

Enterprise Search with MOSS 101

XiKST????

- Pronounced: Zik-st
- Boutique Technology Consultancy of Software Architects
- Focused on delivering business value using Microsoft Office SharePoint Server and emerging .NET technologies
- Microsoft Partner

Why Enterprise Search?

- Time is Money
 - IDC Whitepaper “**The Hidden Costs of Information Work**”

Task	Average Hours per Worker per Week	Cost per Worker per Week* (\$)	Cost per Worker per Year* (\$)
Email: read and answer	14.5	418.3	21,752.9
Create documents	13.3	333.7	19,952.7
Analyze information	9.6	277.0	14,401.9
Search	9.5	274.1	14,251.9
Edit/review	8.8	253.9	13,201.8
Gather information for documents	8.3	240.0	12,481.7

- What does this mean?

Why Enterprise Search?

- Information Explosion

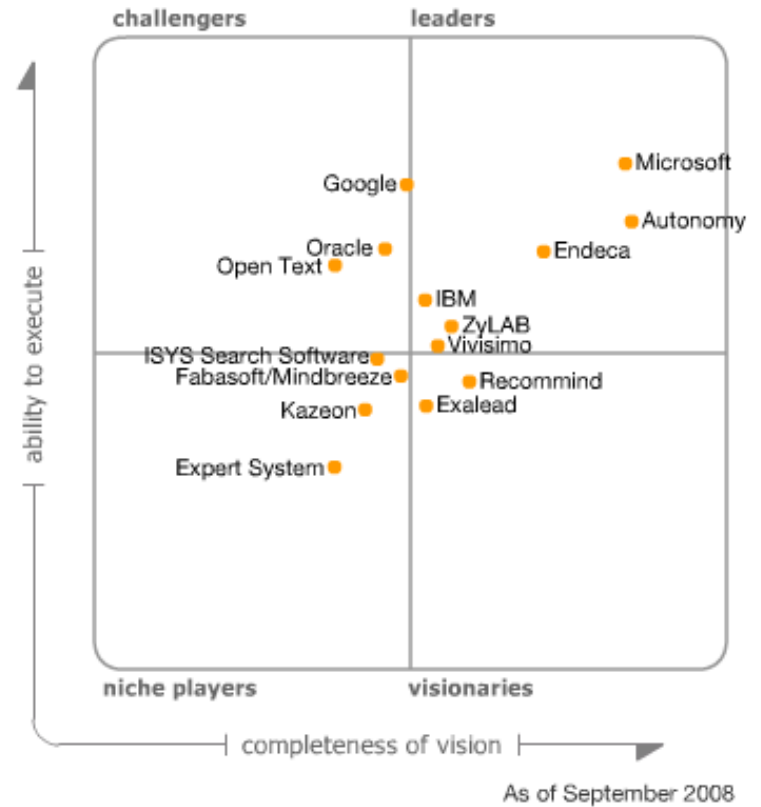
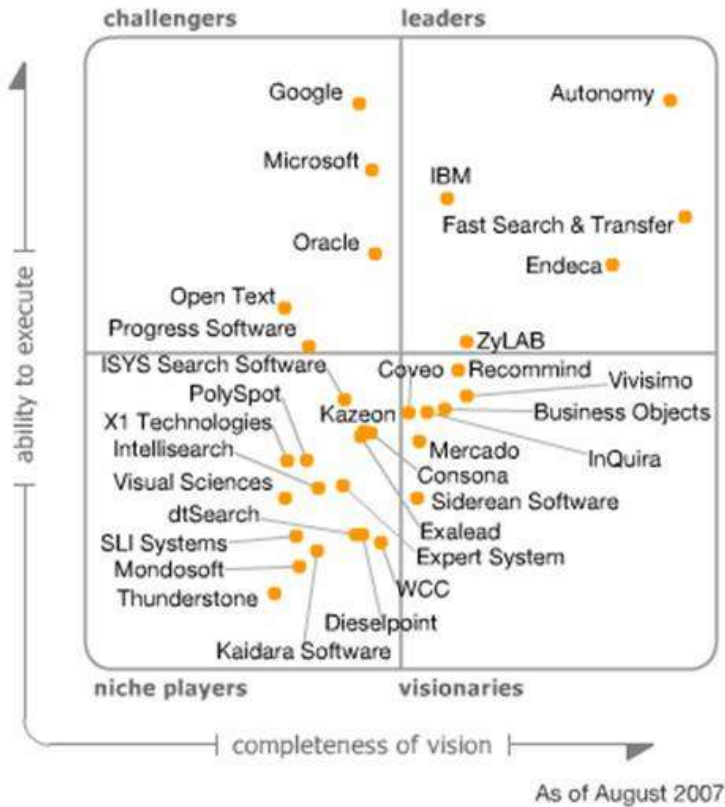
- IDC Whitepaper “Cutting the Clutter: Tackling Information Overload at the Source”

“Each year the amount of information ***created in the enterprise, paper and digital combined, grows faster than 65%.***”

Why Enterprise Search?

- Don't Forget User B
 - User A looks for a **type** of information. He needs intuitive navigation, meaningful taxonomies and a well designed interface.
 - User B looks for an **instance** of information. He needs to get his hands on a specific document...the faster the better.
 - Both users benefit from good architecture, and both benefit from solid search capabilities. User B doesn't come up much in planning meetings.

Why MOSS Search?



What Is Search?

- Relevance is king!
 - “In the context of information science and information retrieval, **relevance** denotes how well a retrieved set of documents (or a single document) meets the information need of the user.” – Wikipedia
- Corpus colossal
 - The corpus is the entire body of searchable information.
 - Documents
 - Pages
 - List Items
 - People
 - Database records

What are the physical/logical pieces?

- Server Roles
 - Index
 - Content Sources
 - IFilter
 - Query
 - To propagate or not to propagate? That is the question.
 - Who knows you?
- Search Database
- Index file
- One index per SSP!

Just how big is that index file?

It's as easy as:

Total_Corpus_Size (in GB) x File_Size_Modifier x 2.85

- File size modifier is between
 - 1.0 for 1KB average
 - .12 for 10KB
 - .05 for 100KB
 - .01 for 500KB
- What's with the 2.85?
- What's the real deal with index sizing?

How does MOSS determine relevance?

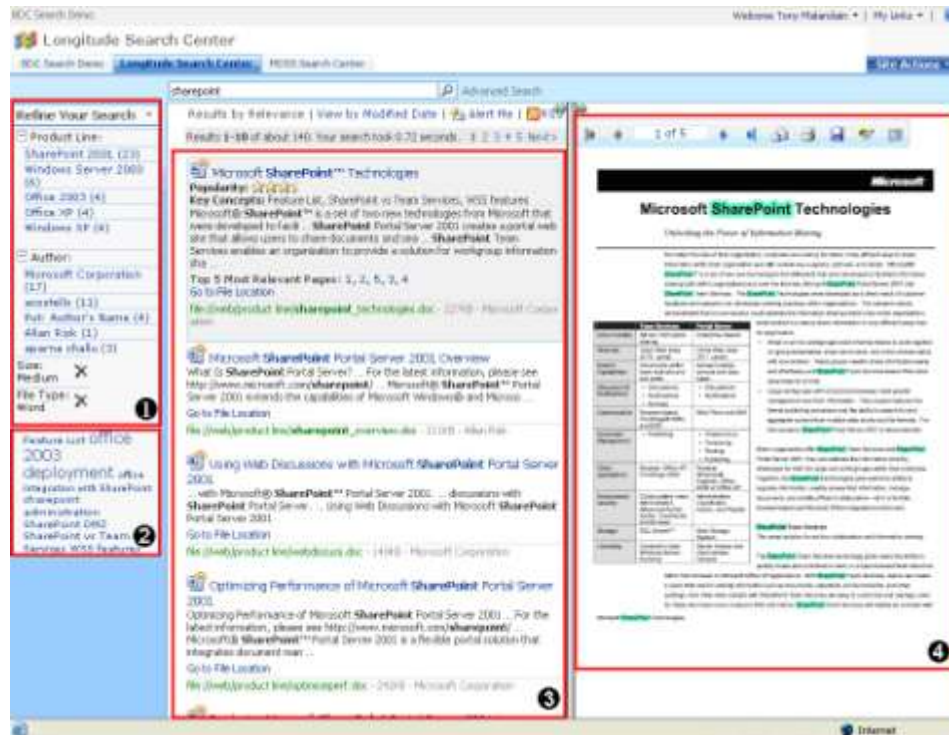
- Anchor Text
- Property Weighting
- Property Length Normalization
- URL Matching
- Title Extraction
- Click Distance
- URL Depth
- Language
- File Type (HTML, PPT, DOC, XML, XLS, TXT, List Item)

What can I do with it?

- Authoritative Pages
- Keywords/best bets
- Managed Properties
- Scopes!
- IE integration
- Thesaurus File
- Noise Words File
- Application Integration
- Check those reports

What else can I do to improve the search experience?

- Faceted Search project on CodePlex
- A product like Longitude search from BA Insights
 - <http://www.ba-insight.net/>



Does Michael talk too much?

- Demo

I've seen the future, and it's...

FAST!

- Huge scalability (billions of documents)
- Tunable indexing and query processing
- Taxonomic navigation and faceted search
- Sub-second query performance
- Search-driven application and process development
- Context sensitive search
- Result ranking and recommendations
- ...more after the SharePoint conference

Links of Interest

- <http://www.point8020.com/EntSearchTraining.aspx>
 - Martin Harwar is a fantastic instructor (he has written a lot of content for MSDN) and has a free 3-day course on search...highly recommended!
- <http://blogs.msdn.com/sharepoint/>
 - The official SharePoint team blog. Check out the 2010 tech preview by clicking the link here.
- <http://technet.microsoft.com/en-us/library/cc835623.aspx>
 - Technet links for improving search results. Lots of good pointers here.

Demo XSL – Raw XML

- To get the raw XML output in the search core results web part, we used this as our xsl stylesheet:

```
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="xml" version="1.0" encoding="UTF-8"
    indent="yes"/>
  <xsl:template match="/">
    <xmp><xsl:copy-of select="*" /></xmp>
  </xsl:template>
</xsl:stylesheet>
```

Demo XSL – Simple Blog List

- For a VERY simple list of recent blogs, we used this as our xsl:

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >
<xsl:template match="/">
<xsl:call-template name="dvt_1.body"/>
</xsl:template>
<xsl:template name="dvt_1.body">
  <div class="srch-results">
    <xsl:apply-templates />
  </div>
</xsl:template>
<xsl:template match="Result">
  <xsl:variable name="id" select="id"/>
  <xsl:variable name="url" select="url"/>
  <span class="srch-Title">
    <a href="{url}" id="{concat('CSR_', $id)}" title="{url}">
      <xsl:value-of select="title"/>
    </a>
    <br/>
  </span>
<div>By: <xsl:value-of select="author" /></div>
<div>On: <xsl:value-of select="write" /></div>
  <div class="srch-Description">
    <xsl:choose>
      <xsl:when test="description[. != '']">
        <xsl:value-of select="description"/>
      </xsl:when>
    </xsl:choose>
  </div >
</xsl:template>
</xsl:stylesheet>
```

