Harvest: A Scalable, Customizable Discovery and Access System

IRTF-RD:

Mic Bowman	(co-PI, Transarc)
Peter Danzig	(co-PI, USC)
Darren Hardy	(PRA, U Colorado)
Udi Manber	(co-PI, U Arizona)
Mike Schwartz	(PI, U Colorado)

Primary support: ARPA
Additional support: AFOSR, Hughes, NSF, Sun

1. Introduction
2. Harvest Approach
3. Features & Demo Brokers
4. Subsystems & Measurements
5. Summary

1. Introduction

Progression of Publishing Tools:

-1991	FTP, NetNews
1992	Gopher, WAIS
1993-	WWW/Mosaic

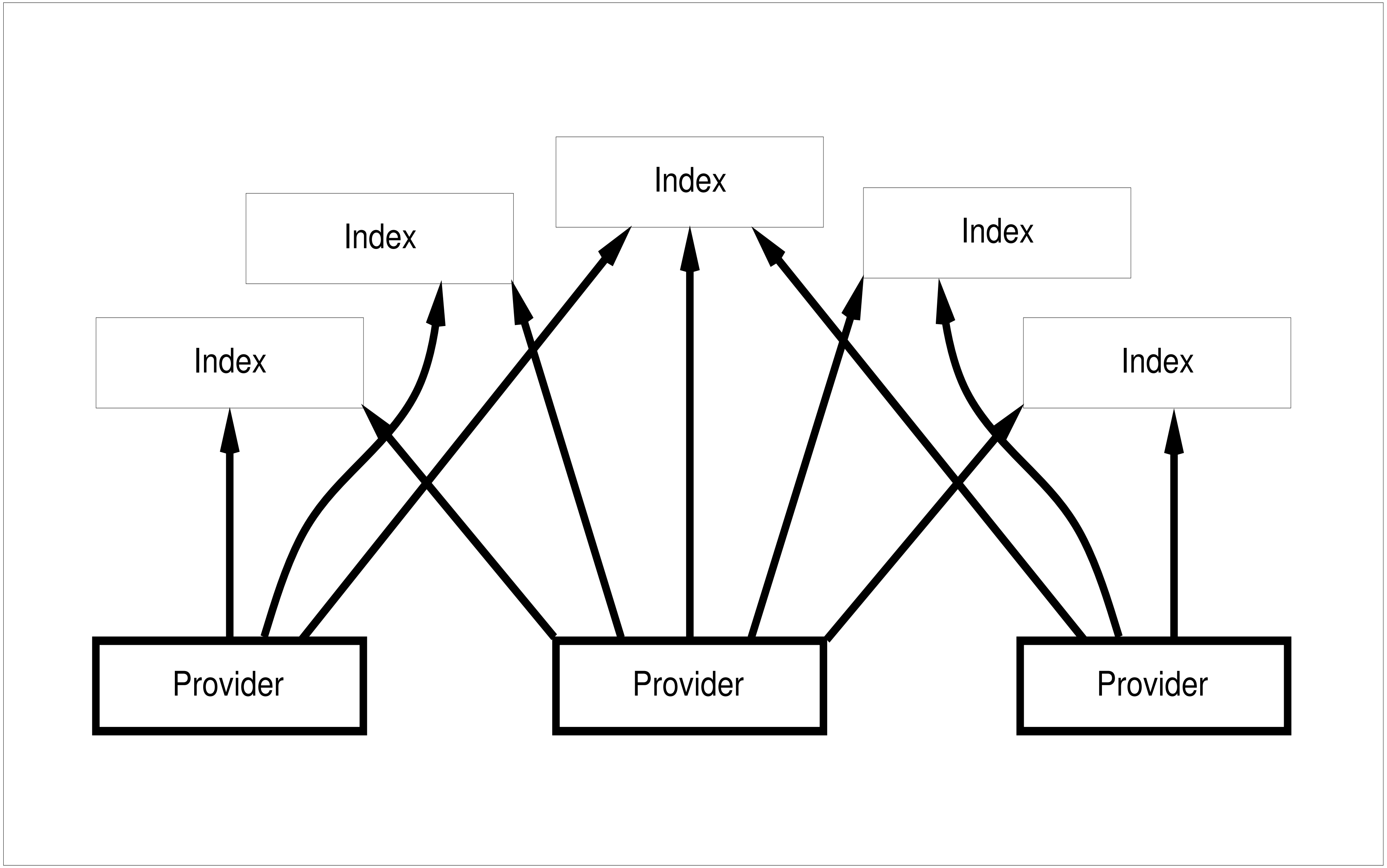
Publishing now easy, but *effective use* difficult:

- can’t find relevant information
- server and network bottlenecks
- little support for structured, complex data

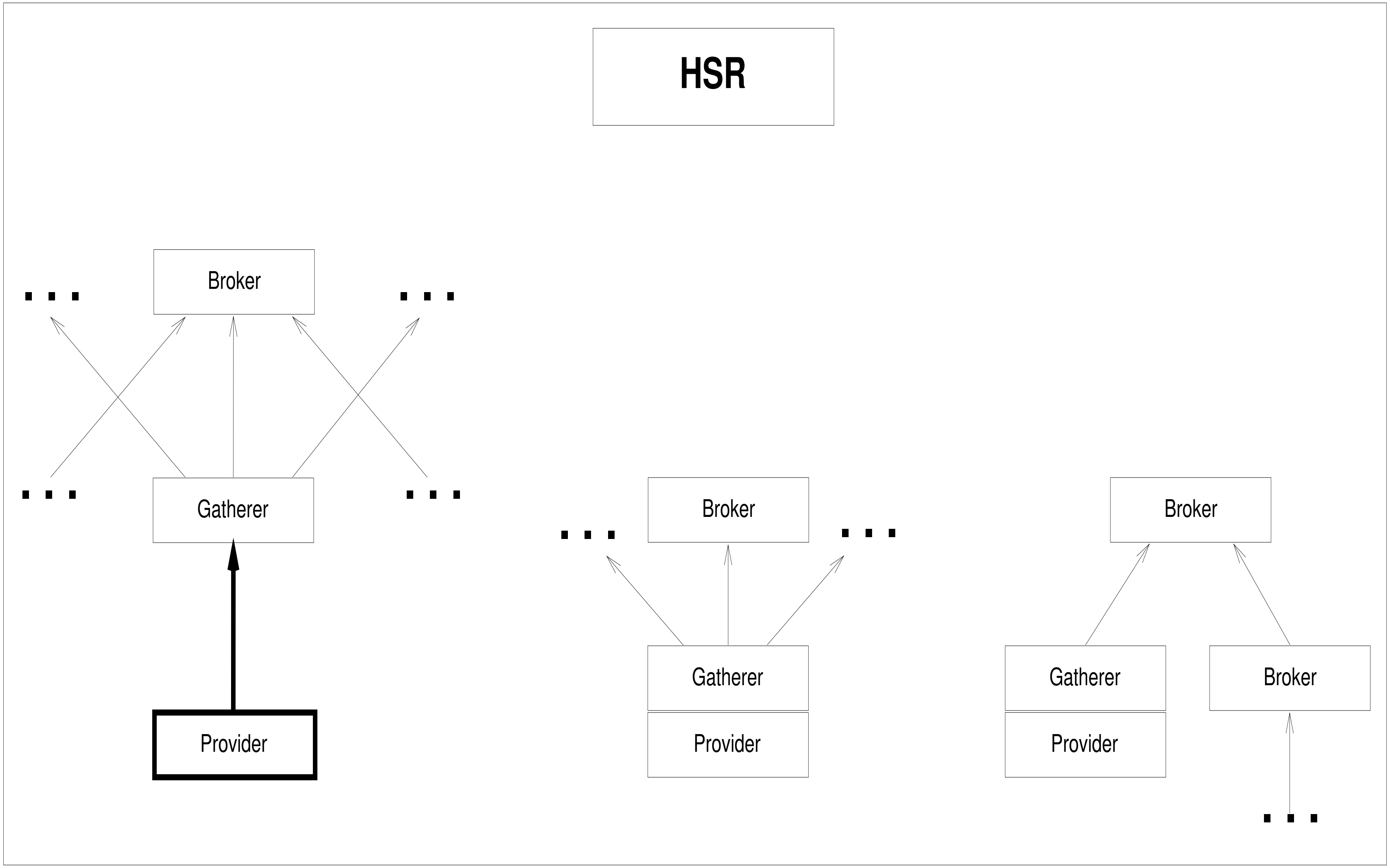
2. Harvest Approach

- Efficient index gathering & distribution architecture
- Customizable information extraction/indexing/search tools
- Structured content summary interchange protocol
- Topology-adaptive server replication
- Hierarchical object caching

Uncoordinated index gathering:



Harvest index gathering:



Gatherer specification:

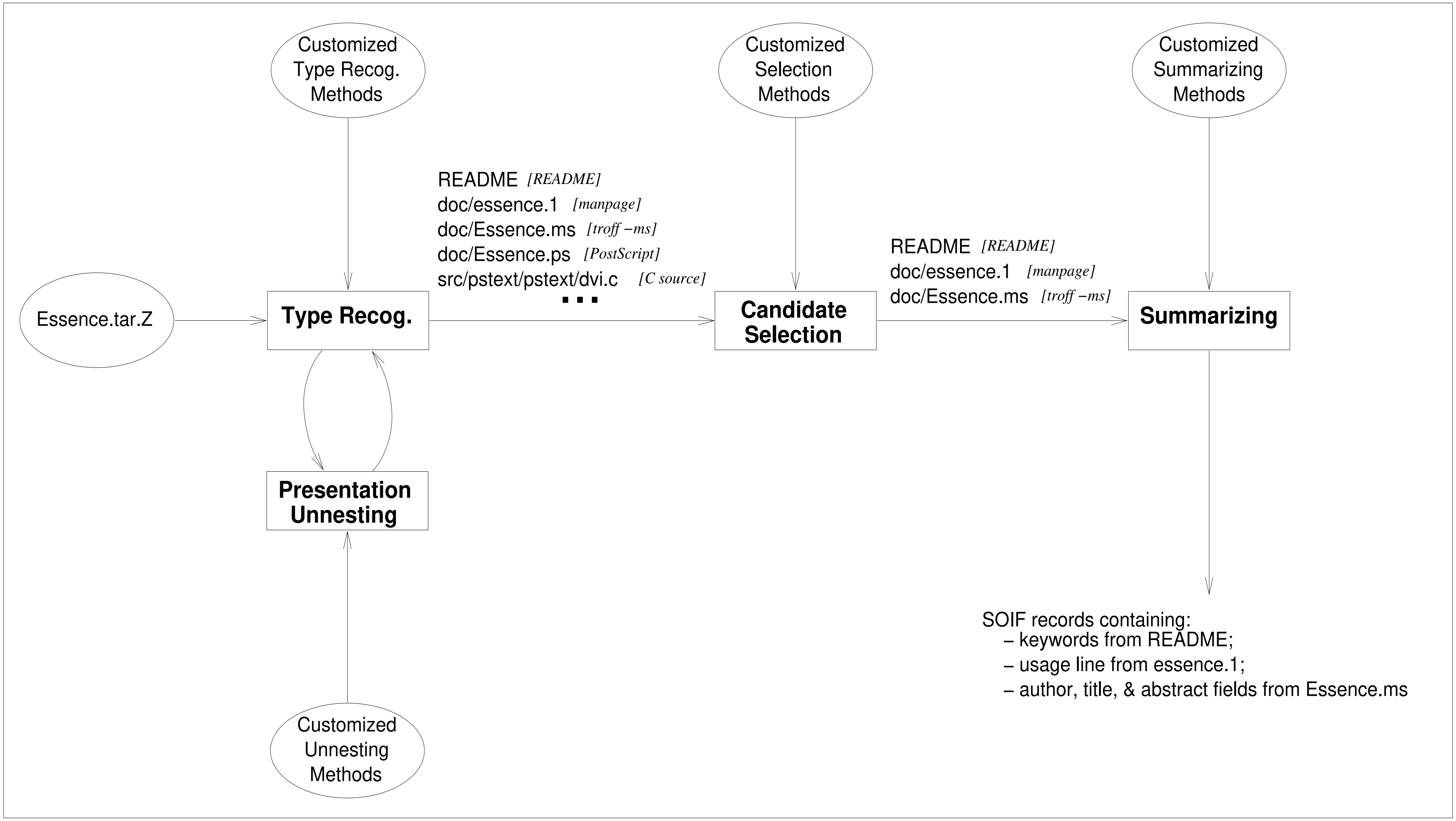
```
<RootNodes>
ftp://ftp.cs.colorado.edu/pub/cs/techreports/hardy
http://rd.cs.colorado.edu/harvest
</RootNodes>

<LeafNodes>
http://powell.cs.colorado.edu/~hardy/Home.html
ftp://ftp.cs.colorado.edu/pub/cs/distribs/essence/Essence.tar.Z
</LeafNodes>
```

Resulting actions:

- enumeration
- retrieval
- unnesting
- type-specific summarizing
- content summary serving

Customizable content extraction (Essence):



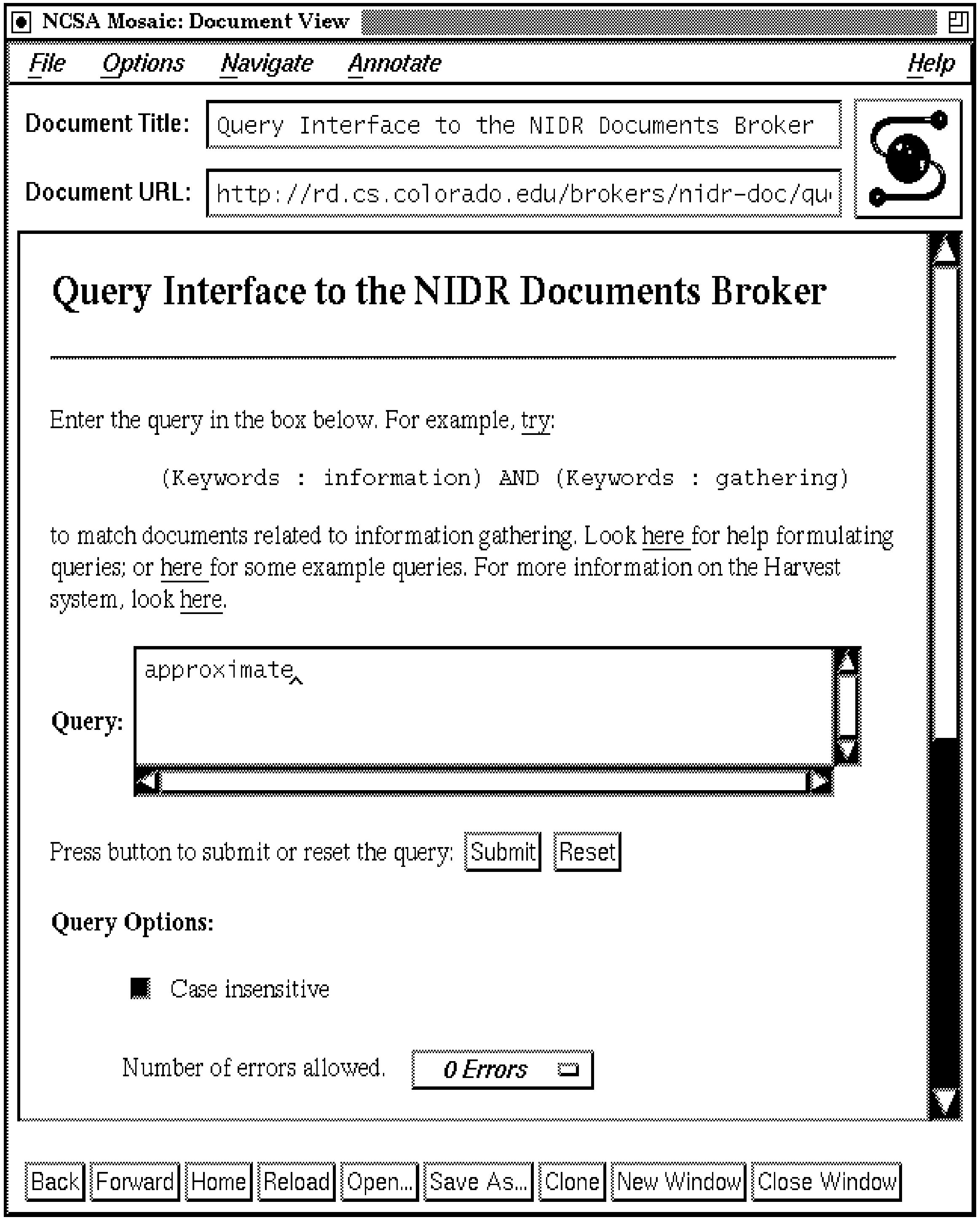
Summary Object Interchange Format (SOIF):

- Gatherer-Broker protocol
- structured object summary stream
- arbitrary binary data + extensible fields
- type + URL + set of attribute-value pairs
- easy interoperation with other standards

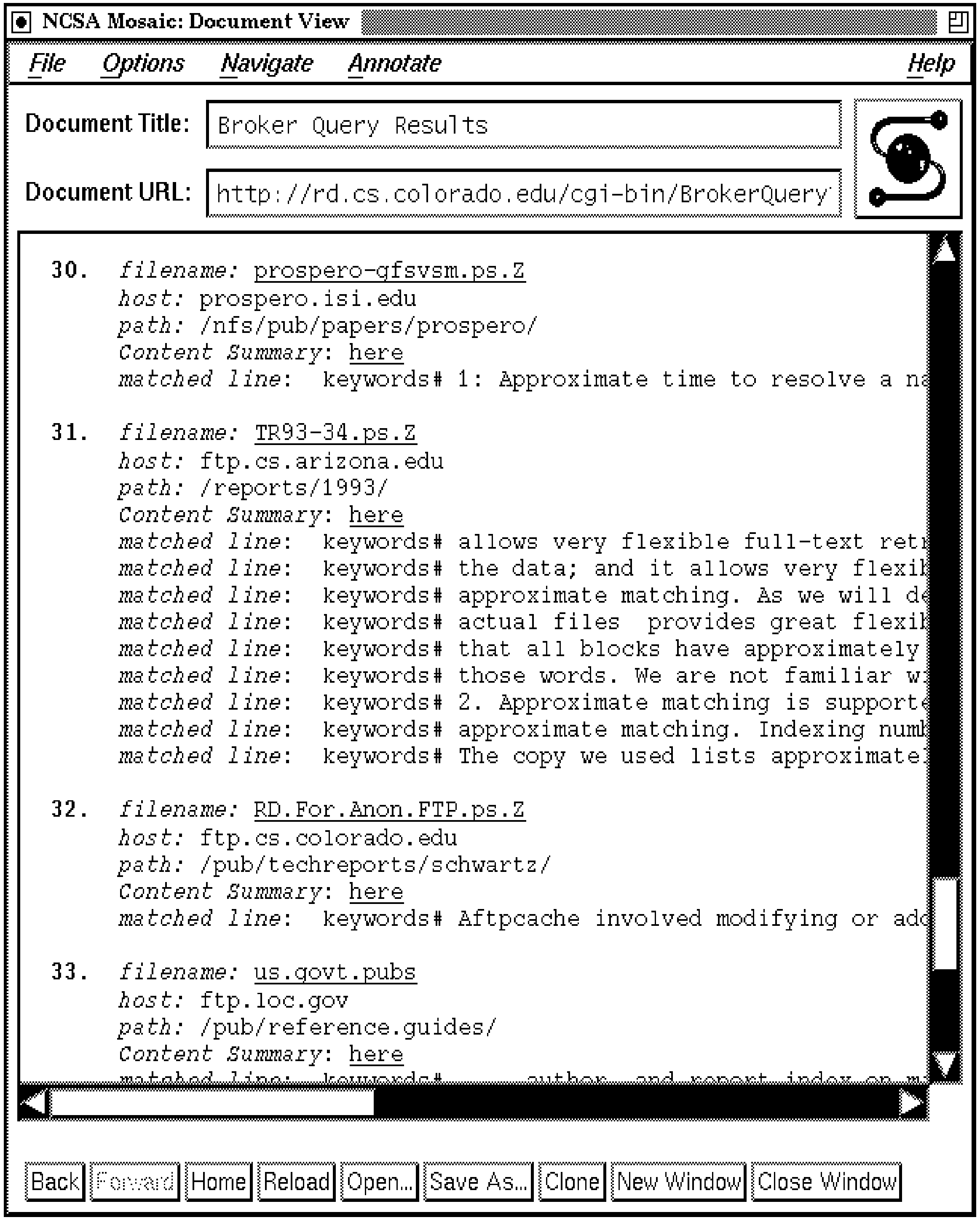
Example:

```
@DOCUMENT{ ftp://ftp.cs.psu.edu/pub/techreps/TR123.ps
Title{28}: Type Structured File Systems
Author-Name{33}: C. Dharap, R. Balay and M. Bowman
Author-Organization-Type{11}: educational
Author-Department{32}: Computer Science and Engineering
Publication-Status{19}: Accepted, To appear.
Keywords{42}: types object-oriented semantic filesystems
Access-Protocol-v*{3}: FTP
Description{457}: The paper describes the design of Nebula.  Nebula is a dynamic, object based
        typed file system.  Nebula uses types to create well structured files as well as to export
        existing files in the internet environment.  This allows for backward scalability.  Files in
        Nebula are abstractions and are defined by objects.  Nebula assumes a flat global
        namespace and operations are provided to manipulate logical namespaces.
}
```

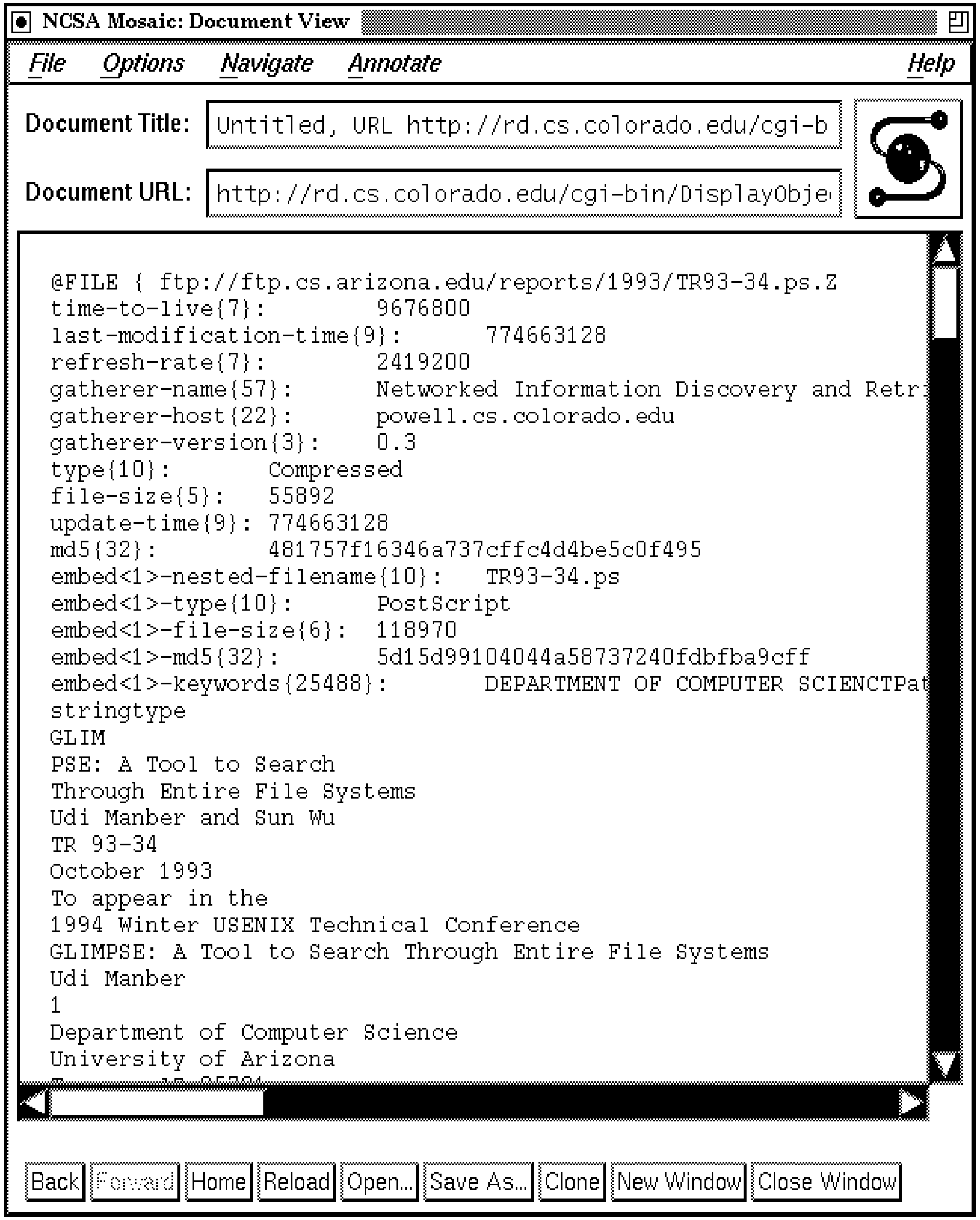
Example query:



Example result:



Example content summary:


NCSA Mosaic: Document View

File Options Navigate Annotate Help

Document Title: Untitled, URL http://rd.cs.colorado.edu/cgi-b

Document URL: http://rd.cs.colorado.edu/cgi-bin/DisplayObje

```
@FILE { ftp://ftp.cs.arizona.edu/reports/1993/TR93-34.ps.Z
time-to-live{7}:        9676800
last-modification-time{9}:      774663128
refresh-rate{7}:        2419200
gatherer-name{57}:      Networked Information Discovery and Retr
gatherer-host{22}:      powell.cs.colorado.edu
gatherer-version{3}:    0.3
type{10}:       Compressed
file-size{5}:   55892
update-time{9}: 774663128
md5{32}:        481757f16346a737cffc4d4be5c0f495
embed<1>-nested-filename{10}:   TR93-34.ps
embed<1>-type{10}:      PostScript
embed<1>-file-size{6}:  118970
embed<1>-md5{32}:       5d15d99104044a58737240fdbfba9cff
embed<1>-keywords{25488}:       DEPARTMENT OF COMPUTER SCIENCTPat
stringtype
GLIM
PSE: A Tool to Search
Through Entire File Systems
Udi Manber and Sun Wu
TR 93-34
October 1993
To appear in the
1994 Winter USENIX Technical Conference
GLIMPSE: A Tool to Search Through Entire File Systems
Udi Manber
1
Department of Computer Science
University of Arizona
```

Back Forward Home Reload Open... Save As... Clone New Window Close Window

3. Features & Demo Brokers

Feature	Demo Brokers	#Objects	#Sites
Topic-specific indexing; Broker cascading	NIDR-SW, NIDR-Doc, NIDR	635	80
Scalable content indexing; Multiple formats; Gathering distribution	CS Technical Reports	18,000	217
Structured indexing; Indexing customizability	SFI Time Series Papers	32	4
Support for other standards; Incorp. manual indexing info	PC Software Packages	25,000	8
Focused global content index vs. exhaustive name index	WWW Home Pages	5,000	1,761
Locate Harvest servers during search & index config	Harvest Server Registry	19	3

Implementing Other Systems with Harvest

Archie, Veronica, WWWW, etc.
Gatherer configuration + Essence extraction script

Content Router
WAIS enumerator + Essence extraction script

WAIS
Essence full text extraction + ranking search engine

WebCrawler
Gatherer configuration + Essence full text extraction

WHOIS++
Gatherer configuration + Essence extraction script for centroids. Query front-end to search SOIF records

4. Subsystems & Measurements

- Gatherer †
- Broker
- Index & Search †
- Replicator
- Object Cache †

Gatherer Subsystem

Server load efficiency:

- scans objects periodically & builds cache of indexing info
- allows indexing info. to be retrieved in single stream, rather than per-object requests
- local Gatherers vs. uncoordinated gathering:
  6,600x server load savings *per indexer* (on average)

Network traffic efficiency:

- uses Essence to pre-summarize data
- supports incremental updates
- compresses response stream
- local Gatherers vs. uncoordinated gathering:
  59x network transmission savings (example)

Index and Search Subsystem

- General Broker-Indexer interface to accommodate variety of index/search engines
- Can use many backends (WAIS, Ingres, etc.)
- Implemented two specialized backends:
  - Glimpse for space-efficiency + flexible interactive queries
  - Nebula for time-efficiency + complex standing queries

Glimpse:

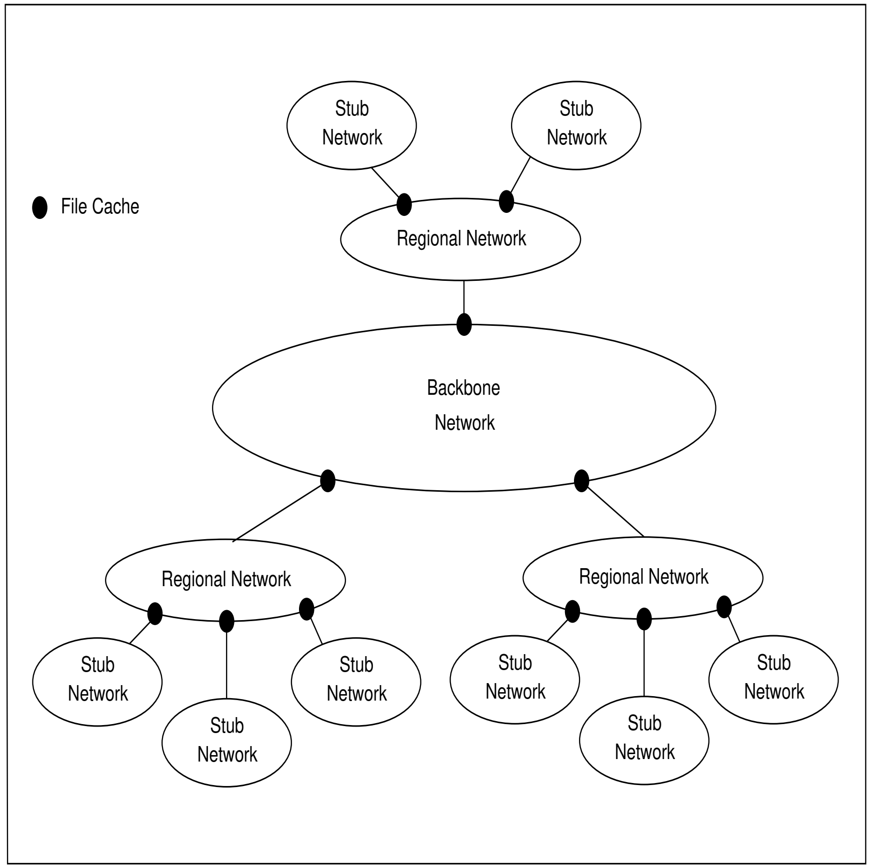
- pointers to adjustable-sized occurrence blocks $\rightarrow$ index is 2-12% size of data (vs. 100+% w/ WAIS)
- booleans, REs, approximate searches
- incremental indexing
- indexing measurements for 6000 files (70MB) on SparcStation IPC:
  - incremental indexing in 2-4 min.
  - complete indexing in 20 minutes
- searches take several seconds (time/space tradeoff)

Nebula:

- each object = set of attribute/value pairs
- separate index (and possibly mechanism) per attribute tag
- view notion: standing query against object base
- easy to refine queries & observe changes interactively
- bigger indexes than Glimpse but 100 msec searches (time/space tradeoff)

Object Caching Subsystem

- Caches Gopher, HTTP, and FTP objects
- Uses Mosaic proxy interface
- TTL-based coherence

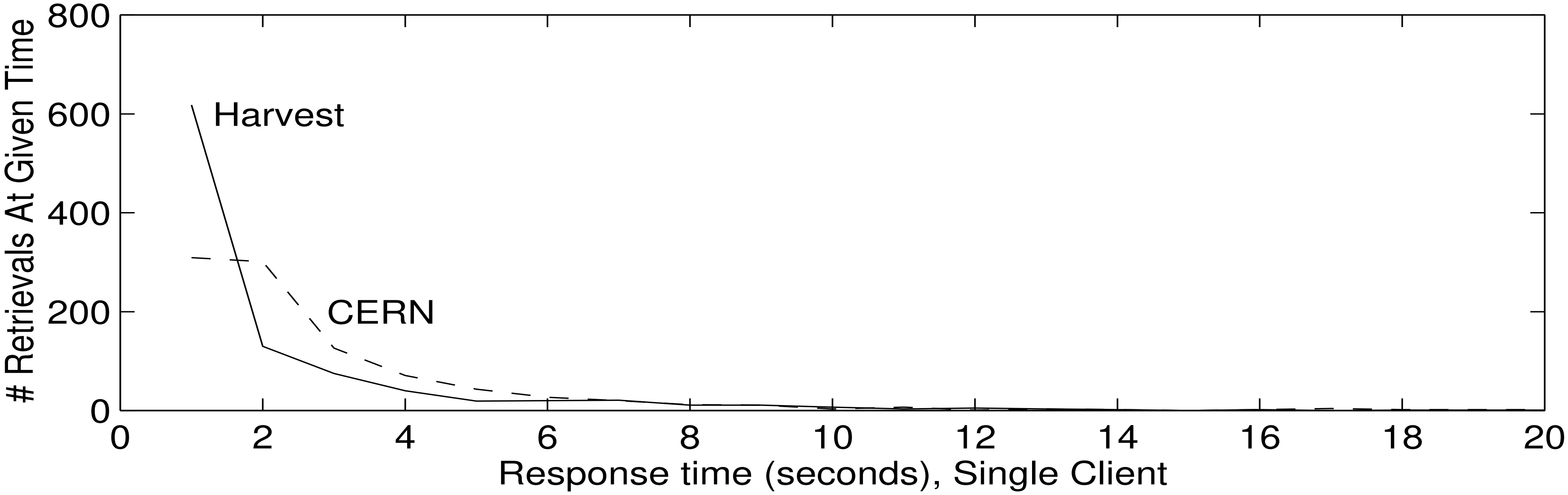


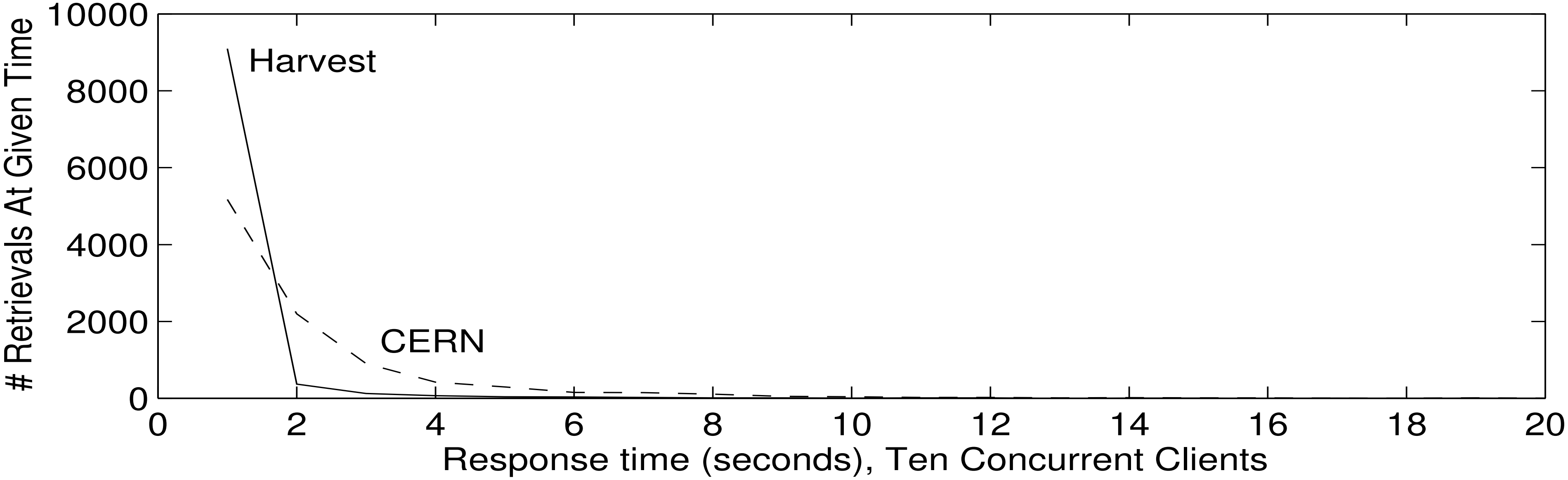
Object location protocol:

- "query" datagrams to each neighbor and parent, plus inetd echo to object’s home site
- choose fastest responding "yes"

Implementation optimizations:

- single, event-driven process
- non-blocking I/O to disk and to cache clients
- concurrent source-cache and cache-client transfers
- separately manages replacement on disk and in VM
- keeps meta-data in VM
- DNS name-to-address cache


800
600
400
200
0
Harvest
CERN
Retrievals At Given Time
0
2
4
6
8
10
12
14
16
18
20
Response time (seconds), Single Client


10000
8000
6000
4000
2000
0
Harvest
CERN
Retrievals At Given Time
0
2
4
6
8
10
12
14
16
18
20
Response time (seconds), Ten Concurrent Clients

Still needed:

- coordinating deployment
- auto-configuring clients
- standard for indicating TTLs

5. Summary

- Simple Gatherer spec $\rightarrow$ enumeration, retrieval, unnesting, customized summarizing, content summary serving
- Provider-site Gatherers: huge server & network load reduction
- Broker-Indexer interface accommodates multiple index/search engines, including
  - Glimpse: flexible search, space efficient
  - Nebula: views + query refinement, fast
- Very fast hierarchical Object Cache
- Topology-adaptive replication
- SOIF easily interoperates with other standards
- Half dozen indexes illustrate scalability & customizability

More information (including demos & paper):
http://rd.cs.colorado.edu/harvest/

Soliciting (source code) beta test sites for:

- Gatherers, partic. at large archive sites
- topic-specific Brokers
- Caches at backbones, regionals, and stub nets

email   harvest-dvl@cs.colorado.edu

v1.0 general release around end of summer 1994