and C.-S. Perng, "Economics of cloud computing for enterprise it," *IBM Research Journal*, Dec. 2011.

[2] M. A. Vouk, "Cloud computing – issues, research and implementations," *Journal of Computing and Information Technology*, vol. 4, pp. 235–246, 2008.

[3] M. Bohm, G. Koleva, S. Leimeister, C. Riedl, and H. Krcmar, "Towards a generic value network for cloud computing," *Lecture Notes in Computer Science*, vol. 6296, pp. 129–140, 2010.

[4] C.-S. Perng, R. Chang, T. Tao, E. So, M. Choi, and H. Shaikh, "Universal economic analysis for it transformation," in *NOMS*, 2012.

[5] "IBM Tivoli Application Dependency Discovery Manager (TADDM)." [Online]. Available: http://www.ibm.com/software/tivoli/products/taddm

[6] A. Keller and S. Subramanian, "Best practices for deploying a cmdb in large-scale environments," in *Proc. IFIP/IEEE International Symposium on Integrated Network Management (IM'09)*, Jun. 2009, pp. 732–745.

[7] "Discovery and the configuration management database (CMDB)," White Paper, CA Service Management, Oct. 2008.

[8] "IBM Tivoli common data model: Guide to best practices," Red Book, IBM, 2008.

[9] (2012) IDC Qualification Performance Indicator. [Online]. Available: http://www.idc.com/QPI/index.jsp

[10] (2012) Idea International: Relative Server Performance. [Online]. Available: http://www.ideasinternational.com/IT-Buyers/Server-Performance