• Ticketmaster Online:
  - ticketmaster.com
  - ticketmaster.(uk|au|nz|it|de|es)
  - livenation.com

• Large Perl shop
  - Perl + Template Toolkit MVC
  - custom Apache C modules

• Make Real Money™
  - 2009: processed $1.3B in ticket sales
Search Redesign Goals

• Product
  – Event-based
  – Drill down
  – "Better"

• Management
  – Generic metadata
  – Current technology

• Engineering
  – Something not a steaming pile of poo
Engineering Issues

• Codebase
  – Fragile
  – Difficult to impossible to maintain

• Performance
  – Application degradation
  – MySQL spiral-of-death

• Architecture
  – Insane DB-to-search population times
  – Scaling
  – Home-grown search technology
Timeline

• Late 2007
  – TM Search officially sucked
  – Management interested in Lucene
  – "Solr Out of the Box" by Chris Hostetter

• April 2008
  – First specification from product
  – Solr proof-of-concept presented

• May 2008
  – Product specification finalized
  – HTML completed
Timeline

• August 2008
  – Front-end demo

• September 2008
  – QA hand-off

• November 2008
  – Partial launch

• January 2009
  – Full launch
The Speed of Success

• Spec to QA: 6 months

• Engineers: 4
  – Architect & Lead Engineer
  – AJAX Rock Star
  – Amazing Sysadmin
  – Jr. Engineer
TM is Solr Powered

- Search
- Browse
- MyAccount
- Alerts
- Sitemap
- Partner Feeds
- Internal API
ticketmaster.com

• 3 forward-facing Solr slaves
  – 8 x 2.8GHz cores
  – 16GB RAM
    • 2.5GB to Solr
  – 90% CPU idle during recent onsales

• 1 Solr master

• Full data construction nightly
  – 30 minutes from DB to slaves

• Incremental updates through the day
  – events: every minute
  – venues and artists: every 3 hours
Old Application Design

mod_perl

MySQL

MySQL

MySQL

MySQL

MySQL

MySQL

MySQL
New Application Design
• Language agnostic
  – HTTP querying
  – JSON output

• Simple

• Feature rich
  – facets
  – mispel

• Large user base and community
Solr, A Perfect Fit?

• Very little data
  – 1GB index

• Broad but shallow
  – 250,000 things
  – 17 languages
  – 11 properties

• Volatile business rules
  – Changes every minute
What's in a Name?

• 250,000 things
  – Artists
  – Events
  – Venues

• 97.325% are proper names

• Proper Names are Hard™

• Eccentric Bands are Even Harder™
• "We should be able to find Hannah Montana with one spelling mistake"
The Google Effect

• "If Google can do it, why can't we?"
• Google has 11,500,000 documents for Hannah Montana... all spelled wrong
| hana mont\n|------------------|------------------|
| hana montana       | 11,500,000 results |
| hana montana games | 7,320,000 results  |
| hana montana songs | 3,270,000 results  |
| hana montana.com   | 5,923,000 results  |
| hana montana music | 10,903,000 results |
| hannah montana pictures | 7,880,000 results |
| hannah montana video | 5,060,000 results  |
| hannah montana guitar | 2,772,000 results  |
| hannah montana lyrics | 2,656,000 results  |
| hannah montana dress up | 1,510,000 results  |
On Haystacks...

• "We should be able to find Hannah Montana with one spelling mistake"
• Fine... if you actually have an artist named "Hannah Montana"
Search is Important

• Although misguided, product is right

• Search
  – drives sales
  – primary point of customer interaction
  – highly visible
  – needs to work

• When search is broken
  – your company loses money
  – *you* hear all about it
  – *your* life sucks
Don't Make Stuff Up

• Look at historical data
  – top 2000 misses for 6 months
• Use usage patterns to drive design
Top 2000 Misses

• City, state
  - boston, ma

• Logical misspell
  - flight of the concords

• Out-of-range misspell
  - circus olay
  - yyy

• Crunched
  - janetjackson

• Non-existent
  - amy lee
Miss-Driven Solution

• Keywords
  – all the stuff people search for

• Synonyms
  – handle out-of-range searches

• Solr toolkit
  – UTF-8
  – spellchecker
Keywords

- Event
- Artists
- Venue
  - city
  - state
  - postcode
- Date
  - month
  - year
  - day of week
- Genre
"search-en":"MLB Anaheim Angels San Diego CA California New York Yankees Jack Murphy Stadium August 2011 Saturday 92108 Baseball mlbanaheimangels anaheimangels newyorkyankees"
<fieldType name="search-en"
class="solr.TextField" positionIncrementGap="100">

<analyzer type="index">
    <tokenizer class="solr.WhitespaceTokenizerFactory"/>
    <filter class="solr.ISOLatin1AccentFilterFactory" />
    <filter class="solr.WordDelimiterFilterFactory"
            preserveOriginal="1"
            generateWordParts="1"
            generateNumberParts="1"
            catenateWords="1"
            catenateNumbers="1"
            catenateAll="1"/>
    <filter class="solr.LowerCaseFilterFactory"/>
    <filter class="solr.SynonymFilterFactory"
            synonyms="synonyms.txt"
            ignoreCase="true"
            expand="true"/>
    <filter class="solr.StopFilterFactory"
            ignoreCase="false"
            words="stopwords-en.txt"/>

</analyzer>

</fieldType>
<analyzer type="query">
  <tokenizer class="solr.WhitespaceTokenizerFactory"/>
  <filter class="solr.ISOLatin1AccentFilterFactory" />
  <filter class="solr.WordDelimiterFilterFactory"
    preserveOriginal="0"
    splitOnCaseChange="0"
    generateWordParts="1"
    generateNumberParts="1"
    catenateWords="0"
    catenateNumbers="0"
    catenateAll="0"/>
  <filter class="solr.LowerCaseFilterFactory"/>  
  <filter class="solr.StopFilterFactory"
    ignoreCase="false"
    words="stopwords-en.txt"/>
</analyzer>

</fieldType>
On Stemming...

• Language-specific search fields
  - search-en
  - search-de

• Snowball too aggressive
  - Wicked => Wick
  - Chuck Wicks => Wick

  - Angels Baseball => Angel
  - Los Angeles => Angel
Synonyms

• Help with hard and out-of-range stuff
  – John Cougar, John Mellencamp
  – STP, Stone Temple Pilots
  – First Union, Wachovia
  – P!NK, Pink

• Applied at index time
  – re-index required to apply changes
<requestHandler name="Search::Model::JSON::Event::Search"
class="solr.DisMaxRequestHandler" >
  <lst name="defaults">
    <str name="echoParams">none</str>
    <str name="indent">off</str>
    <int name="rows">500</int>
    <int name="start">0</int>
  </lst>
  <lst name="invariants">
    <str name="mm">100%</str>
    <str name="wt">json</str>
    <str name="facet">false</str>
    <str name="sort">EventDate asc, EventName asc</str>
  </lst>
  <lst name="appends">
    <str name="fq">Type:Event</str>
    <str name="fq">-SearchableUntil: [* TO NOW]</str>
  </lst>
</requestHandler>
http://host:8080/solr/select
  ?q=boston red sox
  &qf=search-en
  &fq=VenueCountry:US
  &fq=+DomainId:1 +LangCode:en-us
  &qt=Search::Model::JSON::Event::Search

{
  "responseHeader":{
    "status":0,
    "QTime":59},
  "response":{
    "numFound":1,"start":0,"docs":[
    {
      "DocumentId":"Event+260043378B043C67+en-us+1",
      ...
Clean and Simple++

• 16 requestHandler entries
• Code kept clean

• Everything for display stored in Solr
• Some data is very lightly massaged
  – Event on sale "now"?
  – Multiple events at a single venue

• No DB interactions
• Code kept simple
Miss-Driven Solution

• Start with expanded terms and apply tokenizers and filters
  – latin1
  – synonyms

• If match found
  – present results
  – suggest alternatives

• If no match found
  – use first suggestion to re-search
  – suggestions guaranteed to exist
<fieldType name="spell"
        class="solr.TextField"
        positionIncrementGap="100">

    <analyzer type="index">
        <tokenizer class="solr.KeywordTokenizerFactory"/>
        <filter class="solr.LowerCaseFilterFactory"/>
    </analyzer>

    <analyzer type="query">
        <tokenizer class="solr.KeywordTokenizerFactory"/>
        <filter class="solr.LowerCaseFilterFactory"/>
    </analyzer>

</fieldType>
Holy Hanna, Batman!

• Search for "Hanna Montanna"
• 9 occurrences of "Hannah"
• 20 occurrences of "Hanna"
• 20 of "Montana"
• "Did you mean Hanna Montana?"
• "Did you mean Red Sex?"
Request

http://host:8080/solr/select
?q=boston red socks
&qf=search-en
&spellcheck.q=boston red socks
&fq=+DomainId:1 +LangCode:en-us
&qt=Search::Model::JSON::Scan

{"responseHeader":{
   "status":0,  
   "QTime":133},  
"response":{"numFound":0,"start":0,"docs":[]},  
"spellcheck":{
   "suggestions":[
      "boston red socks",{
         "numFound":5,  
         "startOffset":0,  
         "endOffset":16,  
         "suggestion":[
            "boston red sox",  
            "boston celtics"],
      "boston red sox",{
         "numFound":5,  
         "startOffset":0,  
         "endOffset":16,  
         "suggestion":[]
      },
      "boston celtics":{
         "numFound":5,  
         "startOffset":0,  
         "endOffset":16,  
         "suggestion":[]
      }
   ],
   "boston red sox":{
      "numFound":5,  
      "startOffset":0,  
      "endOffset":16,  
      "suggestion":[]
   },
   "boston celtics":{
      "numFound":5,  
      "startOffset":0,  
      "endOffset":16,  
      "suggestion":[]
   }
}
You Sank My Battleship!

• Tier-1
  – more search terms
  – better tokenization
  – synonyms
  – 570 successful searches of 2000
  – 30% outright improvement

• Tier-2
  – misspell logic
  – only 160 missed searches
Suggested Reading

• http://bit.ly/wired-on-google