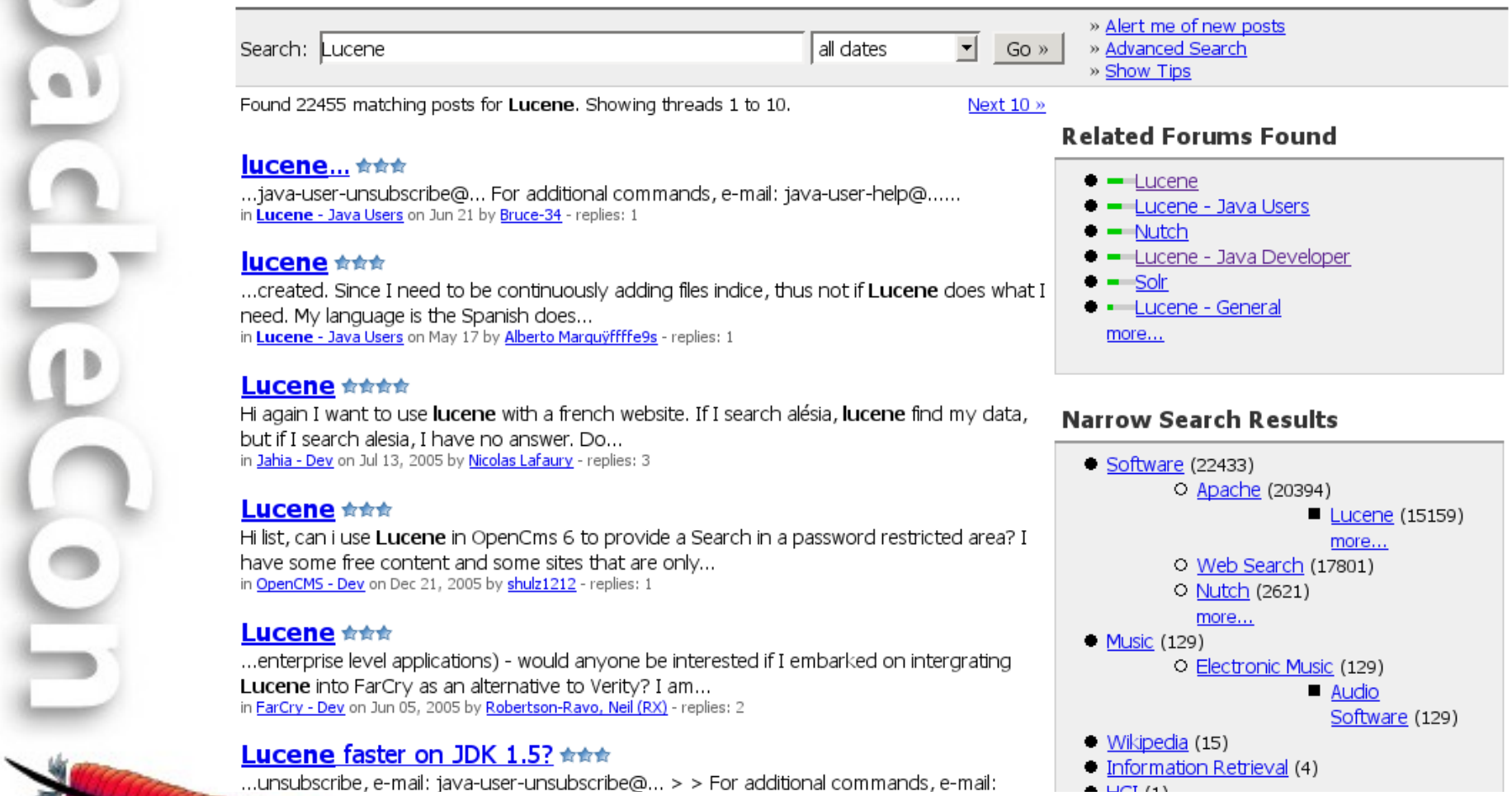
Faceted Searching With Apache Solr

October 13, 2006
Chris Hostetter
hossman – apache – org
http://incubator.apache.org/solr/

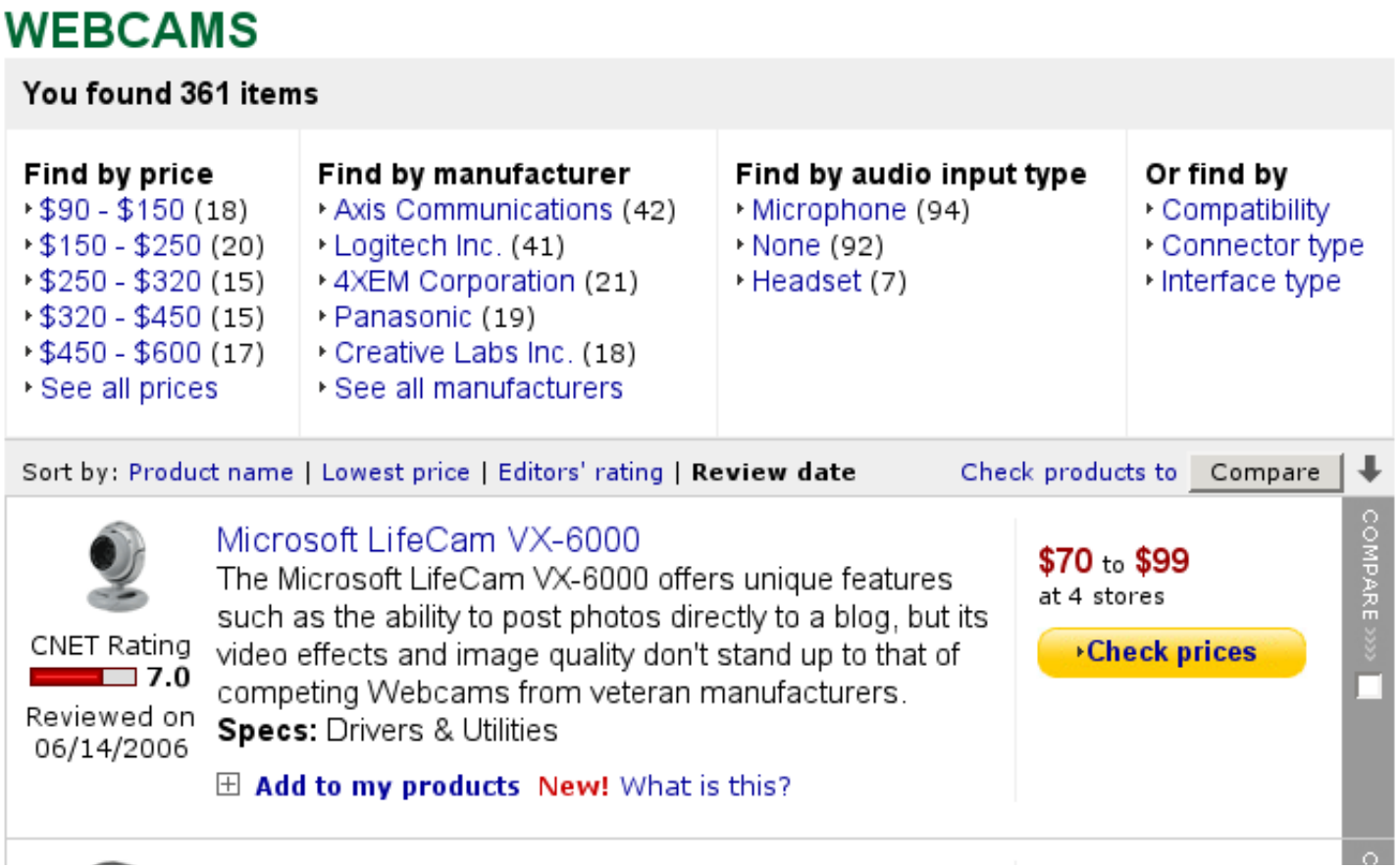
What is Faceted Searching?

Example: Epicurious.com

Example: Nabble.com



Example: CNET.com



Aka: “Faceted Browsing”

"Interaction style where users filter a set of items by progressively selecting from only valid values of a faceted classification system"

- Keith Instone, SOASIS&T, July 8, 2004

Key Elements of Faceted Search

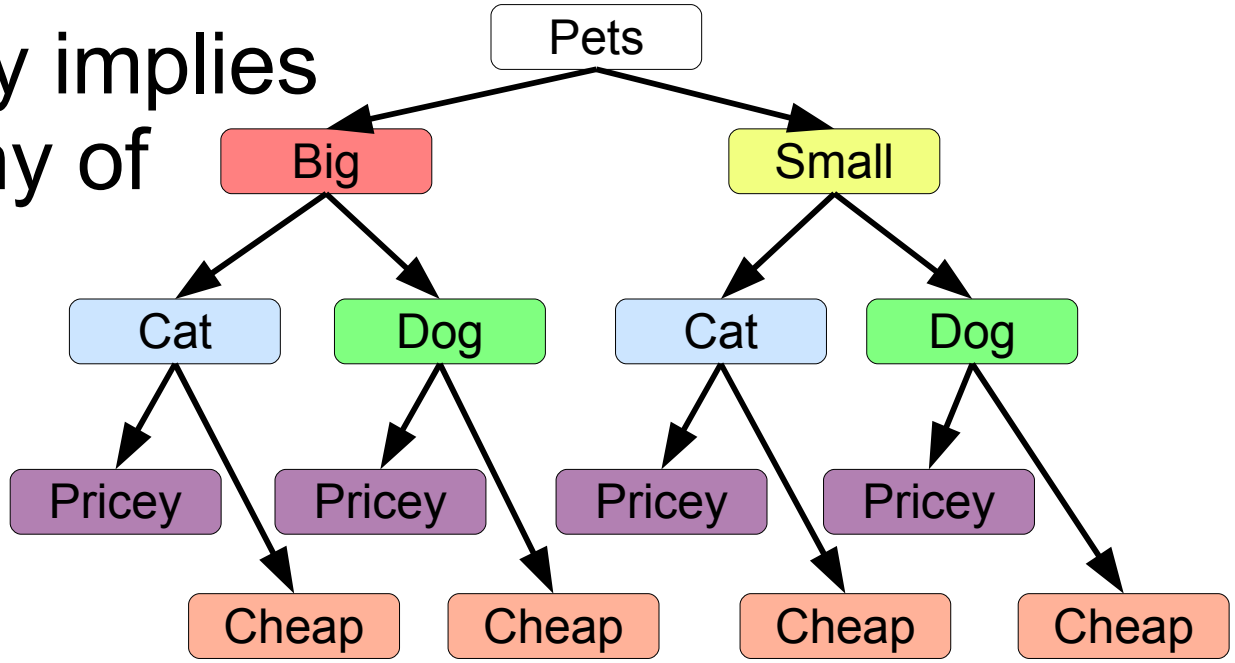
- No hierarchy of options is enforced
  - Users can apply facet constraints in any order
  - Users can remove facet constraints in any order
- No surprises
  - The user is only given facets and constraints that make sense in the context of the items they are looking at
  - The user always knows what to expect before they apply a constraint

Explaining My Terms

- Facet: A distinct feature or aspect of a set of objects; “a way in which a resource can be classified”
- Constraint: A viable method of limiting a set of objects

Dynamic Taxonomy? No.

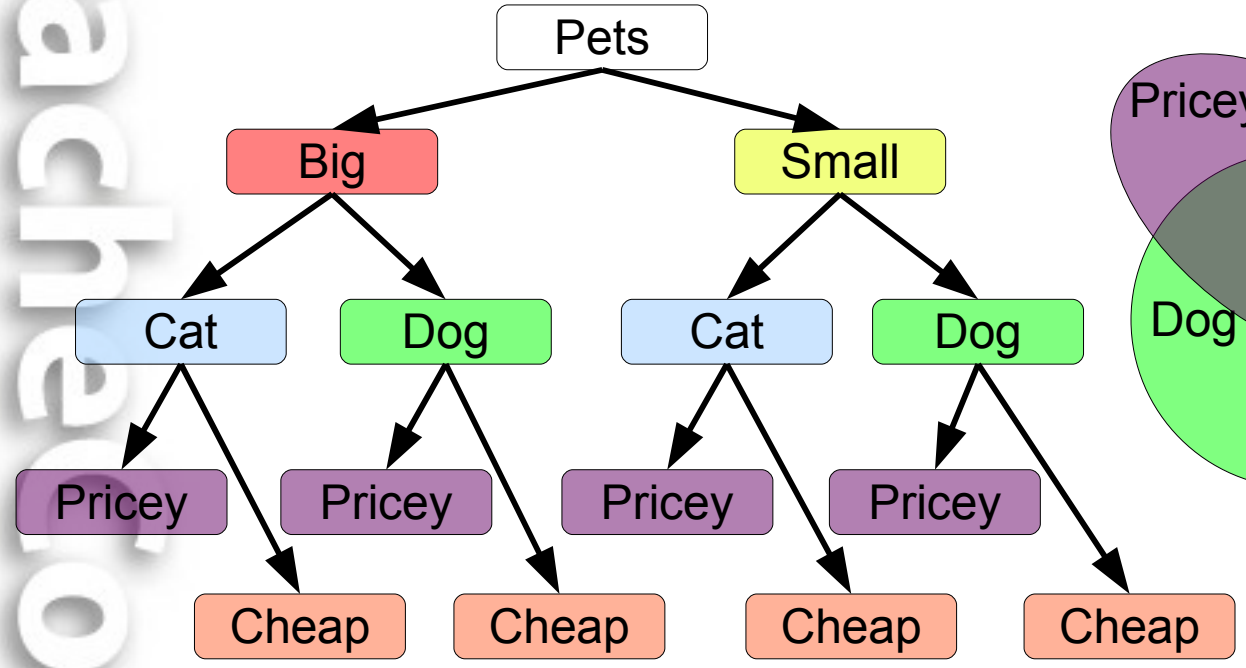
- Bad Description
- Taxonomy implies a hierarchy of subsets



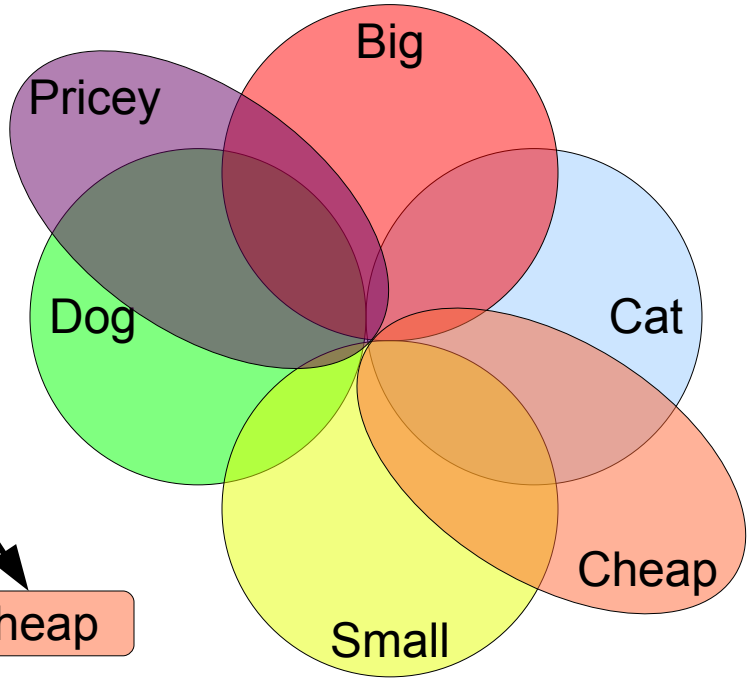
- Hierarchy implies ordered usage of constraints

Why Is Faceted Searching Hard?

Taxonomy Approach



Faceted Approach



- **LOTS** of set intersections
- All permutations can't be easily precomputed

What is Solr?

Elevator Pitch

"Solr is a open source enterprise search server based on the Lucene Java search library, with XML/HTTP APIs, caching, replication, and a web administration interface."

What Does That Mean?

- Information Retrieval application
- Java5 WebApp (WAR) with a web services-ish API
- Uses the Java Lucene search library
- Initially built at CNET
- Now an Apache Incubator project

Lucene Refresher

- Lucene is a full-text search library
  - Maintains inverted index: terms -> documents
- Add documents to an index via IndexWriter object
  - A document is a collection of fields
  - No config files, dynamic field typing
  - Text analysis performed by Analyzer objects
  - No notion of "updating" or "replacing" an existing document
- Search for documents via IndexSearcher object
  Hits = search(Query,Filter,Sort,topN)
- Scoring: tf * idf * lengthNorm

Solr in a Nutshell

- Index/Query via HTTP and XML
- Comprehensive HTML Administration Interfaces
- Scalability - Efficient Replication to Other Solr Search Servers
- Extensible Plugin Architecture
- Highly Configurable and User Extensible Caching
- Flexible and Adaptable with XML configuration
  - Data Schema with Dynamic Fields and Unique Keys
  - Analyzers Created at Runtime from Tokenizers and TokenFilters

Example: Adding a Document

```
HTTP POST /update

<add><doc>
 <field name="article">05991</field>
 <field name="title">Apache Solr</field>
 <field name="subject">An intro...</field>
 <field name="cat">search</field>
 <field name="cat">lucene</field>
 <field name="body">Solr is a full...</field>
 <field name="inStock">true</field>
</doc></add>
```

Example: Execute a Query

```
HTTP GET
  /select/?qt=foo&wt=bar&start=0&rows=10&q=solr

<?xml version="1.0" encoding="UTF-8"?>
<response>
  <responseHeader>
    <status>0</status><QTime>1</QTime>
  </responseHeader>
  <result numFound="1" start="0">
    <doc>
      <arr name="cat">
        <str>lucene</str><str>search</str>
      </arr>
      <bool name="inStock">true</bool>
      <str name="title">Apache Solr</str>
      <int name="popularity">10</int>
      ...
```

Example: SimpleRequestHandler

```
public void handleRequest(SolrQueryRequest req,
                          SolrQueryResponse rsp) {
   try {
      Query q = QueryParsing.parseQuery
         (req.getQueryString(),req.getSchema());

      DocList results =
         req.getSearcher().getDocList
            (q, (Query)null, (Sort)null,
             req.getStart(), req.getLimit());

      rsp.add("simple results", results);
      rsp.add("other data", new Integer(42));

   } catch (Exception e) {
      rsp.setException(e);
   }
}
```

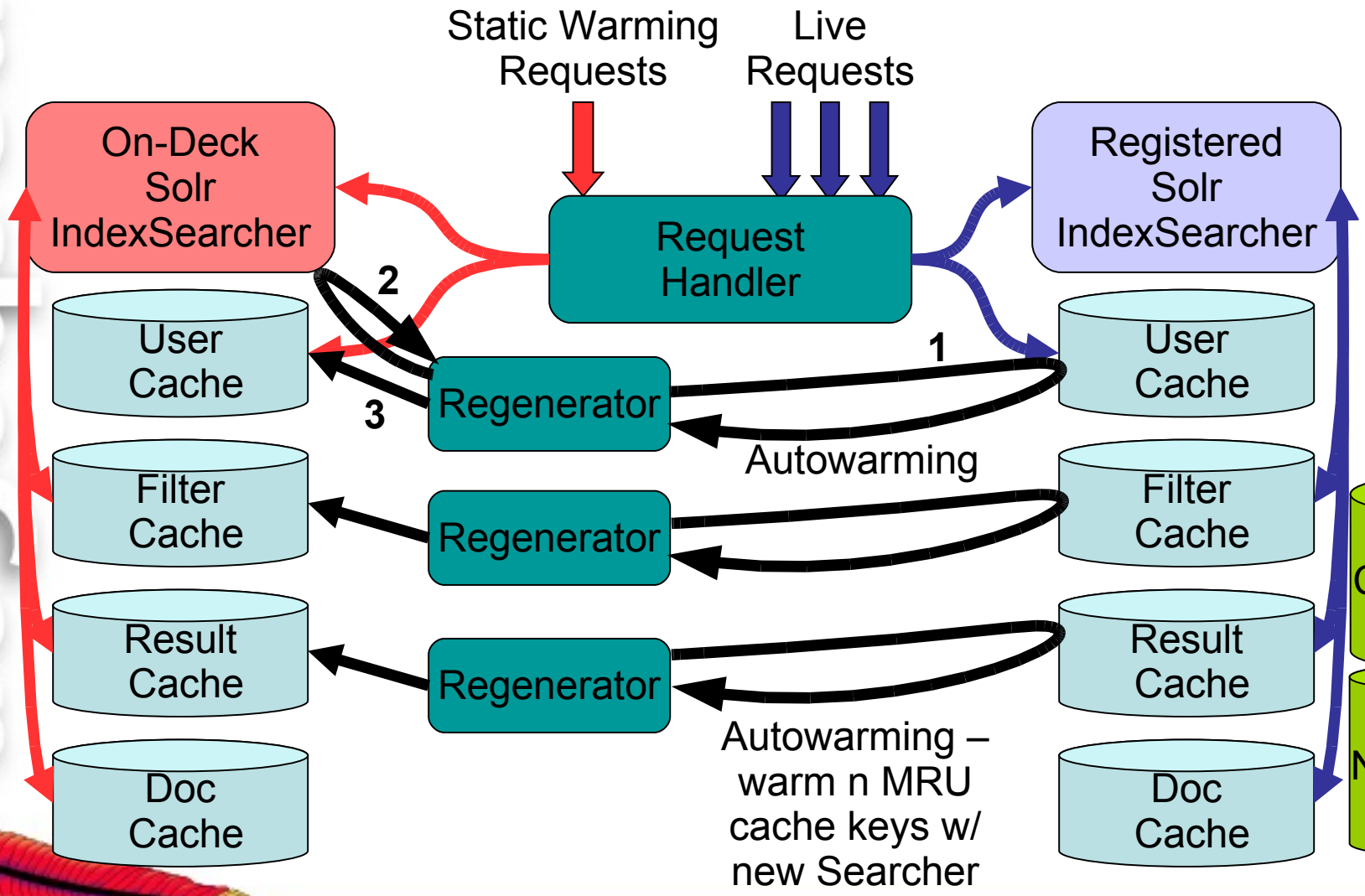
DocLists and DocSets

- DocList - An ordered list of document ids with optional score
  - A subset of the complete list of documents actually matched by a Query
- DocSet - An unordered set of Lucene Document Ids
  - Typically the complete set of documents matched by a query
  - Multiple implementations optimized for different size sets
  - Foundation of Faceted Searching in Solr

Caching

- IndexSearcher's view of an index is fixed
  - Aggressive caching possible
  - Consistency for multi-query requests
- Types of Caches:
  - filterCache: Query => DocSet
  - resultCache: (Query,Sort,Filter) => DocList
  - documentCache: docId => Document
  - userCaches: Object => Object
    - application specific, custom query handlers

Smart Cache Warming



Case Study

CNET's First Solr Powered Page

Old Crappy Version

Sort by: Review date Go! 1-23 of 23

Filter results

Price: any

Manufacturer: any

Audio input type: any

Compatibility: any

Connector type: any

Interface type: any

Filter my results

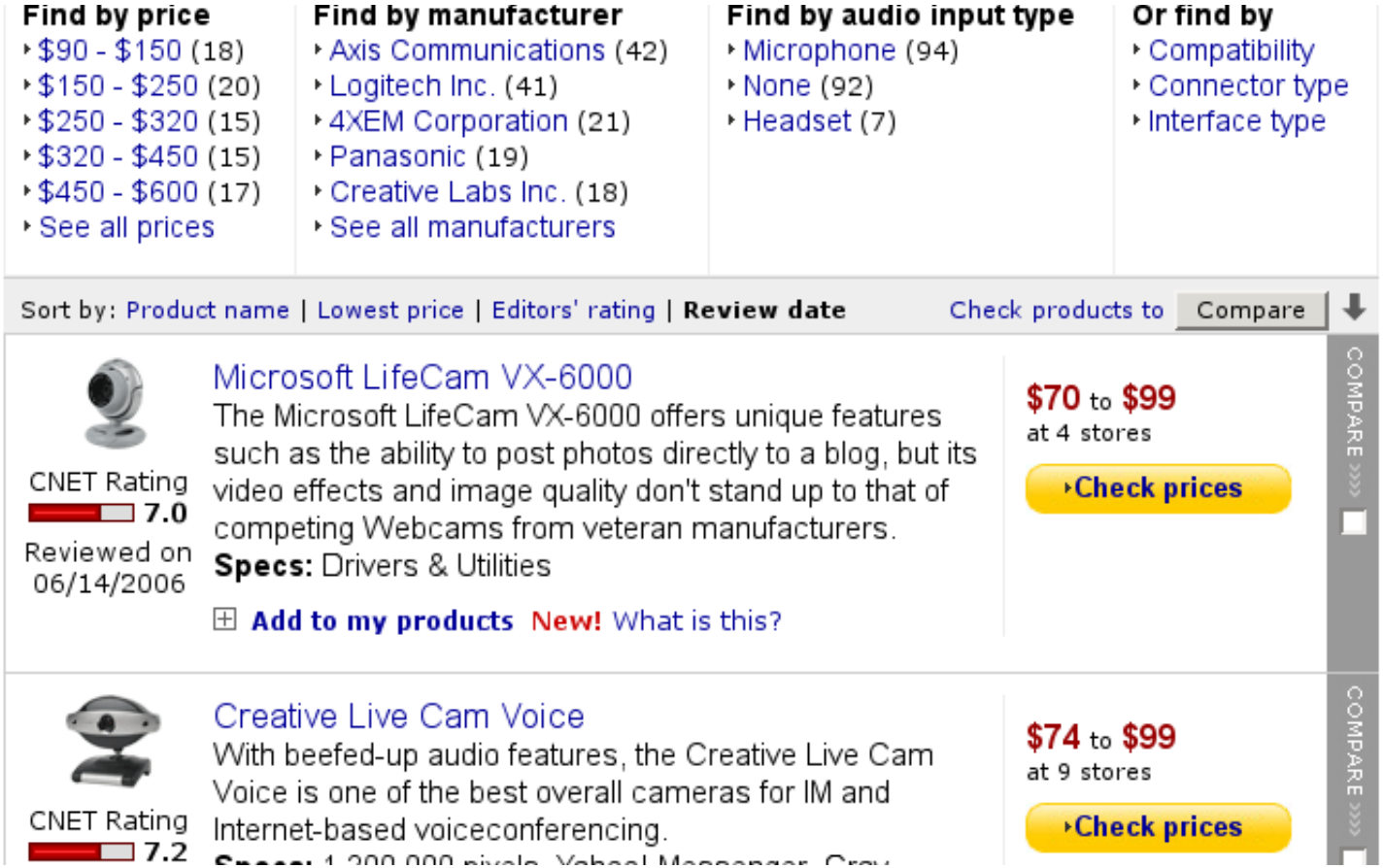
Don't see what you're looking for? Use the filter menus above to narrow down the results.

advertisement
sponsored

Lexar Memory Cards
Compatible with all Digital Cameras SD Card, CompactFlash, Memory

COMPARE		Product	Editors' rating	Price
		**Microsoft LifeCam VX-6000** The Microsoft LifeCam VX-6000 offers unique features such as the ability to post photos directly to a blog, but its video effects and image quality don't stand up to that of competing Webcams from veteran manufacturers. **Review date:** 06/14/2006 **Release date:** 06/13/2006 **Specs:** Drivers & Utilities CNET editor's take	7.0 Very good	Email me when this product is available
		**Creative Live Cam Voice** With beefed-up audio features, the Creative Live Cam Voice is one of the best overall cameras for IM and Internet-based voiceconferencing. **Review date:** 05/16/2006 **Release date:** 05/16/2006 **Specs:** Yahoo! Messenger =Gray =1,300,000 pixels CNET editor's take	7.2 Very good	Email me when this product is available
		**WiLife LukWerks Starter Kit** The WiLife LukWerks system is easy to configure and use, but the software can be cantankerous. Potential users may suffer sticker shock, but it's a deal compared to professionally installed security systems. **Review date:** 03/28/2006 **Release date:** 02/01/2006 **Specs:** Drivers & Utilities CNET editor's take	7.1 Very good	Email me when this product is available

Shiny New Faceted Version



Category Metadata

- Category ID and Label
- Category Query
- Ordered List of Facets
  - Facet ID and Label
  - Facet "Display Type"
    - Ordered List of Constraints
    - Constraint ID and Label
    - Constraint Query

Key Features We Needed In Solr

- Loose Schema with Dynamic Fields
- Efficient implementation of sets and set intersection
- Aggressive set caching
- Plugin Architecture

RequestHandler Psuedo-Code

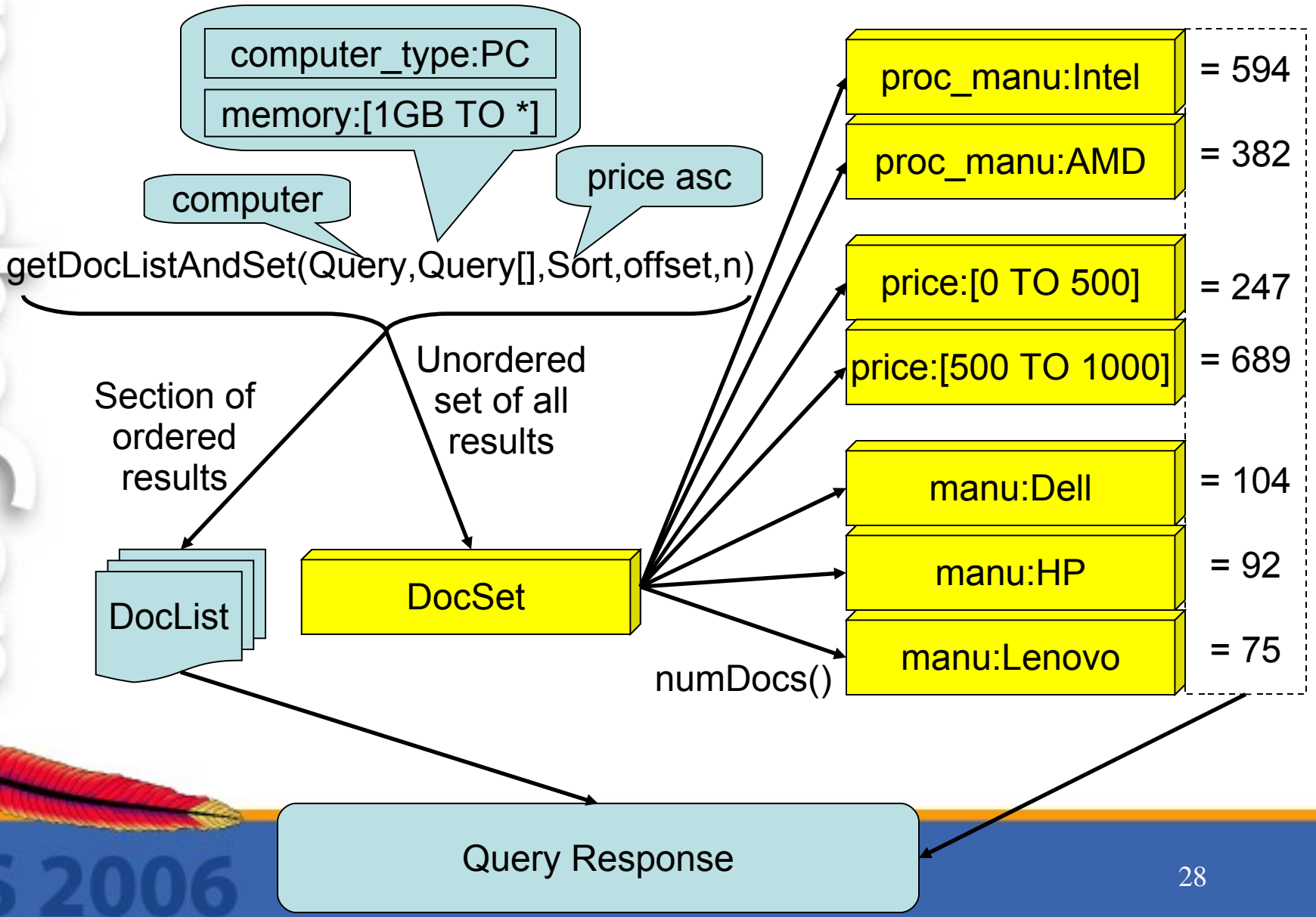
```
Document catMetaDoc =
   searcher.getFirstMatch(categoryDocId)
Metadata m = parseAndCacheMetadata
   (catMetaDoc, searcher).clone()

DocListAndSet results =
   searcher.getDocListAndSet(m.catQuery, ...)

response.add(results.docList)

foreach (Facet f : m) {
   foreach (Constraint c : f) {
      c.setCount(searcher.numDocs(c.query,
                                  results.docSet))
   }
}
response.add(m.dumpToSimpleDatastructures())
```

Conceptual Picture



XML Response

```
- <response>
  - <responseHeader>
      <status>0</status>
      <QTime>17</QTime>
    </responseHeader>
  + <result name="products" numFound="5461" start="0"></result>
  - <lst name="metadata">
    - <lst name="100021">
        <int name="rankDir">1</int>
        <int name="formelement">10</int>
      + <lst name="values"></lst>
        <int name="datatype">3</int>
        <int name="rating">94</int>
        <str name="name">Price</str>
        <int name="attributeId">100021</int>
      </lst>
    - <lst name="1000036">
        <int name="rankDir">0</int>
        <int name="formelement">7</int>
      - <lst name="values">
        - <lst name="5260113">
            <int name="valueId">5260113</int>
            <str name="label">ABS Computer Technologies Inc.</str>
            <str name="rating">50</str>
            <int name="count">7</int>
          </lst>
        - <lst name="11795388">
            <int name="valueId">11795388</int>
```

Simple Faceted Request Handlers

SimpleFacetedRequestHandler

```
...
SolrIndexSearcher s = req.getSearcher();
SolrQueryParser qp = new
   SolrQueryParser(req.getSchema(), null);
Query q = qp.parse( req.getQueryString() );

DocListAndSet results = s.getDocListAndSet
   (q, (List<Query>)null, (Sort)null,
    req.getStart(), req.getLimit());

NamedList counts = new NamedList();
   for (String fc : req.getParams("fc")) {
      counts.add(fc, s.numDocs(qp.parse(fc),
                               results.docSet));
   }
rsp.add("facet constraint counts", counts);
rsp.add(“your results”, results.docList);
...
```

SimpleFacetedRequestHandler

```
?qt=qfacet&q=video&fc=inStock:true&fc=inStock:false
```

```
- <response>
  - <responseHeader>
      <status>0</status>
      <QTime>1</QTime>
    </responseHeader>
  - <lst name="facet constraint counts">
      <int name="inStock:true">1</int>
      <int name="inStock:false">2</int>
    </lst>
  - <result numFound="3" start="0">
    - <doc>
      - <arr name="cat">
          <str>electronics</str>
          <str>music</str>
        </arr>
      - <arr name="features">
          <str>iTunes, Podcasts, Audiobooks</str>
        - <str>
            Stores up to 15,000 songs, 25,000 photos, or 150 hours of video
          </str>
        - <str>
            2.5-inch, 320x240 color TFT LCD display with LED backlight
          </str>
          <str>Up to 20 hours of battery life</str>
        - <str>
            Plays AAC, MP3, WAV, AIFF, Audible, Apple Lossless, H.264 video
```

DynamicFacetedRequestHandler

```
...
IndexReader r = s.getReader();
NamedList facets = new NamedList();
for (String ff : req.getParams("ff")) {
   Map counts = new HashMap();
   facets.add(ff, counts);

   TermEnum te = r.terms(new Term(ff,""));
   do {
      Term t = te.term();
      if (null == t || ! t.field().equals(ff))
         break;

      counts.put(t.text(), s.numDocs
         (new TermQuery(t), results.docSet));
   } while (te.next());
}
rsp.add("facet fields", facets);
rsp.add(“my results”, results.docList);
...
```

DynamicFacetedRequestHandler

```
?qt=dfacet&q=video&ff=cat&ff=inStock
```

```
- <lst name="facet fields">
  - <lst name="cat">
      <int name="search">0</int>
      <int name="memory">0</int>
      <int name="graphics">2</int>
      <int name="card">2</int>
      <int name="connector">0</int>
      <int name="software">0</int>
      <int name="electronics">3</int>
      <int name="copier">0</int>
      <int name="multifunction">0</int>
      <int name="camera">0</int>
      <int name="music">1</int>
      <int name="hard">0</int>
      <int name="scanner">0</int>
      <int name="monitor">0</int>
      <int name="drive">0</int>
      <int name="printer">0</int>
    </lst>
  - <lst name="inStock">
      <int name="F">2</int>
      <int name="T">1</int>
    </lst>
</lst>
```

In Conclusion...

Go Use Solr!