

Technical aspects of the open-search project

Robin Gareus

December 12, 2006

- 1 Introduction
overview
- 2 Technical Aspects
abstraction models
technical details
impact of technical decisions on design
- 3 Outlook
community
possible points of failure
ideas and discussion

requirements

Summary of goals to start off open-search:

- open and free: open source, cross platform
- quick search query execution
- consistent and up-to date search results
- scalable to $> 10^{10}$ documents in the index
- secure and reliable
- have a technical proposal by end of Jan 2007

Robin Gareus

Outline

Introduction
overview

Technical
Aspects

Outlook

- identify tasks into building blocks
- categorize and specify interfaces
- build "chains"

basic structure

Outline

Introduction

Technical
Aspects

**abstraction
models**

technical details
impact of
technical
decisions on
design

Outlook

- user interface
- query parser, result generator
- crawler, sniffer
- data indexer, merger
- network interface

Robin Gareus

Outline

Introduction

Technical
Aspects

**abstraction
models**

technical details
impact of
technical
decisions on
design

Outlook

network layer

Outline

Introduction

Technical
Aspects

abstraction
models

technical details
impact of
technical
decisions on
design

Outlook

- distributed Hash Tables on top of TCP
- routing via TCP/IP not on top of it.
- compromise: different channels / networks for different purpose:
 - high speed low latency query interface
 - mid speed query replies
 - low QoS data exchange, maintenance

data formats

Outline

Introduction

Technical
Aspects

abstraction
models

technical details

impact of
technical
decisions on
design

Outlook

- XML on user level
- hashed indices, data pointers, links.
- binary compatible data link for distributed hash tables on top of TCP/IP

content parsing and data indexing

Technical
aspects of the
open-search
project

Robin Gareus

Outline

Introduction

Technical
Aspects

abstraction
models

technical details

impact of
technical
decisions on
design

Outlook

network topology

Outline

Introduction

Technical
Aspects

abstraction
models

technical details

impact of
technical
decisions on
design

Outlook

- distributed objects / hash tables (eg. Pastry, Tapestry)
- anonymity (eg. tor)
- p2p network overlay layer
- route discovery with congestion control on IP level.

routing tables

Outline

Introduction

Technical
Aspects

abstraction
models

technical details

impact of
technical
decisions on
design

Outlook

- routing the most crucial part in the project !!!
- it depends on the topology of the network.
- compromising anonymity (centralized index) might be a key issue.
- technical discussion is beyond the scope of this introduction.

comparison of existing implementation

Technical
aspects of the
open-search
project

Robin Gareus

Outline

Introduction

Technical
Aspects

abstraction
models

technical details

**impact of
technical
decisions on
design**

Outlook

trade offs

Outline

Introduction

Technical
Aspects

abstraction
models
technical details
impact of
technical
decisions on
design

Outlook

- long-living maintained version vs. prototype vs. simulation or emulation
- latency and network speed vs. anonymity
- portability vs. development speed
- result consistency vs. query speed
- trust vs. anonymity

the glue software approach 1/2

- no need to re-invent the wheel
- back-end in POSIX c or c++ (gnu libs)
- import/export data plugins (slow but useful)
- but beware of basic design mistakes when collating small tools to accomplish complex tasks!
- do not borrow buggy code - link against supported libraries.

the glue software approach 2/2

Networking:

- distributed objects / hash tables (eg. Pastry, Tapestry)
- anonymity (eg. tor)
- various p2p client implementations under GPL.

Content management:

- text parsing, indexing and correlation
- XML, xslt result generation
- cache replication
- rating, trust

combine existing solutions and fill in the missing parts.

Communities:

- end user
- tester / maintainer / editor
- high-bandwidth mirror nodes - admin
- developer

possible points of failure

- scalability and speed
- spamming, pollution and corruption
- resource incompatibilities - inhomogeneous network
- (firewalls and corporate technology)

Summary of Ideas

Outline

Introduction

Technical
Aspects

Outlook
community
possible points
of failure
ideas and
discussion

- decentralization, to prevent manipulation (censorship, biasing certain sites)
- redundancy, to exclude single-points-of-failure and to make manipulation difficult
- easily accessible: to maximize participation the number of possible technical obstacles will be kept to a strict minimum, so no complicated installation procedures and no excessive memory usage.
- resistant against attacks (ddos, pollution/manipulation, censoring)