“Movin’ On Up – Planning for Moving an Academic Libraries’ Infrastructure to the Cloud”

Educause Live!
Thursday, April 30, 2015

Mark Dehmlow
Program Director, Library Information Technology
Hesburgh Libraries
University of Notre Dame
mark.dehmlow@nd.edu
Contributors

• Robert Minvielle, Manager Enterprise Systems Unit, University of Notre Dame Hesburgh Libraries
• Mike Chapple, Sr. Director IT Service Delivery, University of Notre Dame Office of Information Technology
Notre Dame Cloud First Strategy

Creating Notre Dame’s IT future

CLOUD FIRST

WE NEED WHAT THE CLOUD OFFERS

- Demand for IT services will continue INCREASING
- Funding and staffing remain CONSTANT
- We must find NEW WAYS TO EXCEL in delivering IT services

agility

scalability

sustainability

flexibility

resiliency

cost savings

TO MEET OUR GOAL OF IT SERVICE EXCELLENCE:

We will move 80% of Notre Dame IT services into the cloud by Dec 31, 2017

ND already has more than 100 IT services in the cloud.

Source: “Cloud First” marketing materials. Notre Dame OIT.
http://oit.nd.edu/cloud-first/
Cloud First Hosting Models

Software as a Service (SaaS)

Platform as a Service (PaaS)

Infrastructure as a Service (IaaS)

On-Premises Solutions

http://www.educause.edu/library/resources/cloud-strategy-higher-education-building-common-solution
Why Amazon Web Services For IaaS?

- Most advanced data centers in the world
- Proven pace of innovation
- Relentless focus on customer success

Source: “Cloud First” marketing materials. Notre Dame OIT.
http://oit.nd.edu/cloud-first/
IT Evolution Through Abstraction

Programming

- Machine Language
- Assembly
- Procedural
- Object Oriented

IT Infrastructure

- VMware
- Microsoft Azure
- Amazon Web Services
- Google Cloud Platform
Benefits

- Cost?
  - Financial
  - Human Resources
    - Repurpose
    - Best Practices Adoption

- Scalability
  - Elasticity

- Availability
  - On Demand

- Redundancy
  - Global Infrastructure
Local vs. Cloud Hosting

• Capital Expenditure vs. Pay as You Go

5-7 Year Cycle

- Purchase
- Deploy
- Maintain
- Decommission

Monthly/Yearly

- Network
  • Data Transfer Out

- Storage
  • Total Data Stored
  • Data Transfer Out

- Computing
  • CPU
  • Memory
The same data can lead to many different, but still valid, financial models. These models all show a *true cost* for a service.

<table>
<thead>
<tr>
<th>Viewpoint / Financial</th>
<th>University</th>
<th>CIO</th>
<th>Central IT</th>
<th>Department</th>
<th>Researcher</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equipment Replacement</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Capital Cost</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Space Cost</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Electricity</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operations</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disaster Recovery</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Opportunity Cost</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>Network Cost</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>...</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source: “Cloud First Financial Models/Value Proposition.”
Jim Jokl, Associate VP and Chief Enterprise Architect, University of Virginia
Benefits of Service/Hosting Assessment

- Comprehensive Service Inventory
- Re-assess value of services
  - Look at consolidation and decommissioning
- Assess and Apply Service Levels
  - Importance and Impact
  - Platinum, Gold, Silver
<table>
<thead>
<tr>
<th>Service</th>
<th>Attachments</th>
<th>Cloud Readiness</th>
<th>Notes</th>
<th>Depends on</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserves</td>
<td>x</td>
<td>Poor</td>
<td>Perl catalyst application primarily used by the security monitors but also</td>
<td></td>
</tr>
<tr>
<td>Factotum (factotumprod)</td>
<td>x</td>
<td>Poor</td>
<td>DLT may be working on a replacement; this was the first PURL service that</td>
<td></td>
</tr>
<tr>
<td>Intranet (perlprod)</td>
<td>x</td>
<td>Good</td>
<td>Extremely critical web service; central data repo; ties into LDAP / EDS and</td>
<td></td>
</tr>
<tr>
<td>Puri (perlprod)</td>
<td>x</td>
<td>Good</td>
<td>Primo uses this heavily to display UI record information</td>
<td></td>
</tr>
<tr>
<td>API (apiprod)</td>
<td></td>
<td>Medium</td>
<td>E-Journal locator; relies on SFX data; relies on Aleph;</td>
<td></td>
</tr>
<tr>
<td>Fresh Writing (fwprod)</td>
<td></td>
<td>Good</td>
<td>Runs the lobby kiosk</td>
<td></td>
</tr>
<tr>
<td>Discover (discoverprod)</td>
<td></td>
<td>Good</td>
<td>Current mobile site but we are switching to Kurogu</td>
<td></td>
</tr>
<tr>
<td>EJournal Locator (ejiprod)</td>
<td>x</td>
<td>Good</td>
<td>Workflow tool for the reformatting unit and cataloging</td>
<td></td>
</tr>
<tr>
<td>Kiosk (kioskprod)</td>
<td></td>
<td>Medium</td>
<td>Front end for the digital exhibits framework</td>
<td></td>
</tr>
<tr>
<td>Mobile website (mobileprod)</td>
<td>x</td>
<td>Medium</td>
<td>Processing and syndication of images; image server</td>
<td></td>
</tr>
<tr>
<td>Siphon (siphonprod)</td>
<td></td>
<td>Medium</td>
<td>Web asset server (JS, CSS, images)</td>
<td></td>
</tr>
<tr>
<td>Honeypot (honeypotprod)</td>
<td></td>
<td>Medium</td>
<td>Admin tool for digital exhibits framework</td>
<td></td>
</tr>
<tr>
<td>BeeHive (beehiveprod)</td>
<td></td>
<td>Medium</td>
<td>Currently deployed in AWS; uses Postgres; relies heavily on api server;</td>
<td></td>
</tr>
<tr>
<td>Asset server (assetprod)</td>
<td>x</td>
<td>Medium</td>
<td>ERM system used by selectors, cataloging and acquisitions;</td>
<td></td>
</tr>
<tr>
<td>Honeycomb (honeycombprod)</td>
<td>x</td>
<td>Medium</td>
<td>Data mining tool; relies on Aleph data; large db footprint;</td>
<td></td>
</tr>
<tr>
<td>Annex IMS</td>
<td></td>
<td>Good</td>
<td>Database finder back end; runs under Tomcat; Java app; relies on Lucene for</td>
<td></td>
</tr>
<tr>
<td>Coral (pphprod)</td>
<td></td>
<td>Poor</td>
<td>Springboard to a perl / mysql search engine that houses the Ambrosiana</td>
<td></td>
</tr>
<tr>
<td>DataMart (dmprod)</td>
<td></td>
<td>Good</td>
<td>Catholic biography database;</td>
<td></td>
</tr>
<tr>
<td>Database Finder</td>
<td></td>
<td>Good</td>
<td>Event registration system for the campus TEDx event</td>
<td></td>
</tr>
<tr>
<td>Italien (isamount)</td>
<td></td>
<td>Medium</td>
<td>Front end to the Metaval aggregated search engine (an Ex Libris product)</td>
<td></td>
</tr>
<tr>
<td>Biographies (biographiesprod)</td>
<td>x</td>
<td>Poor</td>
<td>Legacy electronic theses and dissertations app; Written in Perl;</td>
<td></td>
</tr>
<tr>
<td>TedX (tedexprod)</td>
<td></td>
<td>Poor</td>
<td>Legacy digital guides system; written in antiquated ruby / RoR; changing to</td>
<td></td>
</tr>
</tbody>
</table>
How to Determine Move to AWS?

- Is it a web service?
- Does it need reliability, scalability, and redundancy?
- How is it architected?
  - Smaller components interoperating are less expensive and better fit the Amazon model than monolithic forklifted applications.
- What is the projected cost?
Applications designed for load balancing, locking files, tables etc.

Architecture Cntd.

Scalability wrapped around AMI
Thresholds to grow and shrink
• What should be hosted by a vendor?
  – Software that they write and maintain.
  – If your vendor meets expected hosting requirements for performance (up time, throughput, etc.)
  – Look at history of support and up time.
AWS Billing Model

• Resources
## Resource Usage Monitoring - Ganglia

**Weekly Average RHEVM Node Resource Use (By Node) From 04/01/2015 To 04/08/2015**

<table>
<thead>
<tr>
<th>Node</th>
<th>Disk</th>
<th>Mem</th>
<th>CPU (User)</th>
<th>Net In</th>
<th>Net Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaron.library.nd.edu</td>
<td>18.75</td>
<td>56.98</td>
<td>0.46</td>
<td>1500.97</td>
<td>282.88</td>
</tr>
<tr>
<td>abiathar.library.nd.edu</td>
<td>1116.14/2396.30</td>
<td>51.01</td>
<td>CPU: 2.47</td>
<td>764.07</td>
<td>515.08</td>
</tr>
<tr>
<td>ahitub.library.nd.edu</td>
<td>8.08</td>
<td>53.21</td>
<td>0.38</td>
<td>557.38</td>
<td>184.47</td>
</tr>
<tr>
<td>alacartepprd-vm.library.nd.edu</td>
<td>5.40</td>
<td>19.01</td>
<td>0.77</td>
<td>18.50</td>
<td>5.58</td>
</tr>
<tr>
<td>alacarteprod-vm.library.nd.edu</td>
<td>6.41</td>
<td>19.01</td>
<td>1.54</td>
<td>16.57</td>
<td>3.65</td>
</tr>
<tr>
<td>aleph1.library.nd.edu</td>
<td>29.64</td>
<td>72.28</td>
<td>27.65</td>
<td>721.68</td>
<td>961.48</td>
</tr>
<tr>
<td>aleph2.library.nd.edu</td>
<td>21.66</td>
<td>72.28</td>
<td>0.22</td>
<td>283.11</td>
<td>320.00</td>
</tr>
<tr>
<td>annex-staging.library.nd.edu</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>api2pprd-vm.library.nd.edu</td>
<td>8.88</td>
<td>19.01</td>
<td>0.92</td>
<td>17.53</td>
<td>4.71</td>
</tr>
<tr>
<td>api2prod-vm.library.nd.edu</td>
<td>9.52</td>
<td>19.01</td>
<td>1.72</td>
<td>19.21</td>
<td>6.84</td>
</tr>
<tr>
<td>apiprod-vm.library.nd.edu</td>
<td>12.20</td>
<td>19.04</td>
<td>1.31</td>
<td>36.77</td>
<td>23.83</td>
</tr>
<tr>
<td>assetpprd-vm.library.nd.edu</td>
<td>14.32</td>
<td>19.04</td>
<td>1.95</td>
<td>36.03</td>
<td>27.74</td>
</tr>
<tr>
<td>assetprod-vm.library.nd.edu</td>
<td>7.40</td>
<td>50.72</td>
<td>0.34</td>
<td>28.37</td>
<td>15.19</td>
</tr>
<tr>
<td>azariah.library.nd.edu</td>
<td>11.30</td>
<td>50.72</td>
<td>0.34</td>
<td>23.25</td>
<td>9.85</td>
</tr>
<tr>
<td>beehivepprd-vm.library.nd.edu</td>
<td>8.52</td>
<td>53.21</td>
<td>0.03</td>
<td>38.88</td>
<td>1.93</td>
</tr>
<tr>
<td>beehiveprod-vm.library.nd.edu</td>
<td>6.18</td>
<td>19.01</td>
<td>0.71</td>
<td>13.79</td>
<td>1.06</td>
</tr>
<tr>
<td>biographiespprd-vm.library.nd.edu</td>
<td>6.35</td>
<td>19.01</td>
<td>0.61</td>
<td>13.69</td>
<td>1.04</td>
</tr>
<tr>
<td>biographiesprod-vm.library.nd.edu</td>
<td>7.04</td>
<td>19.01</td>
<td>0.74</td>
<td>22.83</td>
<td>9.82</td>
</tr>
<tr>
<td>bukki.library.nd.edu</td>
<td>62.12</td>
<td>255.48</td>
<td>1.30</td>
<td>1760.77</td>
<td>761.41</td>
</tr>
<tr>
<td>caiaphas.library.nd.edu</td>
<td>10.19</td>
<td>53.21</td>
<td>1.97</td>
<td>975.91</td>
<td>363.08</td>
</tr>
<tr>
<td>cds.library.nd.edu</td>
<td>4.89</td>
<td>19.04</td>
<td>0.71</td>
<td>14.97</td>
<td>2.12</td>
</tr>
<tr>
<td>confluence-devl.library.nd.edu</td>
<td>6.41</td>
<td>19.01</td>
<td>1.03</td>
<td>14.62</td>
<td>1.12</td>
</tr>
</tbody>
</table>
Other Resources

- 3rd Party Vendors
  - Monitoring
  - Billing
  - Migrating
- APIs
  - JSON and Beanstalk
- Amazon Suite of Tools
  - To Manage Infrastructure
  - Trusted Advisor
To Meet the December 2017 Deadline

Planning

- Evaluation
    - Experimentation
    - New Services
    - Resource Usage Prediction
    - Budget Reallocation

- Budgeting
    - Budget Request
    - SaaS Estimates
    - Preliminary Migration

- Migration
  - Aug 2016 – Dec 2017
    - Migration
      - To SaaS Providers
      - To AWS
      - To New Data Center

Aug 2016 – Dec 2017
Challenges

- **Vendor Lock-in**
  - Optimization requires doing it the Amazon way
- **May lose knowledge of how to administer systems**
- **Cost balance is realized through optimized utilization of resources**
  - What can be turned off
  - during non-business hours?
• Shifting the Mental Model
  – Flexibility vs. Efficiency
• Introducing New Perspectives
  – Stasis in Patterns of What You Know Works
• Bringing Your Staff Along
• Change is Scary and Requires Investment
  – Defining New Positions
  – Retraining
• Be careful of sunk cost fallacy
Recommendations from the Front

- Experiment
  - Not just plug and play
- Training
- Expect Mistakes
  - Budget for them - can be costly
- Rules can be counter-intuitive
- Get developers involved
- Don’t Underestimate the Important
  - of Change Management
- Develop an Exit Strategy
  - Build that into your implementation
  - Balance Optimization with Exit Flexibility