Express Mailing Label No. EV 343632189 US

Provisional Patent Application
Docket No. 3412.2.3p

UNITED STATES PROVISIONAL PATENT APPLICATION

of

G. EDWARD POWELL, JR. and MICHAEL ANDERER

for

SYSTEM FOR SOFTWARE SOURCE CODE COMPARISON

SYSTEM FOR SOFTWARE SOURCE CODE COMPARISON

DESCRIPTION

[01] Computers and computer technology have become an integral part of today's society. Computer software running on computers enables the computers to accomplish many different tasks. Computer software includes any kind of computer program that may be run on a computer.

[02] Software engineers develop software by writing source code that is then transformed into a computer program or into part of a computer program. Source code is typically a type of document that people can read and understand. When it is compiled and/or linked it is turned into code that is understood by the computer so that it may be executed.

[03] Many software engineers change jobs in their lifetimes. Their employers may be concerned with whether or not they take source code from their current job to their new job. The development of software often costs a company a substantial amount of resources. Many companies would like to protect their investments in the software they have developed.

[04] Intellectual property law may be used to protect many aspects of computer software.

[05] The open source movement has made it very easy for source code written by one person or company to become part of another piece of software. Open source software is freely available and can be downloaded from the Internet. Employees, with or without their employers permission, may submit source code to open source projects.

[06] It would be beneficial if systems and methods were provided for a person or company to determine whether they are using source code from another source. It would also be beneficial if systems and methods were provided for a person or company to determine whether someone else (e.g., an open source project) is using their proprietary source code.

[07] Figure 1 is a block diagram illustrating a system for comparing source code. Software source code A is taken as input by a profiling block which performs profiling on the source code to produce a metadata file A. The profiling tool may perform multi-source characterizing and a one-way transform. By making the transform a one-way transform the system will protect the proprietary nature of the source code. If the source code could be reverse engineered from the

metadata file, very few companies with proprietary source code would want to use the system for fear of disclosing their source code to others. By making it a one-way transform, they may be comfortable that their confidential information and source code will be kept confidential.

[08] Software source code B is taken as input by a profiling block which performs profiling on the source code to produce a metadata file B. The metadata files are then compared to one another and a report is generated. The report will reflect how closely the two pieces of software resembled one another.

[09] Figure 2 is a block diagram illustrating a system for managing a database of source code and a database of metadata. In one embodiment this system may be on the Internet so that it can be accessed by third parties. Third parties may submit source code, metadata and the like and they may also request comparisons of source code and metadata. In addition, a web agent or crawler may be used to mine data from the Internet or other sources.

[10] What is claimed is:

1. A method for comparing software source code, the method comprising:

profiling one or more pieces of software source code using a one-way transform having a natural language characterization algorithm;

generating metadata based on the profiling;

comparing two or more metadata files; and

generating output reflecting the results of the comparison.

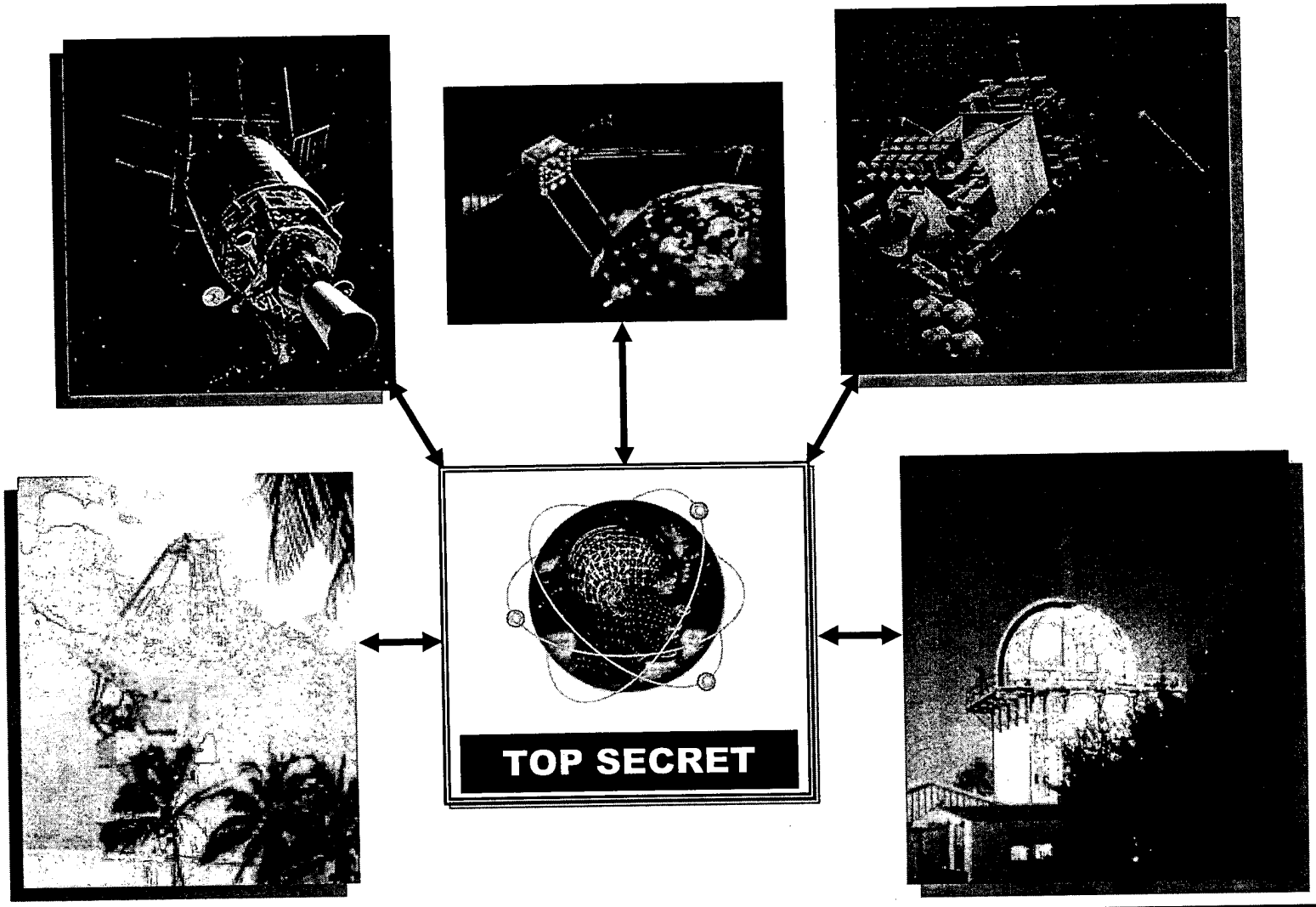
New Applications

IP Clearing House

- Data Fusion
- Multi-dimensional Characterization
- Public Key Inspection

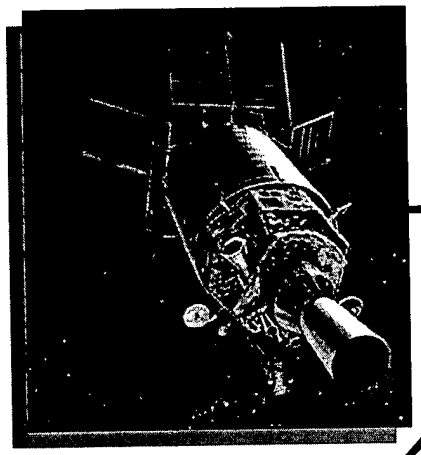
Background

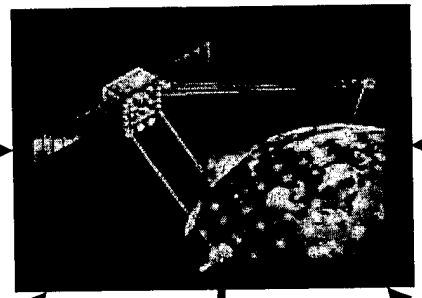
Data Fusion / Multi-source Profiling / Threat Assessment

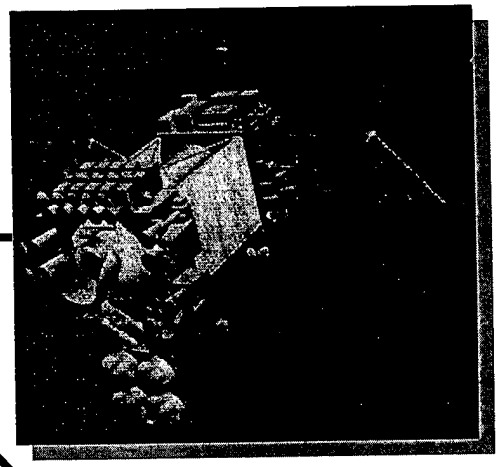


Transform

Data Fusion / Object Characterization

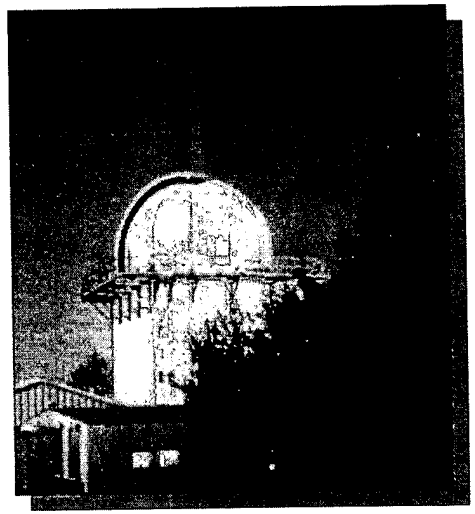






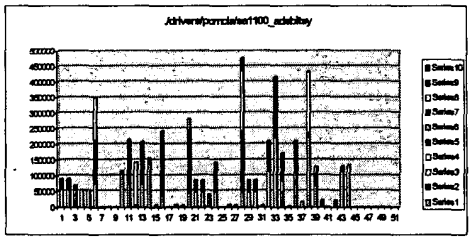
Multi-spectral
Multi-source
Characterization

UNCLASSIFIED



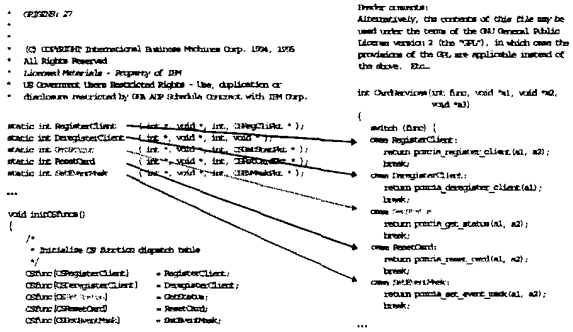
Multi-source Characterization

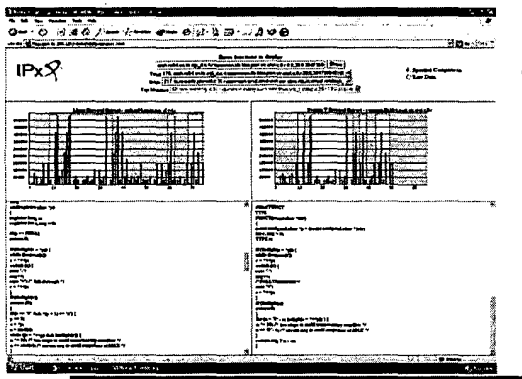
IP Characteristics

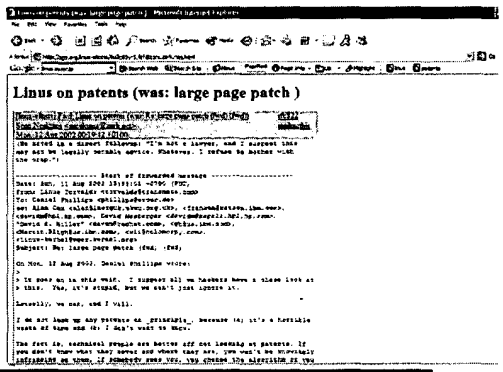


Structure

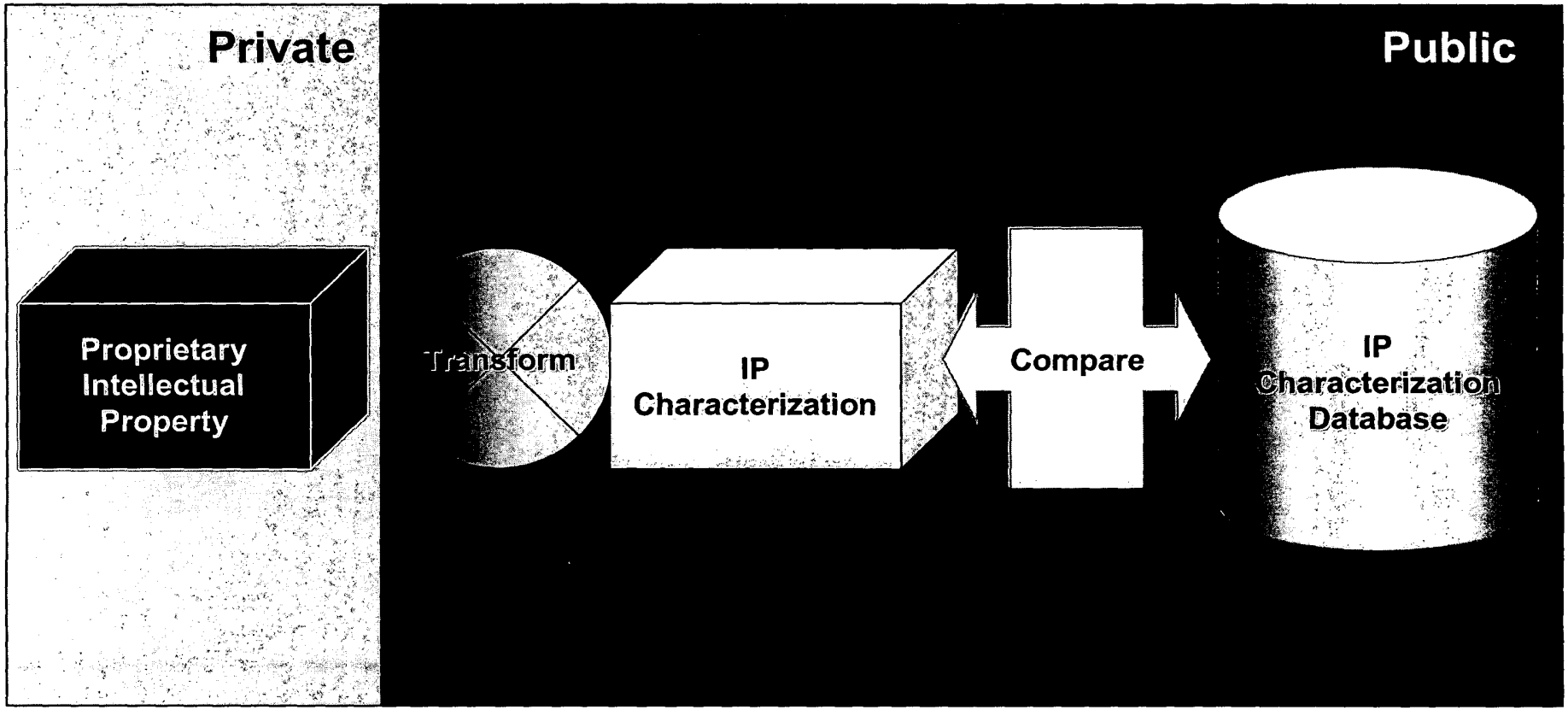
Content	Same	Different
Same	Line-by-line CheckSum	Rarewords
Different	Content Spectra	Author Comments and Spectra







Public Key Inspection™



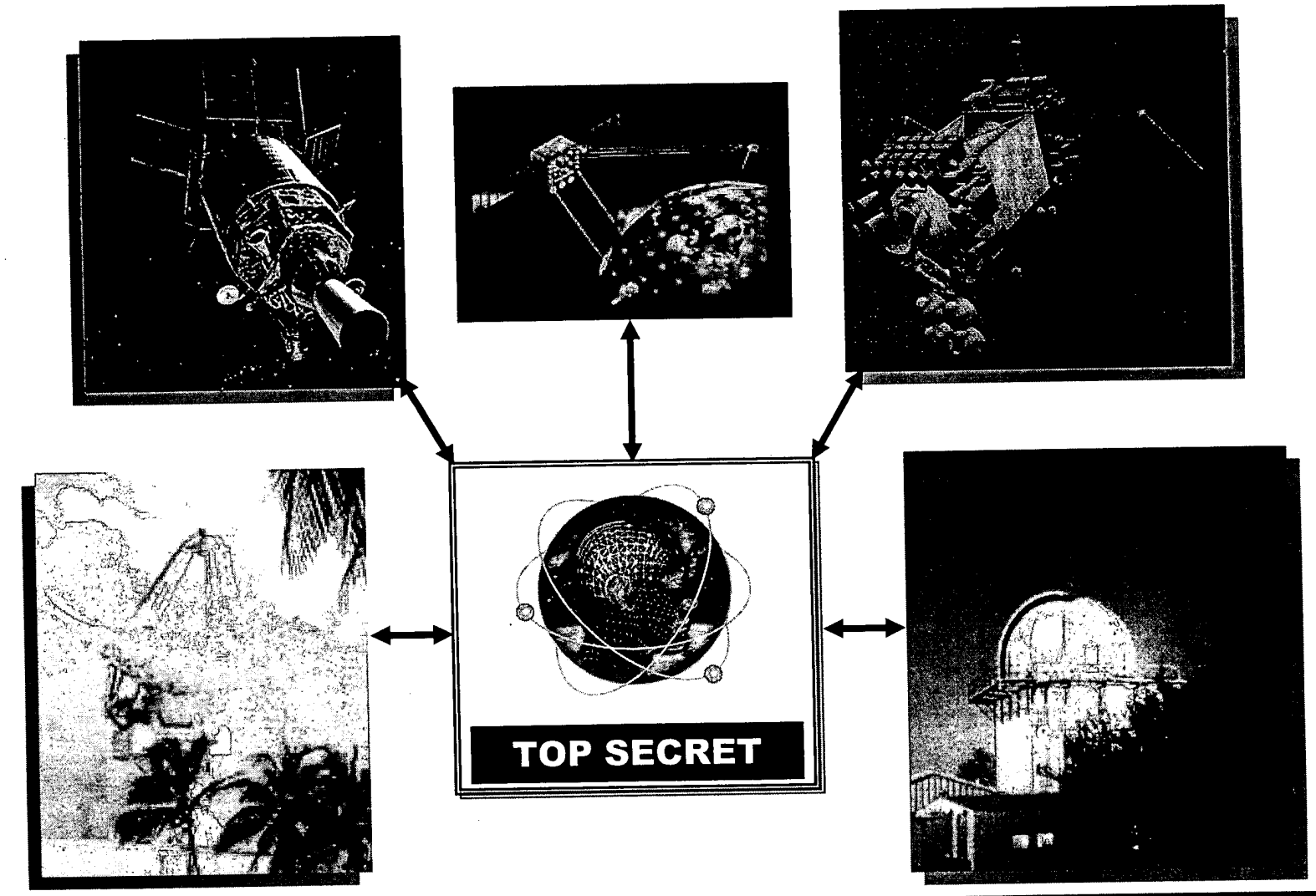
Patent P nding

IP Analysis Tools

- Data Gathering
- Natural Language Processing
- Spectral Analysis
- Pattern Recognition

Background

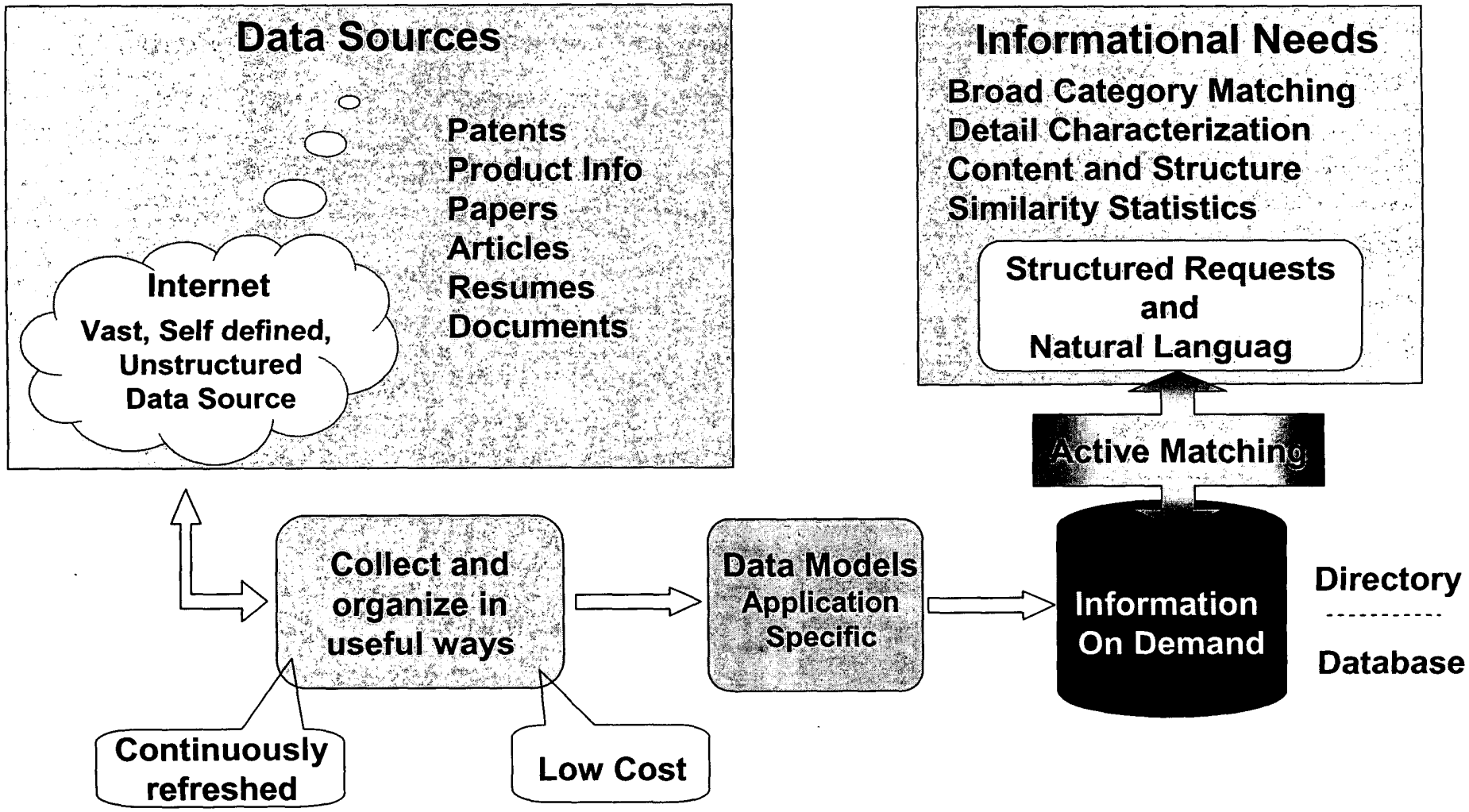
Data Fusion / Multi-source Profiling



Outline

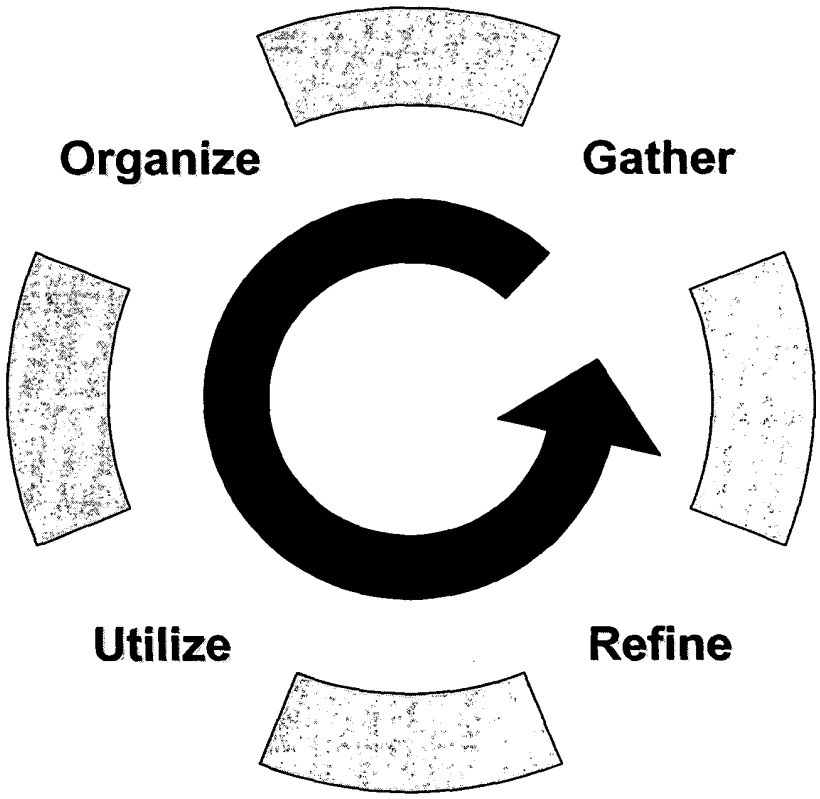
- The SCO IP Model
- Products and Services based on Technology
  - Natural Language Processing
    - Hirehub
    - ZillionResumes

Natural Language Processing

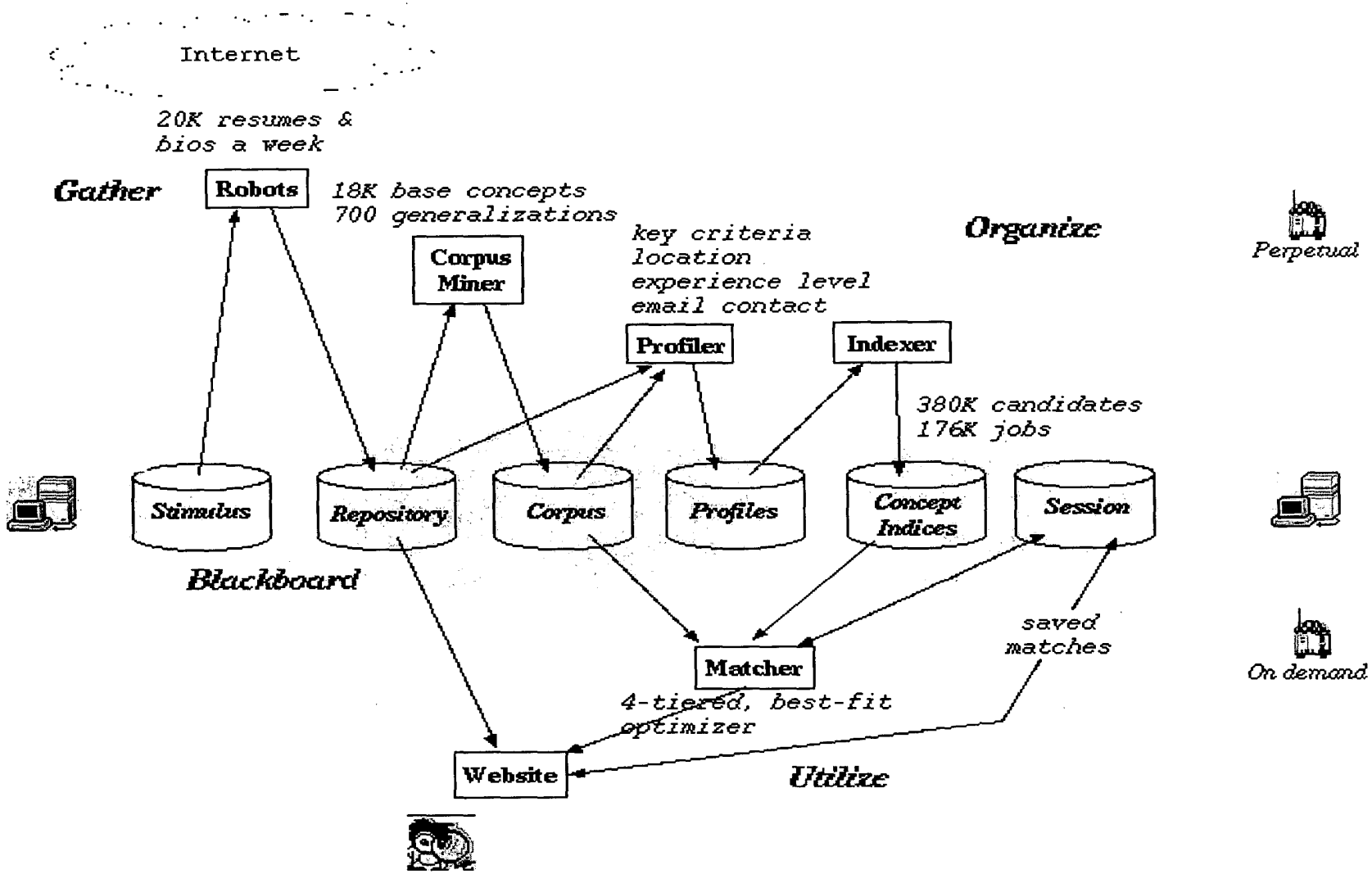


Content Gathering Technology

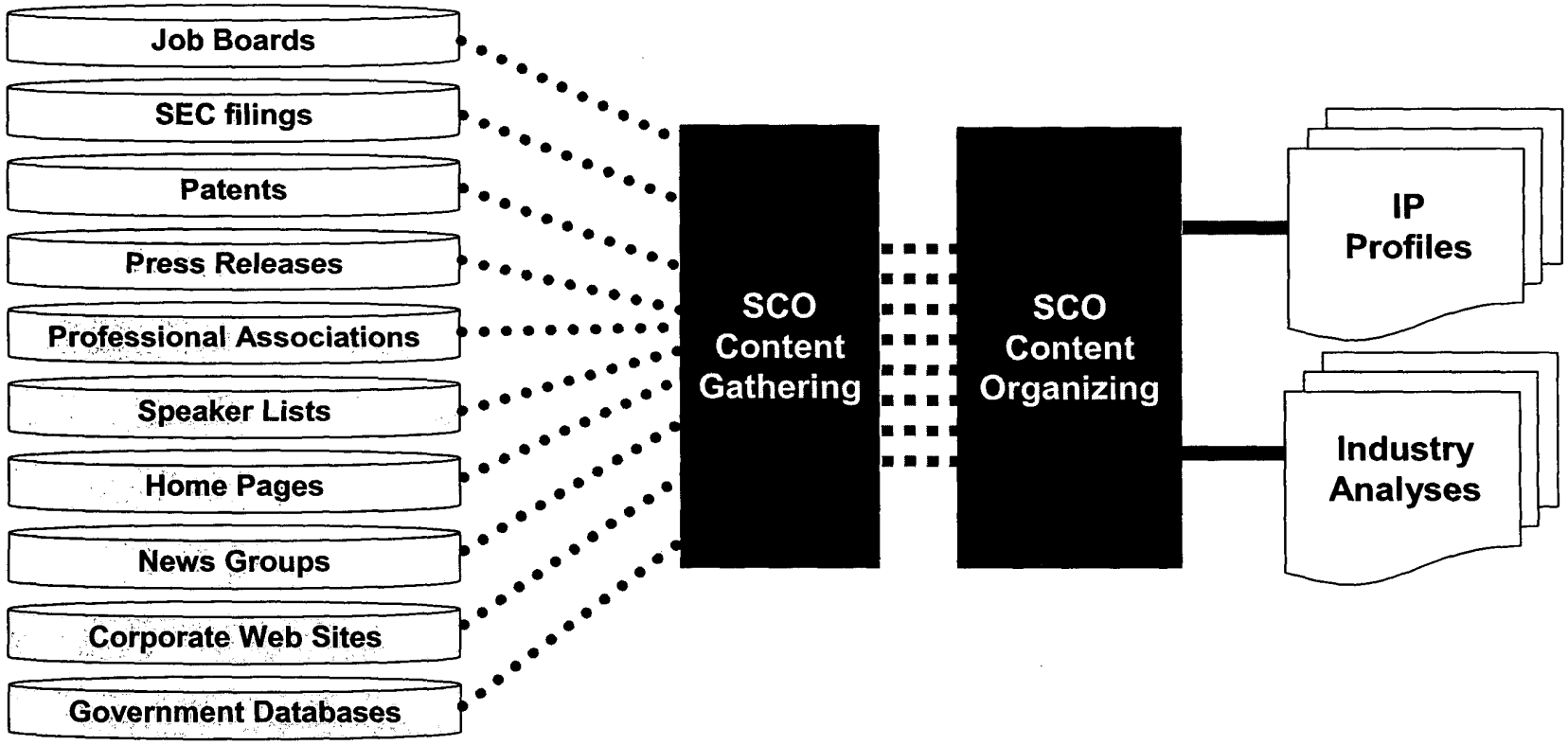
- Gather
  - Goal-driven robots provide focused, deep traversal of web sites
    - No labor required to gather, input, or change
- Organize
  - Robots populate pre-defined topically organized data model
  - Learning robots discover and correlate key concepts, then dynamically define and add new data fields to the data model
- Utilize
  - Create summaries and aggregate information products
  - Apply advanced matching robots to deliver content consistent with user profiles
- Refine
  - Perpetual activity and feedback continuously tune the robots



Deployed Architecture



SCO Information Gathering



The SCO Difference

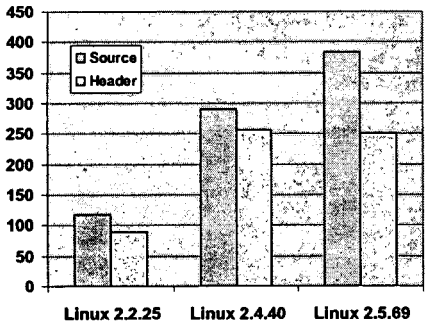
SCO

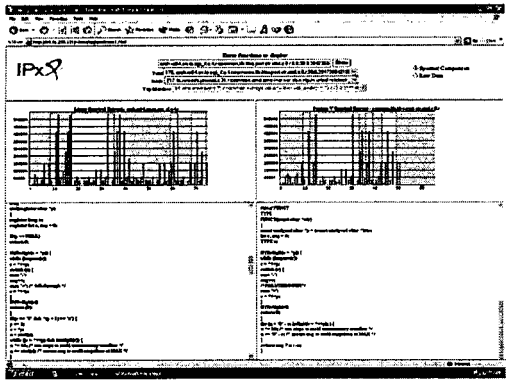
- Demonstrated Technology
  - Gathering and Organizing Large Unstructured Data Sets
    - Resumes
    - Patents
    - Source Code
  - Distributed, scalable, agent based content gathering infrastructure
    - NLP, fact extraction, auto profiling heuristics semantically targeted at industry specific content
    - Deployed applications in e-recruiting, software IP tracking
- Ability to Innovate
  - Merge various traditionally unassociated technologies to provide discontinuous innovation.

Example Analysis Results

Example

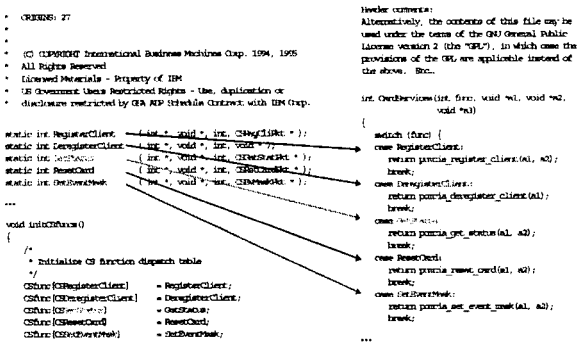
IP Tracing in Source code

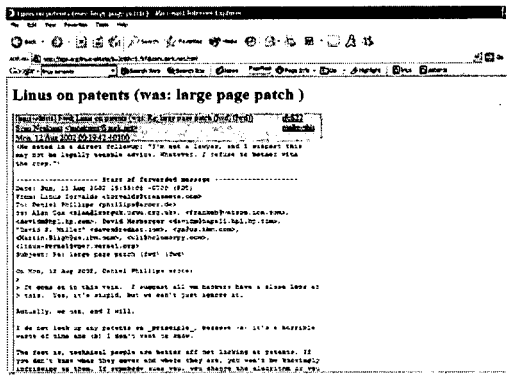




Structure

Content	Same	Different
Same	Shell / Script Grep Diff Comm	Content Analysis Rareword searches
Different	Spectral Analysis Similarity matching	Human Intelligence IP thread analysis





Approach

Concept Dictionaries for Source Code

- Process every file, line by line, in a code base
  - Different dictionary for .c files and for header (.h) files
- Strip away:
  - Comments
  - White spaces
  - C-specific characters like *, &, *, ;, (,)
- Gather concept information:
  - Produce a "raw concept file" which retains the line structure and records the concepts in a dictionary file
  - Tally up the "raw power" of each concept – which is the number of times that concept is used in the entire code base
  - Produce a "raw concept frequency file" for each source file, which records for each line the concepts replaced by their respective powers.
  - Assigns a frequency (or "power number") to every term used in a code file

Approach

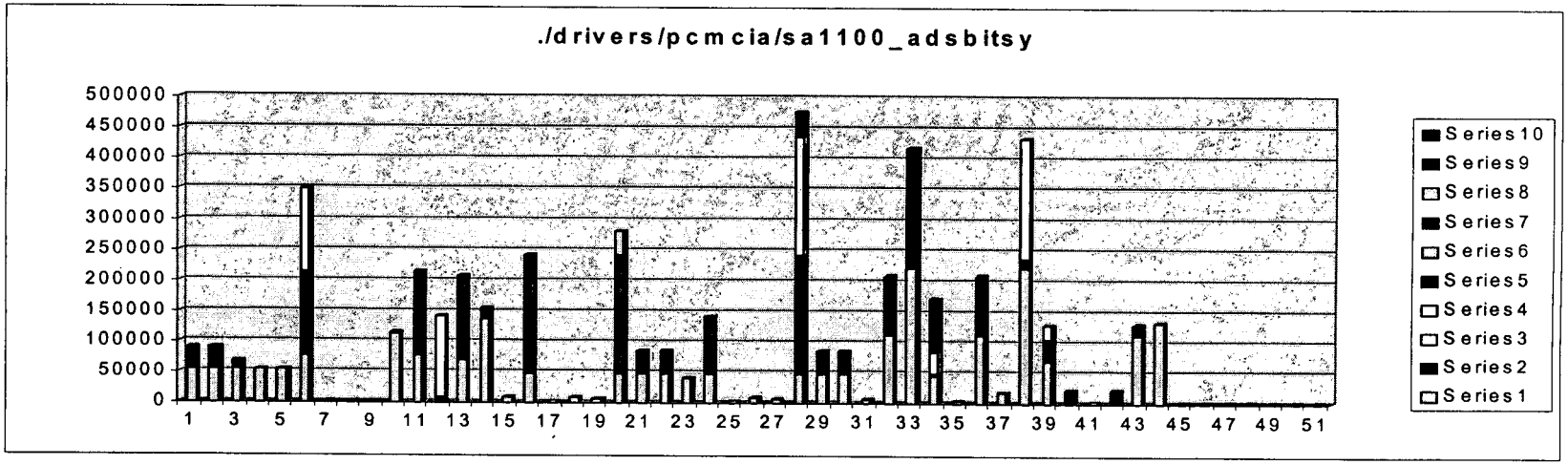
Produce Concept Histograms

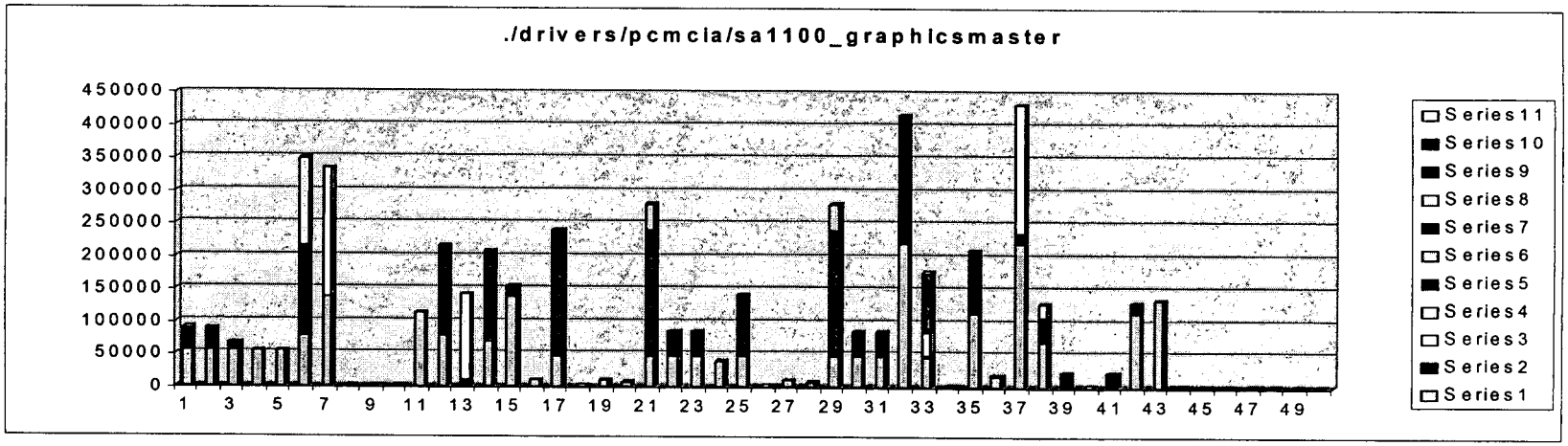
- For each line in a file, translate each concept to the power of that concept from the corpus dictionary
  - Example: a line might look like 2363:12:300:41 (the colons are artificially put in to distinguish different concepts on a line for easy input to a spreadsheet)
- Produce Spectral summary charts
  - A line will contain the name of the file, the number of lines in the file, the number of distinct concepts used in the file, and the total power of the lines in the file
- Can plot and display the file in bar-chart format
  - Horizontal axis is the line number of the file
  - Vertical axis is the total aggregate power of that line from the concept dictionary

Approach

Produce Concept Histograms, examples

- Two histograms from driver files in Linux kernel produced by the same manufacturer – the "spectral" similarity is apparent





Spectral Analysis

Source Code Comparison Window - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites Media

Address http://69.41.235.121/~beta/icp/spectrums1.html Go Links

IPx

Enter functions to display

arch.ia64.sn.io.sgi_if.c.4.r:common.lib.libc.port.str.atol.c.0.r:6:6:30:0.3947368 Show

Total 178. arch.ia64.sn.io.sgi_if.c.4.r:common.lib.libc.port.str.atol.c.0.r:30:0.39473684210

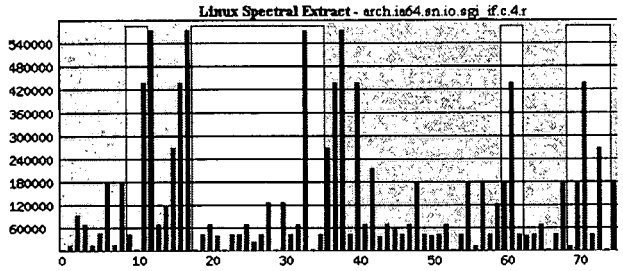
fs/xfs 217. fs.reiserfs.journal.c.31.r:common.cmd.cmd-inet.usr.sbin.ntp.in.xntpd.refclock_

Top Matches 98. mm.memory.c.31.r:common.subsys.udi.env.linux.udi_osdep.c.35.r:15:0.31914

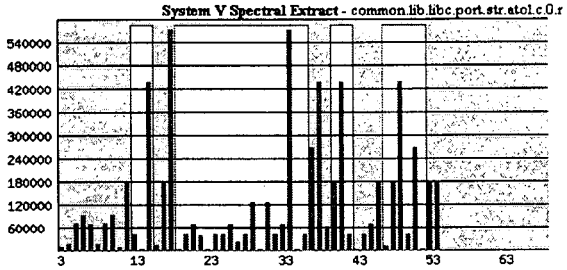
Spectral Comparison

Raw Data

Linux Spectral Extract - arch.ia64.sn.io.sgi_if.c.4.r



System V Spectral Extract - common.lib.libc.port.str.atol.c.0.r



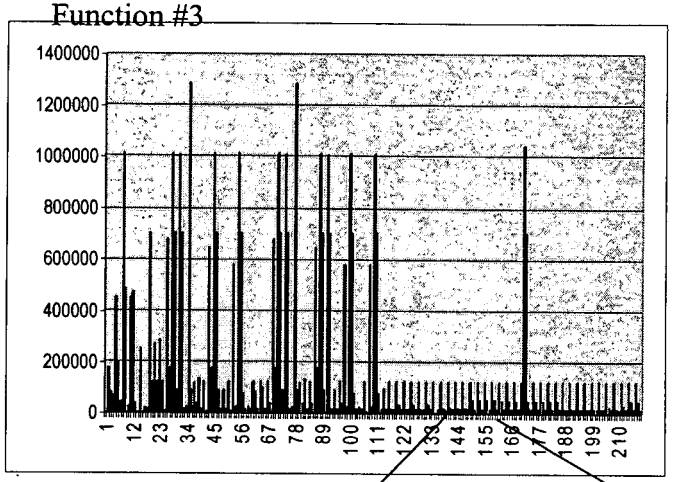

```
long
atol(register char *p)
{
register long n;
register int c, neg = 0;

if (p == NULL)
return 0;

if (!isdigit(c = *p)) {
while (isspace(c))
c = *++p;
switch (c) {
case '-':
neg++;
case '+': /* fall-through */
c = *++p;
}
if (!isdigit(c))
return (0);
}
if (c == '0' && *(p + 1) == 'x') {
p += 2;
c = *p;
n = xtod(c);
while ((c = *++p) && isxdigit(c)) {
n *= 16; /* two steps to avoid unnecessary overflow */
n += xtod(c); /* accum neg to avoid surprises at MAX */
}
```

```
#ifdef FUNCT
TYPE
FUNCT(const char *str)
{
const unsigned char *p = (const unsigned char *)str;
int c, neg = 0;
TYPE n;

if (!isdigit(c = *p)) {
while (isspace(c))
c = *++p;
switch (c) {
case '-':
neg++;
/*FALLTHROUGH*/
case '+':
c = *++p;
}
if (!isdigit(c))
return 0;
}
for (n = '0' - c; isdigit(c = *++p); ) {
n *= 10; /* two steps to avoid unnecessary overflow */
n += '0' - c; /* accum neg to avoid surprises at MAX */
}
return neg ? n : -n;
}
```

Internet

start Source Code Compari... Microsoft PowerPoint ... 12:13 AM

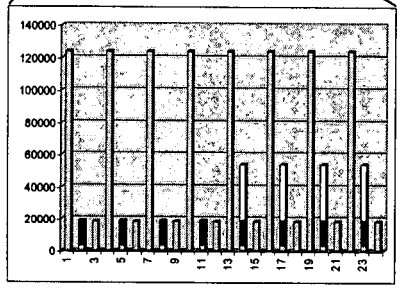
SystemV File: common.uts.net.pf.pf.c
Function #3



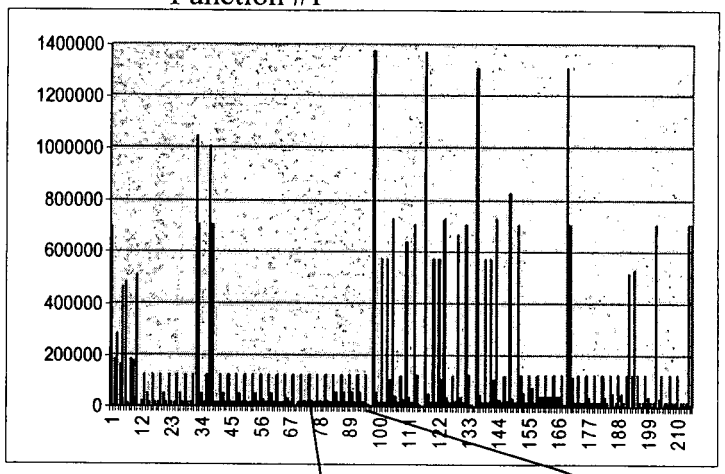
Code Extract

```
case BPF_JMP|BPF_JGT|BPF_K:
        pc += (A > pc->k) ? pc->jt : pc->jf;
        continue;

case BPF_JMP|BPF_JGE|BPF_K:
        pc += (A >= pc->k) ? pc->jt : pc->jf;
        continue;

case BPF_JMP|BPF_JEQ|BPF_K:
        pc += (A == pc->k) ? pc->jt : pc->jf;
        continue;

case BPF_JMP|BPF_JSET|BPF_K:
        pc += (A & pc->k) ? pc->jt : pc->jf;
        continue;

case BPF_JMP|BPF_JGT|BPF_X:
        pc += (A > X) ? pc->jt : pc->jf;
        continue;

case BPF_JMP|BPF_JGE|BPF_X:
        pc += (A >= X) ? pc->jt : pc->jf;
        continue;

case BPF_JMP|BPF_JEQ|BPF_X:
        pc += (A == X) ? pc->jt : pc->jf;
        continue;

case BPF_JMP|BPF_JSET|BPF_X:
        pc += (A & X) ? pc->jt : pc->jf;
        continue;
```

Spectral Extract



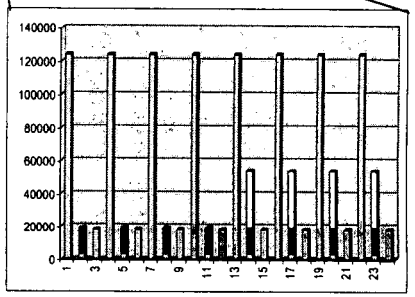
Linux File :net.core.filter.c
Function #1



Code Extract

```
case BPF_JMP|BPF_JGT|BPF_K:
        pc += (A > fentry->k) ? fentry->jt : fentry->jf;
        continue;

case BPF_JMP|BPF_JGE|BPF_K:
        pc += (A >= fentry->k) ? fentry->jt : fentry->jf;
        continue;

case BPF_JMP|BPF_JEQ|BPF_K:
        pc += (A == fentry->k) ? fentry->jt : fentry->jf;
        continue;

case BPF_JMP|BPF_JSET|BPF_K:
        pc += (A & fentry->k) ? fentry->jt : fentry->jf;
        continue;

case BPF_JMP|BPF_JGT|BPF_X:
        pc += (A > X) ? fentry->jt : fentry->jf;
        continue;

case BPF_JMP|BPF_JGE|BPF_X:
        pc += (A >= X) ? fentry->jt : fentry->jf;
        continue;

case BPF_JMP|BPF_JEQ|BPF_X:
        pc += (A == X) ? fentry->jt : fentry->jf;
        continue;

case BPF_JMP|BPF_JSET|BPF_X:
        pc += (A & X) ? fentry->jt : fentry->jf;
        continue;
```

Spectral Extract



IP Tracing

Examples of Rare Words found with search

- **Consider how this method was re-factored from AIX**

Linux Source

```
Header comments:
Alternatively, the contents of this file may be
used under the terms of the GNU General Public
License version 2 (the "GPL"), in which case the
provisions of the GPL are applicable instead of
the above.  Etc…

int CardServices(int func, void *a1, void *a2,
                 void *a3)
{
   switch (func) {
   case RegisterClient:
      return pcmcia_register_client(a1, a2);
      break;
   case DeregisterClient:
      return pcmcia_deregister_client(a1);
      break;
   case GetStatus:
      return pcmcia_get_status(a1, a2);
      break;
   case ResetCard:
      return pcmcia_reset_card(a1, a2);
      break;
   case SetEventMask:
      return pcmcia_set_event_mask(a1, a2);
      break;
…
```

AIX Source

```
*   ORIGINS: 27
*
*
*   (C) COPYRIGHT International Business Machines Corp. 1994, 1995
*   All Rights Reserved
*   Licensed Materials - Property of IBM
*   US Government Users Restricted Rights - Use, duplication or
*   disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

static int RegisterClient      ( int *, void *, int, CSRegCliPkt * );
static int DeregisterClient    ( int *, void *, int, void * );
static int GetStatus           ( int *, void *, int, CSGetStatPkt * );
static int ResetCard           ( int *, void *, int, CSRstCardPkt * );
static int SetEventMask        ( int *, void *, int, CSEvMaskPkt * );

…

void initCSfuncs()
{
    /*
     * Initialize CS function dispatch table
     */
    CSfunc[CSRegisterClient]      = RegisterClient;
    CSfunc[CSDeregisterClient]    = DeregisterClient;
    CSfunc[CSGetStatus]           = GetStatus;
    CSfunc[CSResetCard]           = ResetCard;
    CSfunc[CSSetEventMask]        = SetEventMask;
…
```

The methods are implemented differently, but provide the SAME function, use the SAME variable names in the SAME order.