Virtual Machines in Transnational Digital Government: a Case Study

Andréa Matsunaga, Maurício Tsugawa and José A. B. Fortes
ACIS Laboratory, Dpt. of Electrical and Computer Eng., University of Florida
PO Box 116200, Gainesville, FL, 32611-6200, USA
1 (352) 392-4964
{ammatsun, tsugawa, fortes}@acis.ufl.edu

Introduction: Governments of different countries are increasingly expected to work together to address regional and global problems, provide disaster relief, satisfy international agreements, enable trade and movement of people across borders, and, in general, collaborate in achieving mutually-agreed goals. Information technology plays a key role in these transnational collaborations, particularly when country-specific information and IT resources are needed for each and every involved country to play their role in the collaborative efforts. In general, heterogeneity across IT systems is inevitable as it results from differences in economical and technical capabilities across countries, as well as from conceptual differences due to differences in agency missions and their regulatory context (which may, for example, specify what kind of software must be used) and differences in human IT resources. Integration may require use of new and/or existing hardware and/or software at different locations, processing and accessing data located in distinct agencies, and communication among many IT entities. In this context, heterogeneity can lead to several forms of incompatibilities among infrastructures, namely: (1) Hardware incompatibility: when machines do not have the expected architecture or capabilities to run machine-dependent software; (2) Software incompatibility: when needed software components require additional software that is either unavailable or conflicts with software used for local government functions; (3) Communication incompatibility: when multiple software components miss (a) interfaces for communication and (b) commonly understood protocols for communication; (4) Data incompatibility: when data maintained by software components use different organization, structure, semantic and natural language; and (5) Security and accessibility incompatibility: when integrating components with distinct or absent mechanisms for network security and access control.

This article describes the use of virtualization technologies [4] in a specific digital government project [5] to mask away hardware and software heterogeneity, complemented by the use of Web-services to address interoperation and security incompatibilities.

Case Study: The TDG system [5] is intended for information sharing, coordination and collaboration among government agencies in two countries: Belize and Dominican Republic. It integrates software components developed by several groups in different universities: (1) Distributed Query Processor (DQP) System: a distributed system that allows intra- and inter-government agencies to share information; (2) Local Information System (IS): manages and contains the actual information that is of interest for each agency and/or country; (3) Machine Translation System (MTS): multi-engine translation framework that translates sentences from a source language to a target language, in particular between English and Spanish; (4) Conversational System (CS): natural language parsing and dialogue management framework that extracts information from spontaneous speech; maintains dialogue context and history; and builds database queries; (5) Event Notification System (ENS): a distributed system that provides automatic distribution of selected important events to subscribing users.

The relationship among these components is shown in Figure 1. Each participant country's agency has its own information in a local information system (a), which is fully or partially shared with other participants through the DQP system (b). Since a query can be issued to the DQP system for example in an English speaking country (Belize) and some desired data may be in a Spanish speaking country (Dominican Republic), the DQP system makes requests as necessary to the MTS (c) to translate parts of the query or response. The users of this system can use either a web interface provided by the DQP system or the CS (d) which accepts queries in natural language and maintain a dialogue context. Finally, the DQP also interacts with the ENS (e) when some event of interest occurs in the DQP system, sending notifications to its subscribers.

x86-machines are assumed to be available since that was deemed to be the case for the prototype system. The DQP and ENS components were partially programmed anew in a platform independent language (Java) with their main tasks exposed as Web-services; however, these components were developed reusing some non-Java components deployed in Windows environment. On the other hand, MTS and CS were developed requiring Linux libraries. Therefore the DQP and ENS components cannot coexist in the same execution environment with the MTS and CS. VMware [6] is used to create virtual machines and eliminate software incompatibilities. VMware VMs were allocated to each OS and application so that they could run on the same physical machine. The local IS components cannot coexist in the same hardware for geographical and political reasons. The MTS component is network accessible, but because of its non-standard network protocol, it can not be easily integrated with other components and it lacks security features. The CS component does not offer any network capability and is stateful; thus, requiring special care in properly wrapping it as a Web-service. Each country's IS uses its own communication protocol, tied to the chosen relational database management system and has its own structure.

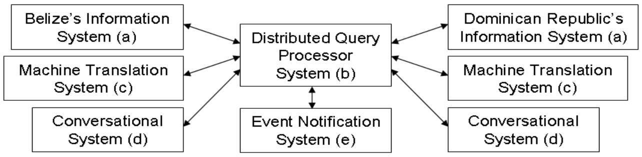
Belize's Information System (a)
Machine Translation System (c)
Conversational System (d)
Distributed Query Processor System (b)
Event Notification System (e)
Dominican Republic's Information System (a)
Machine Translation System (c)
Conversational System (d)

Figure 1. Relationship between the main components.

The integrated system prototype was deployed successfully using VMs with remote access and active-remote display to all university collaborators through In-VIGO system [1]. This setup made it possible for different component developers to work independently and collaborate during the integration process despite their geographical distribution and relatively poor network bandwidths to Belize and Dominican Republic. The possibility of easily changing hardware characteristics is another good feature of VMs used during system tests. Moreover, the system could be easily replicated in Belize and Dominican Republic since virtualization technology enable whole machines to be cloned, which eliminates software reinstallation process. This feature has the potential of greatly facilitating field deployment of the system.

The legacy components were manually wrapped as Web-services, and required a Web-service expert to understand the CS and the MTS components to create the proper wrapper. Once these systems became network-accessible, it was critical to provide security in this communication by reusing standard Web-services' network security protocols. The communication differences in the local IS were solved by using appropriate JDBC technology-enabled drivers, the data semantic differences (in particular the use of different natural languages) are solved including procedures during the distributed query processing that transforms the data using the MTS, and the data structural differences are solved using the DQP system [5], which facilitates the manual creation of a global schema with mappings among data entities and attributes based on local schemas exported with some meta-information; and issues distributed retrieval queries mediating the schema differences based on the global schema. Figure 2 presents the prototype system utilizing the presented solutions.

The overhead of using VMs depends on the system and workload characteristics [2], which is high when an application requires frequent system and I/O activity. Since IS runs without virtualization (it is part of government IT infrastructure) and DQP and ENS have moderated I/O activity, CS and MTS are the critical components. Currently, MTS makes use of a larger training database than CS, being the worst case scenario when running in VMs. Execution time measurements have been taken running MTS in a VM (VMware Workstation 4.5.2) and in the host physical machine (a dual Xeon 2.4 GHz with 2.5GB of memory). All conditions are favorable to the physical machine: larger memory, more physical and logical processors and direct access to disk. In spite of all disadvantages, the virtualization overhead observed was below 8%, which can be considered low. In the near future, hardware support for virtualization software (e.g., Intel Vanderpool technology [3]) is expected to reduce the performance gap between virtual and physical machines.

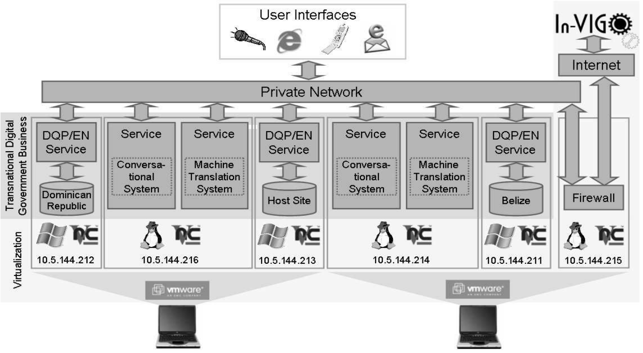

Figure 2. Virtualization eases integration of software components with incompatible operating system requirements in a single hardware and Web services eases integration of communication protocols.

Conclusion: This paper provided a rationale and an exemplary case study for the role of virtualization technologies for the elimination of integrations and interoperation barriers faced in building TDG IT systems. Public and commercial offerings of these technologies along with emerging hardware support in mainstream architectures will turn virtual machines into pervasive capabilities of computers in the near future. We thus believe that the use of virtualization technologies along with other technologies such as Web-services, as put forward in this paper, should be further explored and fully leveraged to address the technical challenges of TDG.

Acknowledgements: This research is funded in part by NSF awards EIA-0107686 and EIA-0131886. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

References:

[1] S. Adabala, et. al., From Virtualized Resources to Virtual Computing Grids: The In-VIGO System. Future Generation Computer Systems, to appear Vol. 21, No.6, April, 2005.

[2] Figueiredo, R. J., Dinda, P., and Fortes. J. A Case for Grid Computing on Virtual Machines. In Proceedings of 23rd International Conference on Distributed Computing Systems (ICDCS), May 19 - 22, 2003, 550-559.

[3] Intel. Intel Vanderpool Technology for IA-32 Processors (VT-x) Preliminary Specification. Intel, January, 2005.

[4] Smith, J. E. and Nair, R. Virtual Machine Architectures, Implementations and Applications. To be published: Morgan Kaufmann Publishers, 2005.

[5] Su, S., et. al., Transnational Information Sharing, Event Notification, Rule Enforcement and Process Coordination. International Journal of Electronic Government Research (IJEGR), Vol. 1, No. 2, April-June, 2005.

[6] VMware, Inc. Introducing VMware Virtual Platform. Technical white paper, February 1999.