

Enterprise-Scale Cloud Migration Orchestrator

Jinho Hwang Yun-Wu Huang Maja Vukovic Nikos Anerousis
IBM T.J. Watson Research Center

ABSTRACT

With the promise of low-cost access to flexible and elastic resources, enterprises are increasingly migrating their existing workloads into the Cloud. Yet, the heterogeneity of the workloads and existing configuration of legacy IT infrastructure make it challenging to enable a one-click, seamless migration process. There are multiple tools available for migrating servers based on their existing configurations and multiple ways of dealing with data synchronization (post migration). In this paper, we present a Cloud Migration Orchestrator (CMO), based on business process management (BPM) approach to provide a systematic framework to automate and coordinate migration activities. CMO coordinates the process of migration, starting from discovery, provisioning, network configuration, execution of migration, cutover and validation. CMO integrates multiple migration technologies, to support different migration scenarios. We present and discuss our results from a preliminary deployment of CMO to migrate 25 VMware instances and discuss how this approach improves the effectiveness of migration, and seamlessly coordinates activities required to be executed.

I. INTRODUCTION

With the promise of reduced IT costs, growth of flexible resources in Cloud, choice of deployment infrastructure and opportunities for next generation development of Cloud based resources enterprises are increasingly moving their existing workloads to Cloud. There are three key factors that make the migration to Cloud process challenging: a) heterogeneity, b) dynamicity and c) business criticality of the existing workloads. Regarding the heterogeneity, it is not uncommon for an enterprise to have 1000s of single-tier applications spread across existing data centers on Windows or Linux instances. In addition, the same enterprise may have 1000s of existing VMs that are part of existing internal Cloud (e.g. managed by VMware hypervisors). Moreover, today's applications are different - often an enterprise may have multiple applications operating on VMs within the same physical server, distributed computing applications, clusters of multiple server is compute resource pool, and ultimately a set of XaaS (“*everything-as-a-service*”) solutions. All of these advances in software applications models are affecting the complexity infrastructure and impacting the migration process.

Migration process needs to be carefully planned, as to ensure that any data that is captured and stored during the migration window is synchronized accordingly from source to target instances. Finally, speaking of business criticality of migration workloads—often the migration process may need to take care of applications that process highly important data, or prepare, for example, quarterly reports. The migration process needs to be resilient and ensure that the source instances

that host the workload remain intact until the migration is completed. Given the different setup of source workloads (e.g. VMware based, etc.) there are multiple ways of migrating the workloads (and different tools available), such as VMware Site Recovery Manager (SRM). As a result, there is a demand for a systematic, standardized approach for migration to Cloud. There are two key capabilities that are needed as part of this effort: a) ability to orchestrate at large-scale (in 1000s of servers) and b) optimize application deployment in the Cloud.

In this paper, we present a Cloud Migration Orchestrator (CMO), based on Business Process Management BPM [1] technology, to enable coordination of large-scale migration efforts. The approach generalizes and standardizes roles and activities in migration process. It integrates services through APIs, and engages humans experts via e-mail to trigger notifications for pending tasks. We discuss how CMO supports IT address retention in target environment, and thereby reduces human errors and cost of post-configuration. CMO embeds multiple migration technologies, such as Site Recovery Manager (SRM) and VMware Converter. We present and discuss results from experimental setup that migrated 25VMs using CMO.

The remainder of the paper is structured as follows. Next section reviews the related research efforts in migration orchestration. Section III provides an overview of the end-to-end migration process, and highlights its complexity and the need for more standardized approach to support large scale migration. Section IV introduces the BPM technology concepts, including process center and process server components. Section V presents our approach to employing BPM for migration process. Section VI presents experimental setup and results from migration of 25VMs. Finally, Section VII concludes and outlines our future work.

II. BACKGROUND AND RELATED WORK

Enterprise migration is a complex process that involves a set of human-driven and automated activities. As shown in Figure 1, the process starts with source environment discovery [2, 3]. This includes discovery of applications, middleware, servers and networking elements [4]. Besides the raw data collection, often analysis follows this step in order to identify misconfigurations [5] and optimize the configuration for the target environment. Whilst raw data collection is often done in an automated fashion, using scripts or other discovery tooling, analysis and optimization phase includes experts who finalize, plan and approve the proposed configurations and their scheduling [6]. This may involve templates and graph-based matching techniques [7]. Once the servers are grouped and scheduled for migration, the next step involves provisioning of the target environment. This includes several sub steps, starting from creating an account in Cloud environment, provisioning

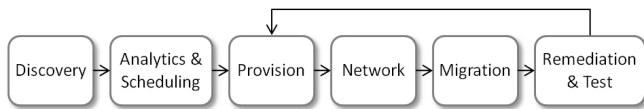


Fig. 1. The Six Stages of End-to-End Migration Process

the necessary network devices, including routers, firewalls, VPN products and configuring them. For example, if owners of the source environments wish to retain the IP addresses in the target side, additional secure WAN connection may need to be established for that purpose. Network configuration is a labor-intensive task, whilst many actions may be automated, some still rely on assistance from networking teams who manage Cloud environment. Once the environment is setup, the next step is to trigger the migration tools and monitor its executing. Finally, once migration completes, a post-configuration activity is performed to finalize and ensure that the target environment is functioning as expected.

Liu et al. [8] propose COPE (Cloud Orchestration Policy Engine), a distributed platform that allows cloud providers to perform declarative automated cloud resource orchestration. In COPE, cloud providers specify system-wide constraints and goals using COPElog, a declarative policy language geared towards specifying distributed constraint optimizations. Menzel and Ranjan [9] present a CloudGenius framework, which automates the decision-making process based on a model, factors and QoS specifically for Web server migration to the Cloud. In contrast, our system, CMO is a multi-tenant solution that employs a business process management technology, based on IBM BPM product [1], which helps us schedule, orchestrate and monitor a set of automated and human-driven activities. CMO integrates external tools and services [10] through REST-based APIs, sends notifications to users via e-mail, and continuously tracks the progress of the execution.

III. END-TO-END ORCHESTRATED MIGRATION PROCESS

Migrating enterprise data centers to cloud is a highly complex endeavor which may span multiple years while requiring a considerable amount of human and computing resources. We employ a migration framework adapted from [6] with some variations. As depicted in Figure 1, the migration framework divides the complex end-to-end process into the following six stages:

- Discovery: collect workload data.
- Analytics and Scheduling: assess and profile workloads to determine if, when, and how they are migrated.
- Provision: create and configure the target site.
- Network: establish secure network between source and target sites.
- Migration: perform actual migration of workloads.
- Remediation and Test: reconfigure and test workloads to ensure that they are running in the target site.

A. The State-of-the-Art Approach

In a typical effort in cloud migration for enterprise data centers, a team is established and divided into subgroups to handle the respective stages. Each stage is equipped with some automated tools to improve productivity. There maybe a centralized data and tool repository and it is up to group members to access the repository to enter or query data

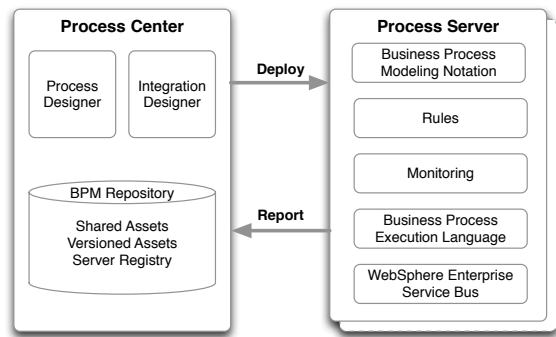


Fig. 2. BPM Architecture

and to execute tools while performing their respective tasks. Members inside or outside the same group communicates through emails. It is the team leader's responsibility to conduct periodical meetings and monitors the overall status.

We believe the current approach has the following shortcomings:

- Hand-off inefficiency. The tasks handing off is done by periodical meetings and members sending emails or making phone calls, which may incur unnecessary time gaps.
- Lack of tool integration. Tools are operated individually and not integrated into a cohesive workflow.
- No single-point global progress monitoring. Without it, exception handling becomes very costly.

B. End-to-End Orchestration

We propose a migration framework based on an end-to-end orchestration of the entire six-stage process as shown in Figure 1. In particular, we specify the following features in our proposed end-to-end orchestration approach:

- Team consists of members that each plays a specific role. For example, a member that is a network expertise handles network related tasks.
- System integrates automated tasks while minimizing human-tools interfacing. The automated tasks include:
 - Infrastructure integration such as API calls to request target cloud or hypervisor services.
 - Tool integration such as API calls to operate tools.
- Supports multiple migration tools to handle different types of workloads.
- System handles flows and hand-offs. System synchronizes parallel flows to improve efficiency. Human task hand-offs are done through notifications whereas auto task hand-offs are executed by the system.
- Provides single-point global status monitoring for more effective exception handling.
- All state and data are persisted by the system for long-running migration process.

We select IBM's Business Process Management (BPM) system to implement our end-to-end migration orchestration framework. We give the overview of BPM in Section IV and a case study of the proposed orchestrated migration approach based on BPM in Section V.

IV. BUSINESS PROCESS MANAGER OVERVIEW

Figure 2 illustrates a simplified BPM architecture divided into two parts: a process center and a process server that run

in WebSphere. Through these two processes, BPM provides a dev/ops model. That is, after developers develop an application process in the process center, the application process is deployed to the process server and monitored business performance. The business monitoring also provides information that helps users identify business problems, correct exceptions, and change processes to improve process efficiency and increase business competitiveness. Also, the process server reports on business operations to keep track of the running processes [1].

Measuring the business processes is an important debugging point. Key performance indicators (KPIs) are quantifiable measurements of the improvement in the performance of activities critical to the entire process execution. KPIs can be used to measure essential activities of business so that how these activities influence process results can be monitored. KPIs gather values such as average, maximum, minimum, sum, count (number of occurrences), or standard deviation. A metric is a holder for information, typically a business performance measurement, in a monitoring context. A metric can be used to define the calculation for a KPI, which measures performance against a business objective. Counters and stopwatches KPIs often depend on elapsed time, or on the number of occurrences of some situation or event [1].

Process Center: provides the repository that is used for BPM and the tools that are used to govern the lifecycle of business processes. In the process center, IBM Process Designer is an Eclipse-based tool that is used by business process developers. IBM Process Designer is available in all configurations of the product, and allows users to model and implement business processes as process applications. IBM Process Designer includes extra tools, Process Inspector and Process Optimizer. These tools are for interacting with processes running on the Process Center Server. IBM Integration Designer also is an Eclipse-based tool that is used by IT developers. IBM Integration Designer is used to author complex integrations and fully automated processes that support process applications designed in the IBM Process Designer. IBM Integration Designer has a fully integrated testing environment that uses test cases and test suites. IT developers can build reusable SOA (service oriented architecture) services, orchestrate those services, and access back-end systems. The Process Center provides the repository that is used for BPM and the tools that are used to govern the lifecycle of business processes [1].

Process Server: provides a single IBM Business Process Manager runtime environment that can support a range of business processes, service orchestration, and integration capabilities. Users can use Process Server to run processes as users build them. Process Server runs processes and services. Process Server handles a process's access to external applications, the Coaches (the graphical user interfaces for process participants), and the business logic. Every process component is loaded and run in Process Server at run time. The Process Server also provides a number of tools, for example, the Process Admin Console, that enable the management of the Process Server and deployed applications. Another tool is the Process Portal, which can be used to give individual users access to run and manage the tasks of the business processes. In addition, the Process Portal measures performance of business processes, teams, and individuals. The Process Portal delivers

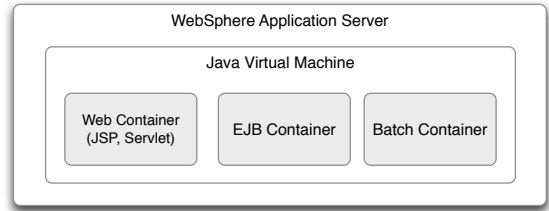


Fig. 3. Websphere Containers

an accurate overview that covers all aspects of the governed running business processes [1].

WebSphere: builds business-critical enterprise applications and solutions, and combine them with functions. The WebSphere Application Server family includes and supports a range of capabilities that helps users develop and serve business applications. We can use these capabilities to build, deploy, and manage dynamic websites and other more complex solutions productively and effectively. BPM runs the WebSphere as a core engine [11].

Figure 3 illustrates a simplified architecture how WebSphere runs a business process inside the JVM. Containers provide runtime support for applications. They are specialized codes in the application server that run specific types of applications. Containers can interact with other containers by sharing session management, security, and other attributes.

V. CASE STUDY: END-TO-END MIGRATION USING BPM

We implemented the Discovery, Network, and Migration stages of the proposed orchestrated migration approach described in Section III using IBM's Business Process Manager (BPM).

A. Orchestration Infrastructure

The CMO infrastructure is built in the target cloud, i.e., SoftLayer, which includes a BPM Process Center, a BPM Process Server, a REST-based API server that also serves as a mail server for notification purpose. We support both multi-tenants and single-tenant models. In the multi-tenants model, the global environment supports multiple migration orchestration instances serving multiple customers. In the single-tenant model, the orchestration infrastructure is replicated to the customer's target site, serving only the said customer.

B. Automation with BPM

The BPM supports integration with custom Java and JavaScript, which CMO utilizes to maximize automation. In particular, CMO includes custom Java libraries we develop to integrate with BPM, in order to:

- Provide REST API wrapper to access external services (through the API server).
- Configure target-side resources remotely using SSH.
- Integrate migration tools, e.g., VMWare's Site Recovery Manager.

The API server serves as a proxy to external services to provide encapsulation, flexibility, and integration of non-Java technologies such as Python to access SoftLayer API.



Fig. 4. Resources Provisioned in the Provision Stage of CMO

C. The Provision Stage

The Provision Stage of CMO is a nearly entirely automated stage. As depicted in Figure 4, there are five types of resources that are automatically provisioned and configured in the target site in SoftLayer. The vCenter here refers to VMWare’s vCenter server whereas the ESX here refers to VMWare’s ESX hosts which provide the virtualization layer hosting server VMs migrated from the customer data centers. We will describe the Gateway, VLAN and VLAN Bridge in more details in subsection V-D. In SoftLayer, while the Gateway (a Vyatta appliance) and ESX are provisioned as bare-metal hardware, the other resources can be provisioned either as hardware, cloud VMs, or even ESX-hosted VMs.

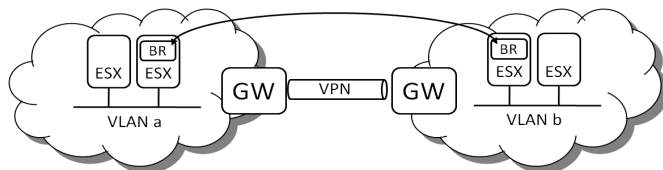
In any given migration run, there may be a need for 0 or more of these resources to be provisioned. For example, in the on-boarding step, one of each of these resources may need to be provisioned while the subsequent runs may only need additional ESX hosts and VLAN bridges depending on the migration plan.

For SoftLayer, the provisioning of these sources may take anywhere from less than an hour to more than a day depending on the type (e.g., hardware or cloud VM). We use a polling technique supported by BPM to check the provision status and notify users through emails when provisioning is complete or exceptions have occurred. Once provisioned, these resources are configured properly through our embedded custom Java code. The order shown in Figure 4 is significant as VLANs need to be associated to provisioned Gateway whereas other computing resources need to be on the provisioned VLANs.

D. The Network Setup Stage

The major difference between migration framework of CMO (Figure 1) and that presented in [6] is the Network Stage in our framework. In our experiences in working with large enterprise customers, one of the most importance issues in private cloud migration is security [12] in which network security is essential. The network security can mean that the target private cloud satisfies the customer’s security policy. But in the context of migration, it can also mean that during the long running process of migrating data centers to cloud, the connection between those migrated and yet to migrated needs to be secure. Thus our orchestration approach also establishes a secure WAN connection between the target cloud and the origin data center during the migration process.

The Secure WAN solution implemented by CMO is illustrated in Figure 5. In effect, it is a VPN connection between the target cloud and the on-premise data center. The Gateway provisioned in the Provision Stage is used to define a private cloud within SoftLayer. It means all communication in or out of the private cloud is routed through the Gateway. Furthermore, by establishing private tunnels that conform to the security requirement, the gateway safeguards the security of WAN connection.



On-Premise DC

Fig. 5. CMO VLAN Bridge for IP Retaining

During the CMO orchestration, a task is presented for the network lead to enter the network configuration data for the on-premise DC (data center). This data is then used by CMO to auto-configure the target-side gateway to establish the target side of the VPN connection. The target-side network details is then presented to the network lead to configure the on-premise side of the VPN. The CMO uses BPM to orchestrate these human and system tasks and use notifications, automation, and exception monitoring (e.g., deadline expiration triggers exceptions) to ensure the timely delivery of services. Another network feature of CMO is IP retaining, meaning the computing resources retain their IP addresses from on-premise DC even after they are migrated over to the target cloud.

The IP Retaining feature provides a great advantage in minimizing the potential complex remediation work incurred by IP change. The CMO achieves this by deploying a pair of network bridge VMs (i.e., Vyatta image) for pair-wise VLANs on target cloud and on-premise DC respectively. As shown in Figure 5, VLAN b is the receiving end of VLAN a on the target cloud. If any servers on VLAN b need to communicate with servers in the on-premise DC during or after migration, a pair of network bridges is created and configured such that servers on VLAN b can retain their original IP addresses.

The CMO automatically provisions and configures VLAN bridges on the target cloud while providing detailed step-by-step instructions for the network lead to create and configure the on-premise counterpart.

E. The Migration Stage

The CMO integrates third-party migration tools to perform server migration. Currently supported migration tools include VMWare Site Recovery Manager (SRM), vSphere Replication (VR), and VMWare Converter (Converter). Table I provides a contrast of the three migration tools. More performance discussions is given in Section VI.

The CMO provides limited automation for SRM, no automation for VR, and a fully automated task for Converter-based migration. The lack of automation on SRM and VR is because the available API is limited for SRM and nonexistent for VR. When CMO is unable to provide full automation for a migration tool, it interacts with the user with detailed instructions and screen shots that include the acquisition, installation, configuration and operation of the tool.

TABLE I. MIGRATION TOOLS

# Name	Replication	Type	Performance	Manageability	Cost
SRM	Array	V2V	High	High	High
VR	Network	V2V	High	Low	Low
Converter	Network	P2V,V2V	Low	Low	Low

F. Discussions

The CMO has fully implemented key stages of the proposed migration framework (shown in Figure 1) based on IBM’s Business Process Management (BPM) software. Using an orchestration engine such as BPM to model complex cloud migration has many advantages, such as the synchronization of automated and manual tasks, centralized state monitoring and managed exception handling, to name a few. There are issues to consider in the design and implementation of the orchestration model. For example:

- **Multi-tenancy vs. single-tenancy:** The multi-tenancy approach provides a low-cost model. While the single-tenancy approach imposes additional product license and deployment cost; it can better safeguard the customers’ potentially stringent security requirements. The CMO supports both approaches.
- **Multi-role orchestration vs. single-user self-service:** The multi-role approach designates tasks to only the right persons whereas the single-user approach delegates all tasks to one person. Different organizations may prefer one model over the other. The CMO supports both approaches.
- **Pure orchestration vs. integrated orchestration:** The pure orchestration approach relegates all non-process-related logic to external servers whereas the integrated approach allows direct integration of business logic into the process. The CMO is a hybrid model. It communicates to a REST API server for external services (e.g., Cloud service) while directly integrates Java services (e.g., SSH) for better performance.
- **Default UI framework vs. custom UI approach:** The default UI provided by the orchestration engine such as BPM allows for rapid composition but lacks sophisticated look and feel. The custom UI can be highly dynamic, but depends on fine-grained API support that may not be fully available from the orchestration engine such as BPM. The current CMO is based on BPM’s default UI framework.

VI. CASE STUDY EVALUATION

In this section we demonstrate a CMO implementation as a case study. We prototype a CMO on BPM with a VMware site recovery manager to realize an end-to-end migration process. We evaluate the following goals:

- demonstrate how much time it takes to provision various resources in SoftLayer (§VI-B),
- show data migration time over the secure WAN tunnel (§VI-C),
- identify time between shutting down the source and turning on the target (§VI-D),
- suggest the total time based on all previous experiments (§VI-E).

A. Experimental Setup

Figure 6 illustrates a testbed with CMO as a Software-as-a-Service (SaaS), placed in SoftLayer, offering to migrate from one data center on premise to SoftLayer as a target environment. CMO orchestrates the end-to-end process including provisioning target environments, setting up gateway network appliances (IPSec), copying source data into target machine,

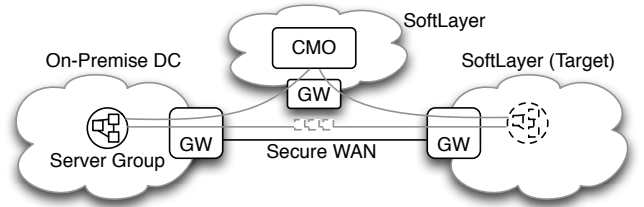


Fig. 6. Case Study Experiment Setup

and cutting over the operations from the source to the target as explained in Section VI-B.

B. Time for Resource Provisioning

Through repetitive experiments, we record how much time it takes to provision various computing resources through SoftLayer’s automated provisioning system (API). Table II shows each computing resource with its maximum provisioning time. Network appliances (bare-metal Vyatta gateway) takes the longest time due to the special sales approval process, hardware installation, and network setup in the router to let networks flow through the gateway. Provisioning bare-metal servers or virtual servers are fully automated so that it does not take too much of time. However, VLAN provisioning is not automated through API, so CMO create a ticket with all the details, which take more time when any unexpected events happen.

TABLE II. PROVISIONING TIME

Network Appliance	Bare-Metal	Virtual Server	VLAN
< 8 hours	< 4 hours	< 30 minutes	< 4 hours

C. Data Transfer Time via WAN

Once the target environment is set up, an image-based migration (block-level data copy) is triggered to start copying source data to the target environment. While we target to migrate a bulk of images at once, we also take into consideration various optimizations such as block deduplication, skipping empty data blocks, and transfer protocol optimization. To observe how VMware SRM performs, we transfer 25 images simultaneously. Table III shows the results of the migration when 25 VMs have only operating systems without any data and 25 VMs filled with the same data. We can easily observe that the data transfer optimization is phenomenal with VMware SRM. To see the relation between the quantity of data and transfer time, one big image is transferred with random data filled. As expected, the result shows the proportional relation between the transfer size and the transfer time.

TABLE III. DATA TRANSFER TIME

# VMs	Total Size (GB)	Transfer Size (GB)	Time
25 VMs (10 Linux and 15 Windows, no data filled)	750	171.75	3:07:00
25 VMs (10 Linux and 15 Windows, same data filled)	750	271.3	4:10:25
1 VM (data filled)	200	168.54	3:01:22

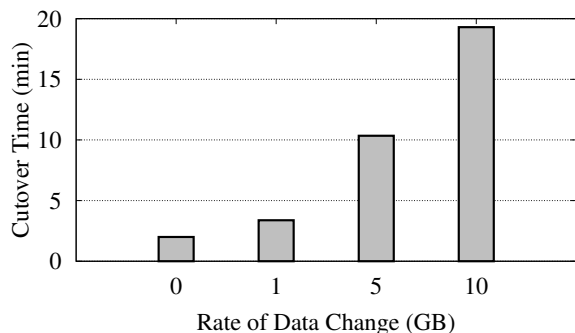


Fig. 7. Cutover Time vs. Rate of Data Change

D. Cutover Time Measurement

After the data transfer is done, we need to shut down the source machines, transfer the delta (changed data), and turn on the target machines (called cutover). The time between shutting down and turning on depends on how much data has been changed. Figure 7 illustrates the cutover time based on the rate of data change¹ (each machine).

E. End-to-End Process Time

To conclude the experiment, we show the estimated total time that takes for an end-to-end migration process using BPM based on our experience. The case study migrates 25 VMs in one ESXi host, and the target infrastructure includes one gateway (Vyatta), a pair of private and public VLANs, one vCenter, and one ESXi host—all the resources are provisioned during the migration process. Table IV summarizes the time. The user input usually takes 1 day after a user receives an email from CMO, and the steady-state means disconnect the network setup between the source and the target, and also between CMO and the target.

TABLE IV. SEGMENTED & TOTAL TIME FOR END-TO-END MIGRATION ORCHESTRATION

One Time		Repeat			Total Time
User Input	Provision	Transfer	Cutover	Steady-State	
24h	16.5h	4h	5m	30m	44h 5m

VII. CONCLUSION

This paper demonstrated a cloud migration orchestrator (CMO) built upon IBM’s Business Process Manager to coordinate an end-to-end migration process that envisions to include the entire migration steps. The CMO provides both automated processes through API or direct ssh access, and manual processes that can not be automated. Also, the backend servers such as an email server and cloud providers’ API servers help CMO automate/orchestrate the work flows. We showed our CMO prototype with VMware Site Recovery Manager (used as a migration tool) as a case study to measure segmented elapsed time to migrate 25 servers from on-premise data centers to cloud data centers. For the future work, we continue to expand the CMO to have source discovery and resource customization plugins and customer onboarding processes to maintain the customer data.

¹ The change has been made manually to reflect the exact size of data change.

ACKNOWLEDGMENTS

We thank Christopher C. Young, David Kohen, and John F. McKay Jr for valuable discussions, advices, and helping set up the evaluation environment. We also thank anonymous IM reviewers whose comments helped us improve the paper.

REFERENCES

- [1] IBM International Technical Support Organization. *Ibm business process manager version 8.0 production topologies*. IBM Redbooks, 2013.
- [2] Michael Nidd, Kun Bai, Jinho Hwang, Maja Vukovic, and Michael Tacci. Automated business application discovery. *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015.
- [3] Jill Jermyn, Jinho Hwang, Kun Bai, Maja Vukovic, Nikos Anerousis, and Salvatore Stolfo. Improving readiness for enterprise migration to the cloud. In *Proceedings of the Middleware Industry Track*, Industry papers, pages 5:1–5:7, New York, NY, USA, 2014. ACM.
- [4] Kun Bai, Niyu Ge, H. Jamjoom, Ea-Ee Jan, L. Renganarayana, and Xiaolan Zhang. What to discover before migrating to the cloud. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pages 320–327, May 2013.
- [5] Jiaqi Zhang, Lakshminarayanan Renganarayana, Xiaolan Zhang, Niyu Ge, Vasanth Bala, Tianyin Xu, and Yuanyuan Zhou. Encore: Exploiting system environment and correlation information for misconfiguration detection. In *Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS ’14*, pages 687–700, New York, NY, USA, 2014. ACM.
- [6] Mohammad Hajjat, Xin Sun, Yu-Wei Eric Sung, David Maltz, Sanjay Rao, Kunwadee Sripanidkulchai, and Mohit Tawarmalani. Cloudward bound: Planning for beneficial migration of enterprise applications to the cloud. *SIGCOMM Comput. Commun. Rev.*, 40(4):243–254, August 2010.
- [7] Birgit Pfitzmann and Nikolai Joukov. Migration to multi-image cloud templates. *2013 IEEE International Conference on Services Computing*, 0:80–87, 2011.
- [8] Changbin Liu, Boon Thau Loo, and Yun Mao. Declarative automated cloud resource orchestration. In *Proceedings of the 2Nd ACM Symposium on Cloud Computing, SOCC ’11*, pages 26:1–26:8, New York, NY, USA, 2011. ACM.
- [9] Michael Menzel and Rajiv Ranjan. Cloudgenius: Decision support for web server cloud migration. In *Proceedings of the 21st International Conference on World Wide Web, WWW ’12*, pages 979–988, New York, NY, USA, 2012. ACM.
- [10] Frank Leymann, Dieter Roller, and M-T Schmidt. Web services and business process management. *IBM systems Journal*, 41(2):198–211, 2002.
- [11] IBM International Technical Support Organization. *Web-sphere application server v8.5 concepts, planning, and design guide*. IBM Redbooks, 2013.
- [12] S. Al-Haj, E. Al-Shaer, and H.V. Ramasamy. Security-aware resource allocation in clouds. In *Services Computing (SCC), 2013 IEEE International Conference on*, pages 400–407, June 2013.