

**CLOUD COMPUTING FOR SMALL NONPROFITS:
AFFORDABLE, LEAPFROGGING TECHNOLOGY**

Ted J. Strickland, Jr., University of Louisville, ted.strickland@louisville.edu

Jeff Trespalacios, University of Louisville, jeff.trespalacios@gmail.com

Ben Whatton, University of Louisville, ben.whatton@gmail.com

ABSTRACT

Cloud computing may become the mechanism for organizations that are struggling to provide adequate IT capabilities to leapfrog to more effective and affordable IT infrastructures and software solutions. For small nonprofit organizations, critical information systems, such as those that monitor delivery of programs and services, may be migrated from spreadsheets and desktop databases to integrated solutions hosted in the cloud. This paper proposes a win-win scenario for including cloud computing in information systems programs, where small nonprofit organizations may acquire custom solutions for monitoring program and services delivery, and where information systems students may gain hands-on experience analyzing, designing, and deploying cloud computing solutions for these organizations.

Keywords: SaaS, education, technological innovation.

INTRODUCTION

Within three years, cloud computing has emerged as a dominant topic of conversation among IT professionals. Forecasts of its efficiencies and cost savings in trade journals [29, 11, 18, 13] and industry white papers [37, 14, 1, 17, 20] have created a perception that cloud computing may be the IT industry’s next panacea. While practitioners and vendors acknowledge that it is not, cloud computing has become attractive for small to medium-sized enterprises (SME) [1]. For many SMEs, where IT is

not funded sufficiently, the IT staff is overworked, and assets are invested in out-of-date hardware and software [11], the promise of lower IT costs is compelling.

Cloud computing provides on-demand, pay-as-you-go, Internet-enabled IT services. For example, the IT director of a typical SME may opt to purchase computing time on a remote server in Amazon’s cloud, foregoing the decision to upgrade the current server. Because he only pays for computing resources used (often on an hourly basis), he may discover that cloud computing is less expensive than acquiring and managing a new server. In the process, he may realize other financial benefits as he exchanges capital investment expenses for operational expenses [1, 17, 11]. Fundamentally, cloud computing represents a change in thinking from buy-and-own to pay-as-you-go [1].

Cloud computing may be characterized in terms of its three major service models, Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS) [11, 37], as shown in Table 1. Service models may be combined behind the scenes in ways that are transparent to the cloud computing customer. For instance, Salesforce’s CRM executes on its Force.com platform. In this case, the user of SaaS does not need to be aware of the PaaS layer. For the example above, the SME IT director only needs IaaS to replace the computing capabilities of his aging server; the other two service models are not relevant to his current decision.

Table 1. Cloud Computing Service Offerings

Service Model	Characteristics	Example
Infrastructure-as-a-Service (IaaS)	Access to servers, data storage, and network bandwidth	Amazon S3 [5] and Amazon EC2 [4]
Platform-as-a-Service (PaaS)	Access to an externally hosted and managed platform for new software development	Force. com [33]
Software-as-a-Service (SaaS)	Access to software applications executing on externally hosted infrastructure	Google Apps [15] and Salesforce CRM [31]

In some respects, SaaS may be viewed as an extension of previous computing models, such as time sharing [9] and application service provider (ASP) [3]. During the last five years, developments within important enabling technologies, including virtualization software [13, 12], high speed networks and open standards [17], and distributed computing, service-oriented architectures, and client-side technologies [11], have made cloud computing viable. Additionally, the current tight economic conditions are conducive for its consideration by SMEs [9].

Cloud computing is not without its drawbacks. Executing software and placing organizational data on servers beyond the IT director's line of sight raises several concerns, including security risks [27], privacy [18], legal compliance and system integration [17], and performance and reliability [11]. Although progress is expected in several of these areas [24, 25, 28, 42], prospective adopters of cloud computing are cautioned to balance these concerns against anticipated productivity gains and cost savings [26].

Cloud Computing and Small Nonprofit Organizations

In some respects, small-sized nonprofit organizations are ideal candidates for cloud computing, as they face many of the same IT pressures of SMEs. A qualitative study of IT practices among small nonprofits (in progress) suggests that many of them struggle with one or more elements of IT planning, IT operations and support, and IT use. The profile of a typical small nonprofit, based on approximately 30 organizations examined to date, has the following characteristics:

- Reliance on office automation tools, especially spreadsheets, for data storage, manipulation, and reporting. Problems associated with multiple copies of data are common.
- Software tools that do not integrate, or that have not been configured for integration, create "islands of automation". Sharing of data among tools is difficult or impossible. Spreadsheet export/import is often the preferred form of data transport among these tools.
- Processes for services and/or program delivery are often unique to each organization. Software tools to monitor delivery of services and/or programs are used infrequently, contributing to the use of spreadsheets and personal databases (e.g., Microsoft Access) as the primary means of monitoring client engagement.

In addition, nonprofits are under pressure from funders to report outcomes of service and/or program effectiveness. The responses of small nonprofits are varied, based largely on ad-hoc approaches to outcomes measurement. Occasionally, service output (e.g., number of clients served) is confused with service outcome (e.g., degree of client improvement). Standardized outcome measures, such as guidelines provided by The Center for What Works and The Urban Institute [40], provide insight regarding performance measures to monitor. However, they are silent when it comes to where and how to store these measures.

For small nonprofits, the "where and how" questions of data storage are not answered easily. Despite decreasing costs of hardware, many organizations do not plan for hardware replacements on a regular schedule, relying on donated computers from the private sector. They do not budget adequately for IT expenses, including IT support. The need for a centralized server to host an outcomes reporting software package may stretch limited resources even further. Should the software and server be obtained, it is unlikely that adequate support may be provided, resulting in the software package becoming unstable or unusable over time.

For the small nonprofit struggling to provide and support an adequate IT infrastructure, cloud computing represents an opportunity to leapfrog technology. In this sense, small nonprofits may gain the benefits of cloud computing anticipated for governments of developing nations [10] and non-governmental organizations of rural developing countries [39]. These opportunities may span several areas of IT use: data storage, e-mail hosting, office productivity and collaboration tools, and application areas, such as constituent management and client management. Examples of these opportunities are summarized in Table 2.

In addition to lower costs, the benefits to nonprofit staff members may include 24/7 access to organizational data and software tools, solutions that conform to industry standards (e.g., CRM), and a richer environment for staff collaboration. A major advantage of cloud computing is the hosting component of the relationship – servers and software elements are managed by the cloud partner. Software solutions that only require a web browser and Internet access reduce the cost/support barriers for small nonprofits. The future of computing for small nonprofits may involve moving the data center to the cloud, and using laptops and mobile devices instead of desktop computers to access data and solutions [7].

Table 2. Leapfrog Technology Opportunities for Small Nonprofits

Current IT Practice: Move from	Cloud Computing Practice: Leapfrog to
Data storage centralized on a local server; e.g., the office LAN	Data stored on a virtual server in the cloud, accessible anytime, anywhere
e-mail server hosted locally; e.g., Microsoft Exchange	Google mail: small nonprofits are eligible for free accounts, where e-mail addresses contain the organization's domain name
Office productivity software installed on desktop; e.g., Microsoft Office	Google Apps: collaborative tools for word processing, spreadsheets, presentations, and calendaring[15]
Contact/constituent management using Spreadsheets and desktop databases	Customer Relationship Management (CRM) software; e.g., Salesforce [31]
Client management using spreadsheets, desktop databases, and manual forms	On-demand, custom solutions (i.e., SaaS) for client management and outcomes reporting

Cloud Computing and IS Students

For IS educators, the rapid emergence of cloud computing elevates a fundamental question often asked of IS programs – what form and how much exposure should IS students have to leading technologies such as cloud computing? Many IS programs look to professional organizations such as ACM and AIS for curricula recommendations. ACM has updated its model IS curricula periodically to keep pace with changes in information technology; i.e., IS '97 and IS 2002 [2]. In 2005, The Joint Task Force for Computing Curricula identified curriculum guidelines for each of five sub-disciplines of computing, including Information Systems [35]. However, significant gaps between practitioner expectations of graduates and the knowledge areas listed in The Joint Task Force's report were identified and questioned [6]. SaaS was listed as one of the areas where the IS curriculum recommendation lagged practitioner expectations. The draft of the IS 2010 curriculum guidelines appears to address some of these deficiencies, especially its emphasis on cloud computing and SaaS [2].

A cursory review of ACM conference publications (i.e., SIGCSE and SIGITE) and the Journal of Information Systems Education provided examples of how cloud computing concepts have been included in (or how they may be added to) computing courses. The examples focused on one or more of cloud computing's service models or examined its enabling technologies: (a) using Teredata's products via SaaS in database courses [21]; (b) testing Amazon's cloud as the infrastructure layer (i.e., IaaS) for an

undergraduate database course [19]; (c) creating computer lab environments where students may experiment with the virtualization technologies [23, 8], parallel storage systems [16], and large data sets [22]; and (d) identifying new programming abstractions for cloud-based software development [36].

In the spirit of exploring how cloud computing may be included in its current courses, the CIS program at the University of Louisville has introduced SaaS concepts into the senior-level systems development course. In keeping with previous expectations for this course, CIS students have recently undertaken projects that address IT needs of nonprofit organizations [38]. Release of Salesforce's Nonprofit Starter Pack (NPSP) [34] in December 2008 provided the impetus to address unique needs of local nonprofits using a common platform as well as an opportunity to expose CIS seniors to software development using SaaS.

**NONPROFIT INFORMATION SYSTEMS
USING SALESFORCE'S NPSP**

The NPSP is similar to Salesforce's popular CRM product, and it includes features to support the activities of the typical nonprofit organization: contact/constituent management, donor and campaign management, grouping of contacts into households, and standardized reporting. Salesforce reported that donated and discounted licenses are used by more than 8,500 nonprofit organizations worldwide [34]. Its popularity was also noted in a recent NTEN survey of nonprofit software use, where it was ranked

as the third most common software package for donor management [30].

License terms make the NPSP attractive to nonprofits. Any organization that may demonstrate its nonprofit status (i.e., IRS determination letter for U.S. organizations) may request 10 donated licenses for no cost and no annual renewal fee. Additional licenses (beyond the initial 10) may be purchased at a substantial discount; e.g., \$360 per user per year [34]. Hosting of the NPSP in Salesforce’s cloud is included.

In addition to customizations for nonprofit use, the NPSP may be extended. New attributes may be added to standard objects, new objects may be added to collaborate with standard objects, and new applications may be created based on new custom objects and/or standard objects. In most cases, custom objects may be created using simple selections from menus and wizards; i.e., programming is not required. With these extensions,

it is possible to create new features or to build a separate application that runs in the cloud.

Nonprofits that adopt the NPSP have three options: (1) use its default configuration as an integrated information system to manage contacts, households, donors, and campaigns; (2) extend it to automate other internal processes and move to an integrated information system; or (3) do both. The second option is appealing to many nonprofits, especially those with a human/social services focus, as it offers automated support for service and/or program delivery to clients; i.e., creation of a client management system. In addition, the need for outcomes monitoring and reporting may be addressed in the design of the client engagement system. A simple model of client engagement may be used to emphasize the data needs of an outcomes reporting module, as illustrated in Figure 1. This model has been used as the basis for the design and implementation of the NPSP solutions described in the following two sections.

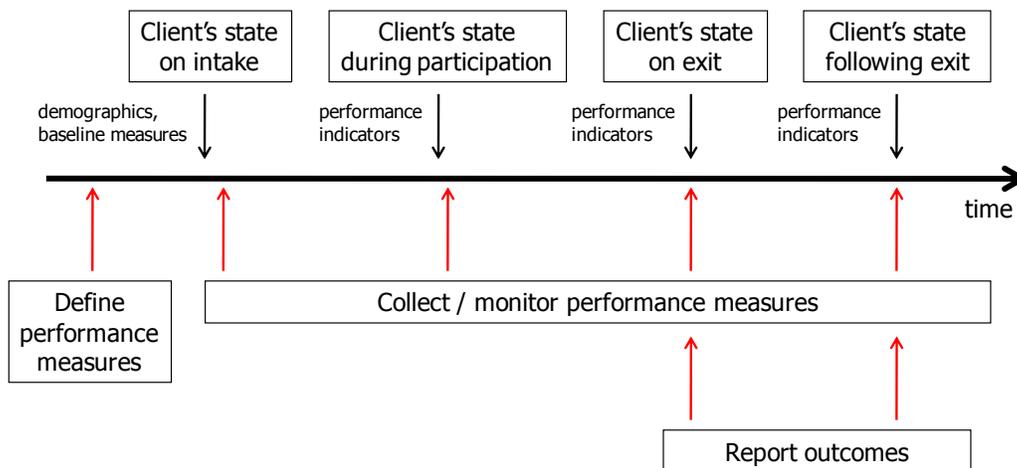


Figure 1. Model of Client Engagement for Outcomes Reporting

For outcomes reporting, the important elements of the model are:

- Performance measures are selected and defined before the service and/or program is initiated. Nonprofit executives may need guidance (see [40] for suggestions) in defining appropriate measures based on the nature of their services/programs.
- Besides demographic information that is captured as clients are enrolled, data elements used for performance measurement may need to be collected during the intake process. For

example, programs for youth education [41] may need data items describing school attendance, study habits, and school-related disciplinary issues as baseline values from which changes may be compared.

- In addition to collecting performance measures while clients are participating in the service and/or program, outcome data may need to be collected after clients complete or exit the service/program. It is typical to request that clients report data associated with outcome measures during six month, 12 month, or 18 month updates following participation. Selection

of outcome measures needs to consider the cost and convenience of collecting these measures once clients leave the program/service.

Introducing the NPSP into Course Projects

A pilot study to explore the capabilities of the NPSP was initiated in February 2009. The primary interest was extension of the NPSP, as its use as an element of the Common Solutions offering was important to the strategic direction of our nonprofit IT services project [38].

The pilot study assessed the feasibility of managing a nonprofit's internal clients. New data attributes and new screens were created using the NPSP's point-and-click interface; no custom objects were created and no custom code was written. Success of the pilot study increased our confidence that the NPSP may be used to automate business processes of nonprofits, especially if custom objects and code were used in the solutions. In Fall 2009 and Spring 2010, the NPSP was used in our senior-level CIS development course. Our experiences with this learning environment are described in this and the following section.

Initially, instructional resources were a concern. However, Salesforce provides a variety of support materials on its web site. Several resources were identified as appropriate for student learning:

- Individual developer account – Each student may join Salesforce's Developer Community [32]. Membership includes a free cloud computing account where the student may create new applications and custom objects. The user interfaces, administrative features, and reporting tools are similar to those of the NPSP. Students may explore design options using their individual accounts early in the semester, and then move to the NPSP once design decisions have matured and team-based development is appropriate.
- Reference books – Three of the books (PDF format) available for download from Salesforce's Developer site [32] have been used as texts and references:
 - Force.com Fundamentals: An Introduction to Custom Application Development in the Cloud – The "Try-it Out" format is conducive to self-paced tutorials for students. The major topics associated with developing a Salesforce application are covered in this book. Also, this text is accessible by students without technical backgrounds.

- The Developer's Guide to the Force.com Platform – This source was used as a reference to supplement the Fundamentals text. Topic coverage is similar to that of the Fundamentals book, but the explanations are deeper. This book is appropriate for information systems and other technically-oriented students.
- Force.com Cookbook: Code Samples and Best Practices – This source was used as reference to resolve technical "how to" questions within the Force.com platform. It includes code segments (Apex) to address specific situations, e.g., creating a button on a user interface screen, overriding the behavior of a standard button, previewing query results, etc. This book is intended for students familiar with programming and systems development.
- Web resources – The Developer Community [32] provides access to an array of resources to assist Salesforce developers. The sources that we explored in getting started with Salesforce include:
 - Guides and Cheatsheets – PDF quick references, covering topics such as Salesforce formulas, the Apex programming language, VisualForce programming, and Salesforce's web services.
 - Discussion boards – Used by students to request technical advice of the Salesforce developer community.
 - Code samples – Apex and VisualForce code to accompany the examples listed in the Developer book and the Cookbook.
 - Data Loader – A desktop application for importing data into the NPSP. This utility is helpful in moving the nonprofit's existing data into its NPSP account.

For complex systems development tasks, Salesforce provides tools to create a complete development environment. From their Developer edition accounts, students may download Salesforce's IDE toolset. This Eclipse tool installs on a local PC, providing access to code stored on the student's account in the cloud. Use of the IDE moves students beyond Salesforce's point-and-click user interface screens, controls, and wizards. Using the IDE, students may define custom behavior in Apex code. Students download Apex code from their Developer accounts, alter and test the code on the local computer, and then return it to their Developer edition accounts when the development task is finished.

Apex is Salesforce's on-demand programming language for building business applications. Its syntax resembles Java, but it is used in situations where stored procedures are used in other development environments; e.g., to customize the actions of triggers. Also, the Eclipse IDE may be used for custom development with the NPSP. Development within the Enterprise edition of Salesforce, including the NPSP, enforces Test Driven Development practices for Apex code of custom objects. Test cases must provide at least 75% coverage in order for new Apex code to be included. This requirement was not discovered until late in the Fall 2009 semester, delaying transfer of code from the students' Developer accounts to the organization's NPSP account.

Our limited experience has not allowed us to explore many of the instructional sources available on Salesforce's web sites. For example, more specialized coverage of Salesforce features, such as workflow rules, are available as a series of PDF texts. The site also includes on-demand video courses. Instructor-led courses are also offered at select locations in the United States.

Lessons Learned from Course Projects

Four nonprofit organizations participated in course projects during the 2009-2010 academic year. Each nonprofit requested automation of an internal business process associated with delivery of its programs or services. Two projects conducted during Spring 2010 also included the identification and design of data elements associated with outcomes reporting, as described above; see Figure 1.

From the organizations' perspectives, the information systems appear to be headed for successful deployments. Organizational representatives collaborated with their respective student teams to verify that system features, especially user interface layouts and report formats, satisfied requirements and met client expectations. Orientation sessions were conducted for one organization during Spring 2010, and system deployment activities are planned for the other three organizations during Summer 2010; i.e., one of Fall 2009 clients wished to delay system deployment until a less busy period. The nonprofit executives appear excited about the possibilities of their new systems. In some cases, initial concerns about cloud computing have been dispelled.

Three organizational representatives were asked their impressions of: (a) the development processes; and (b) the capabilities of their respective systems; i.e., the products. Each representative rated the process and product questions as having: (1) delivered less than expected, (2) delivered what was expected, or (3) delivered more than expected. All responses to both questions were positive, with two of three representatives reporting that students delivered more than expected (process and product) and the third representative indicating that students delivered about what he expected (process and product).

From an instructional perspective, students embraced the opportunity to develop a cloud computing solution. Most of the students acknowledged that they were not aware of the capabilities of tools such as the NPSP; many were surprised by the ease in which systems may be developed without writing code. When complex development issues were encountered, students accessed the texts and online resources to research options. In several cases, students used their Developer edition accounts to test design options prior to implementing them within the nonprofit's NPSP account. Team-based development also provided opportunities for students to develop interpersonal skills, especially those associated with client interaction and scope creep. A summary of each project and our observations regarding its most striking technical lessons and interpersonal lessons are listed in Table 3.

Students who contributed to the Spring 2010 projects described course activities that worked well (and those that did not) during an end of the semester debriefing session. A total of 35 comments were recorded by a student assistant who conducted the session; i.e., the instructor was not present. A qualitative review of the comments focused on three major areas where improvements were suggested:

1. Maintaining sufficient client interaction and managing project scope (26%);
2. Structuring the course so that development activities are efficient; i.e., comparable to professional practices (48%);
3. Enhancing course assignments and learning opportunities (26%).

Overall, the students' comments identified many of the tradeoffs involved when balancing the academic content of the course with the practical concerns of conducting information systems development within the constraints of a single semester.

Table 3. Observations from Course Projects Involving Use of the NPSP

Term / Project Objective	Technical Lessons	Interpersonal Lessons
Fall 2009: automate client records for a community home	Appropriate structure of the data model: NPSP allows quick and easy changes of data attributes and relationships among objects in evaluating design options.	Appeal of most valued features: clients embraced the NPSP and became project champions for its powerful reporting engine and easy-to-use user interface.
Fall 2009: track client involvement in programs of an arts organization	Benefits of extending existing objects: adding data attributes to NPSP standard objects and linking standard objects to custom objects are easy tasks with the NPSP; new capabilities may be added to existing capabilities quickly.	Managing client expectations: the client’s view of the total solution extended to systems/tools besides the NPSP; setting priorities for which ones could be completed within a semester-long course was challenging.
Spring 2010: provide client management for a young adult transition program	Use of prototyping: NPSP allows rapid development of user screens and system features to distill the system requirements; focused, frequent sessions were more productive than system demonstrations at major milestones.	Role of project champions: energetic client challenged the student team to put forth its best effort; staff member’s enthusiasm for the system reinforced the team’s commitment to deliver a high quality solution.
Spring 2010: replace the client management system of a workforce development center	Identifying and resolving tool constraints: NPSP relationship decisions may limit reporting options among custom objects. Design involved balancing data model normalization with reporting capabilities.	Dealing with differing client views of the system: managers’ and users’ viewpoints were resolved using focused and frequent prototype review sessions.

In addition, the projects encountered minor obstacles. Generation of test cases required to replicate code created in a Developer edition account within a NPSP account was not resolved during the course. The task was not complex, only tedious and time consuming. Also, a significant amount of time and effort was invested (by enrolled students, student assistants, and the instructor) in searching for tips and solutions on Salesforce’s site and within the NPSP; i.e., moving from a point-and-click Salesforce developer to moderately-skilled Salesforce developer was more arduous than expected.

Deploying and Supporting Student-based Solutions

In creating unique solutions for nonprofit organizations, our approach to system design preserves the default “out-of-the-box” capabilities of the NPSP. Its native features for donations, campaigns, contacts, and households have not been disturbed by the extensions added by student teams. The extensions involve custom objects, data attributes added to standard objects, and new relationships among custom and standard objects.

This approach allows the organization to use the custom solution for its unique need and the standard features of the NPSP for needs common to the nonprofit sector.

Also, system design restricts how the solution may be modified following deployment. Our nonprofit IT project was formed to develop and to deliver IT solutions that are beyond the capabilities of small nonprofits. Student teams take on the system support role, and only students assigned to the project are permitted to make changes to the solution when new needs arise. In this regard, nonprofit staff members are users of the solution, but they are not granted administrative privileges. This practice ensures that a staff member does not corrupt or destroy the solution inadvertently.

CONCLUSION

Emergence of cloud computing offers the opportunity for small nonprofit organizations to reexamine their approaches to how information systems may be used to support their missions. In addition to reducing costs, system deployment in the cloud reduces the

reliance on spreadsheets and desktop databases, and thus cloud computing may eliminate the “islands of automation” present among many nonprofit IT architectures. Placing important organizational data in the cloud allows wider access across the organization.

Our experiences with Salesforce’s NPSP indicate that a team of information systems students may develop a custom application to monitor and report nonprofit program/service delivery as a single semester course project. Client engagement is essential to the development process. Use the NPSP to demonstrate and revise the information system in an iterative manner has proven useful in building collaboration among nonprofit staff members and students. Salesforce offers the tools and online resources to cover a variety of development interests and specializations, depending on the demands of the problem or the interests of the instructor.

Including the NPSP in the information systems development course has produced a win-win scenario – the nonprofit leapfrogs to a new, more robust information system and students are exposed to SaaS and cloud computing.

REFERENCES

1. Accenture (2009). What the Enterprise Needs to Know About Cloud Computing, Accenture white paper, October 2009, retrieved on March 16, 2010.
2. ACM Curricula Recommendations (2010). <http://www.acm.org/education/curricula-recommendations>, retrieved on April 7, 2010.
3. Altaf, F., Schuff, D. (2010). Taking a Flexible Approach to ASPs, CACM, Vol. 53(2), pp. 139-143.
4. Amazon Elastic Compute Cloud, <http://aws.amazon.com/ec2/>, accessed on April 14, 2010.
5. Amazon Simple Storage Service, <http://aws.amazon.com/s3/>, accessed on April 14, 2010.
6. Andriole, S., Roberts, E. (2008). Technology Curriculum for the Early 21st Century, CACM, Vol. 51(7): pp. 27-32.
7. Asanovic, K., Bodik, R., Demmel, J., et al (2009). A View of the Parallel Computing Landscape, CACM, Vol. 52(10):56-59.
8. Brown (2009). Hadoop at Home: Large-Scale Computing at a Small College, SIGCSE '09, Chattanooga, TN, March 3-7, 2009, pp. 106-110.
9. Campbell-Kelly, M. (2009). Historical Reflections: The Rise, Fall, and Resurrection of Software as a Service, CACM, Vol. 52(5), pp. 28-30.
10. Cleverly, M. (2009). Emerging Markets: How ICT Advances Might Help Developing Nations, CACM, Vol. 52(9), pp. 30-32.
11. Creeger, M. (2009). CTO Roundtable: Cloud Computing, CACM, Vol. 52(8), pp. 50-56.
12. Creeger, M. (2008b). CTO Virtualization Roundtable, Part II, CACM, Vol. 51(12), pp. 43-49.
13. Creeger, M. (2008a). CTO Roundtable on Virtualization, CACM, Vol. 51(11), pp. 47-53.
14. Gartner (2010). Leading in Times of Transition: The 2010 CIO Agenda, Gartner Executive Programs white paper, January 2010, retrieved on March 18, 2010.
15. Google Apps for nonprofits, <http://www.google.com/apps/intl/en/nonprofit/index.html>, accessed on April 14, 2010.
16. Hacker, T., Springer, J. (2008). Meeting the Challenge: Curriculum Development for Parallel Data Systems, SIGITE '08, Cincinnati, OH, October 16-18, 2008, pp. 153-156.
17. Hartman, T., Beck, L. (2009). Defining the Business Value of Cloud Computing, Avande, Inc. Point of View white paper, retrieved on March 23, 2010.
18. Hayes, B. (2008). Cloud Computing, CACM, Vol. 51(7), pp. 9-11.
19. Holden, E., Kang, J., Bills, D., Ilyassov, M. (2009). Databases in the Cloud: A Work in Progress, SIGITE '09, Fairfax, VA, October 22-24, 2009, pp. 138-143.
20. IBM (2008). IBM Perspective on Cloud Computing: The Next Big Thing or Another Fad?, IBM white paper, retrieved on March 16, 2010.
21. Jukie, N., Gray, P. (2008). Teradata University Network: A No Cost Web-Portal for Teaching Database, Data Warehousing, and Data-Related Subjects, Journal of Information Systems Education, Vol. 19(4), pp. 395-402.
22. Kimball, A., Michels-Slettvet, S., Bisciglia, C. (2008). Cluster Computing for Web-Scale Data Processing, SIGCSE '08, Portland, OR, March 12-15, 2008, pp. 116-120.
23. Lunsford, D. (2009). Virtualization Technologies in Information Systems Education, Journal of Information Systems Education, Vol. 20(3), pp. 339-348.
24. Mowbray, M., Pearson, S. (2009). A Client-Based Privacy Manager for Cloud Computing, The Fourth International Conference on Communication System Software and Middleware, Dublin, Ireland, June 16-19, 2009, ACM SIGCOMM, article #5.

25. Pearson, S. (2009). Taking Account of Privacy when Designing Cloud Computing Services, ICSE Workshop on Software Engineering Challenges of Cloud Computing, Vancouver, May 23, 2009, IEEE Computer Society, pp. 44-52.
26. Preimesberger, C. (2010). Top 10 Mistakes Enterprises Can Make When Moving Data into the Cloud, <http://www.eweek.com/c/a/Cloud-Computing/Top-10-Mistakes-Enterprises-Can-Make-When-Moving-Data-into-the-Cloud-811577/?kc=EWKNLEDP03122010B>, March 4, 2010, retrieved on March 16, 2010.
27. Prince, B. (2010). Cloud Computing's 7 Deadliest Security Risks, <http://www.eweek.com/c/a/Security/Cloud-Computings-7-Deadliest-Security-Risks-345990/?kc=EWKNLEDP03122010A>, March 8, 2010, retrieved on March 16, 2010.
28. Ristenpart, T., Tromer, E., Shacham, H., Savage, S. (2009). Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds, 16th ACM Conference on Computer and Communications Security, Chicago, November 9-13, 2009, pp. 199-212.
29. Rosen, G. (2010). The Business of Clouds, ACM Crossroads, Vol. 16(3): pp. 26-28.
30. Ross, H. (2010). Your Voices: 2009 Data Ecosystem Survey, Nonprofit Technology Network, <http://nten.org>, Portland, OR.
31. Salesforce, <http://www.salesforce.com/>, accessed on April 14, 2010.
32. Salesforce developerforce, <http://developer.force.com/>, accessed on April 14, 2010.
33. Salesforce Force.com platform, <http://www.salesforce.com/platform/>, accessed on April 14, 2010.
34. Salesforce.com Foundation, <http://foundation.force.com/home>, accessed on April 14, 2010.
35. Shackelford, R., Cross, J., Davies, G., et al (2006). Computing Curricula 2005: The Overview Report, The Joint Task Force for Computing Curricula 2005, ACM, AIS, IEEE-CS.
36. Shroff, G., (2008). Dev 2.0: Model Driven Development in the Cloud, SIGSOFT 2008, Atlanta, November 9-15, 2008, pp. .????.
37. Somashekar, S. (2010). Opportunities for the Cloud in the Enterprise, CA white paper, January 2010, retrieved on March 16, 2010.
38. Strickland, T.J., Jr. (2008). Student-based IT for Nonprofits – An Alternative to Do-It-Yourself IT, *Issues in Information Systems*, Vol. IX(1):146-151.
39. Tapan, S. (2009). Engineering Rural Development, CACM, Vol. 52(1), pp. 54-63.
40. The Urban Institute Outcomes Indicators Project, <http://www.urban.org/center/cnp/projects/outcomeindicators.cfm>, accessed on April 7, 2010.
41. The Urban Institute Outcomes Indicators Project, Candidate Outcome Indicators: Youth Mentoring Program, http://www.urban.org/center/met/projects/upload/Youth_Mentoring.pdf, accessed on April 14, 2010.
42. Wei, J., Zhang, X., Ammons, G. (2009). Managing Security of Virtual Machine Images in a Cloud Environment, The ACM Cloud Computing Security Workshop, Chicago, November 13, 2009, pp. 91-96.