

▶ Measurement Goal

Collect performance data for a typical User Transaction:

1. Navigate to AOL Portal (www.aol.com)
2. Click Mail button
3. Sign in & Load mail
4. Click Inbox
5. Sign out

AOL
Operations Analysis

2

The slide has a white background with a faint geometric pattern of overlapping triangles. The AOL logo and 'Operations Analysis' text are located in the bottom-left corner. A small number '2' is centered at the bottom of the slide.

Step 1 – Navigate to AOL portal



Step 2 – Click Mail button



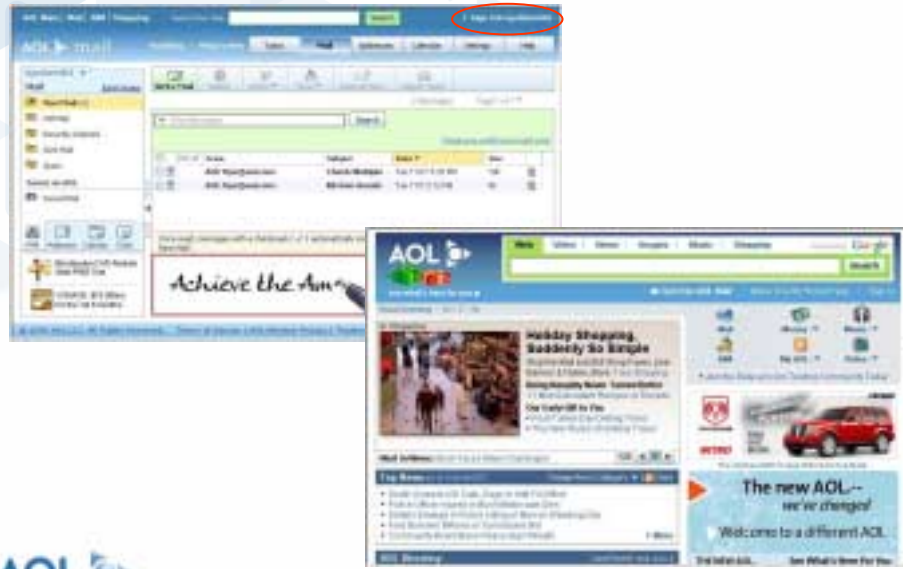
Step 3 – Sign in & Load mail



Step 4 – Click Inbox



Step 5 – Sign out

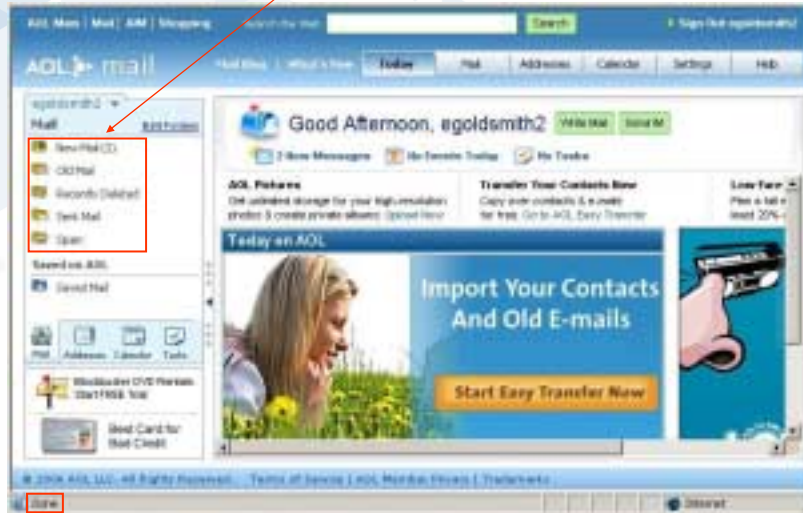


Measurement Challenges

- AOL Webmail is a Rich Internet Application (RIA)
 - Asynchronous data retrieval via hidden IFRAMES
 - Pages are dynamically updated (client-side) as data is retrieved
 - Page content can be updated even after the browser says "Done"
- Detecting Page Completion for RIAs is a challenge
 - Browser may indicate Page Complete (i.e. "Done")
 - But the page may not be complete from the application perspective
 - "Complete" vs. "Really Complete" (or "Complete Enough")

Page Completion

This page is considered *complete* when the Folder List is displayed ...



... which occurs independently of standard Page Completion events (i.e. "Done").

Keynote Transaction Perspective (TxP)

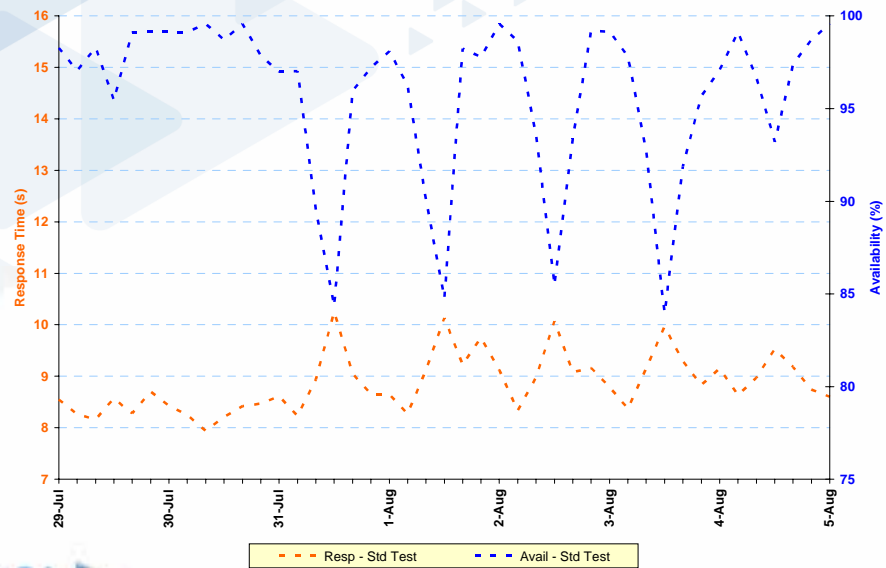
- The Keynote Recorder provides two methods for detecting Page Completion:
 - *Download Cycle count*
 - Corresponds to IE's ReadyState (i.e. when it says "Done")
 - *Document Received count*
 - Corresponds to DocumentComplete events (one for each frame)
- Page Completion can be supplemented with a *Wait Time*
 - After Page Completion is detected, wait up to *x* additional seconds
- However, these methods may not be accurate enough for RIAs
 - JavaScript, Java Applets, Flash and ActiveX controls allow RIAs to make HTTP requests after the standard page completion events
 - This post Page Completion activity is not captured

Standard Test

- **Typical User Transaction:**
 1. Navigate to AOL Portal (www.aol.com)
 2. Click Mail button
 3. Sign in & Load mail
 4. Click Inbox
 5. Sign out
- **Normal Page Completion isn't enough for step 3**
 - 5 second Wait Time added

Standard Test - Results

Response Time & Availability Correlation



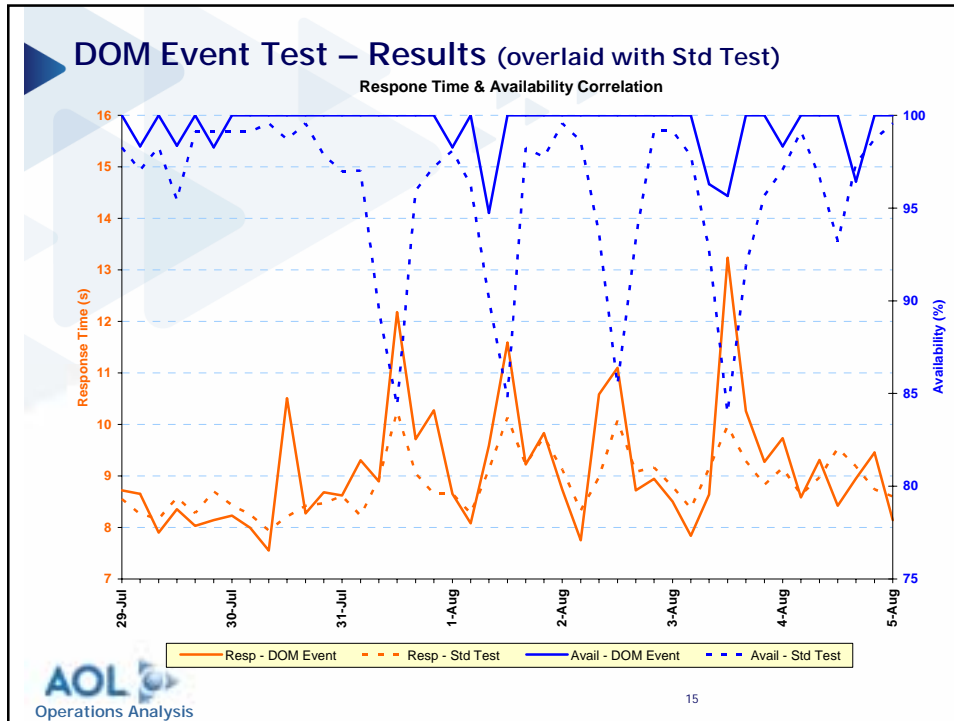
Standard Test – Results Discussion

- **Results show daily, cyclical drop in availability & corresponding increase in response time**
 - Corresponds to peak usage hours
- **Page Completion is occurring before Folder List is displayed**
 - Without Folder List, can't continue with next step
 - Tests fail with 'navigation error'
- **These test failures do not reflect user experience**

Enter Keynote's DOM Event Functionality

- **Page Completion can be tied to HTML DOM¹ activity**
 - Monitor specific DOM element state or event
 - JavaScript can be used to "poll" specific element state
- **Original User Transaction test modified**
 - Step 3 – Sign in & Load mail
 - Page Completion triggered when the size of the folder list DIV element (where the Inbox and other folders are shown) becomes non-zero (i.e. when the folder list is displayed)
- **The new test more accurately reflects the user experience**

¹Document Object Model (DOM) - An interface that provides hierarchical (tree-like) representation of HTML documents and allows programs and scripts to dynamically access and update their content, structure and style.



DOM Event - Results Discussion

- Availability is much more uniform – no longer tracks system load
- Response time 'cycle' peaks have increased
 - Page Completion now occurs when Folder List is actually displayed
 - Takes longer during peak usage hours
- More representative of user experience

AOL Operations Analysis

16