



Fakultät IV – Elektrotechnik und Informatik  
**Intelligent Networks (INET)**  
Research Group Prof. Anja Feldmann, Ph.D.  
An-Institut Deutsche Telekom Laboratories

# The Oracle Protocol

## Draft v0.1

Obi Akonjang  
Vinay Aggarwal  
Anja Feldmann  
Jun Jiang  
Pengchun Xie

{obi,vinay,anja,junjiang,pengchun}@net.t-labs.tu-berlin.de

September 2008

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>The Oracle as a Network Service</b>	<b>3</b>
2.1	Benefits of the Oracle to P2P Systems . . . . .	3
2.1.1	Effect on P2P Topologies . . . . .	4
2.1.2	Effect on Content Download . . . . .	5
2.2	Benefits of the Oracle to the ISP . . . . .	5
<b>3</b>	<b>The Oracle Protocol</b>	<b>5</b>
3.1	Definitions . . . . .	5
3.2	Scope of the Specification . . . . .	6
3.3	Design Goals . . . . .	6
3.4	Design Choices . . . . .	6
3.4.1	Choice of Transport Protocol . . . . .	6
3.4.2	Choice of Approach . . . . .	6
<b>4</b>	<b>Oracle Messages</b>	<b>7</b>
4.1	The Oracle Header Format . . . . .	7
4.2	The Query Message . . . . .	9
4.3	The Response Message . . . . .	10
<b>5</b>	<b>Entities of the Oracle System</b>	<b>11</b>
5.1	The Oracle Client . . . . .	11
5.2	The Oracle Server . . . . .	11
5.3	The Oracle Database . . . . .	12
<b>6</b>	<b>Security Considerations</b>	<b>12</b>
<b>7</b>	<b>Extensibility</b>	<b>12</b>

# 1 Introduction

Although P2P systems depend on the Internet underlay to build their overlay topologies, they do so largely agnostic of it. They use arbitrary neighbor selection mechanisms to construct an additional routing layer on top of (and in duplication of) the Internet routing underlay. The lack of collaboration between the two layers, i.e., the query/key based overlay routing layer and the prefix/policy based underlay routing layer often leads to issues such as routing mismatch and policy breaching.

P2P systems that select arbitrary neighbors to build overlays, then use these overlays to send and receive queries/responses and after a successful search, select arbitrary neighbors to download from, are not only inefficient with respect to topological structure, but also exhibit poor performance with respect to end user experience [3]. The amount of signaling traffic involved in sustaining the system is another factor that affects performance both on the overlay and the underlay.

P2P traffic currently constitutes a very large fraction of the total Internet traffic. In fact, it does not only swarm the Internet, but also has the potential to use up added bandwidth before it is made available to other Internet applications, starving them of critically needed bandwidth. The constant growth in P2P swarms makes it increasingly difficult for ISPs to manage/control the huge volume of traffic that traverses their backbones. This poses a big traffic engineering challenge to the ISPs, who are still looking for an appropriate way to deal with the issue.

Obviously, establishing a means for P2P systems and ISPs to cooperate will not only help ease the situation but can also help eliminate mismatches and improve overall performance. A proposal and feasibility studies based on this approach is provided in [1]. ISPs use their knowledge of the network to offer a service that helps P2P systems select appropriate neighbors to build optimal topologies. P2P systems can also use the same service after a successful search resulting in multiple hits, to determine preferred sources to download from. This service is called the "Oracle".

To determine optimal neighbors from a list of potential neighbors, a peer sends a list of its potential neighbors to the Oracle server for ranking. The server uses its knowledge of the network to rank the neighbors according to pre-determined criteria and sends back to the peer. The peer then uses the ranked list to establish optimal links and build a more efficient overlay.

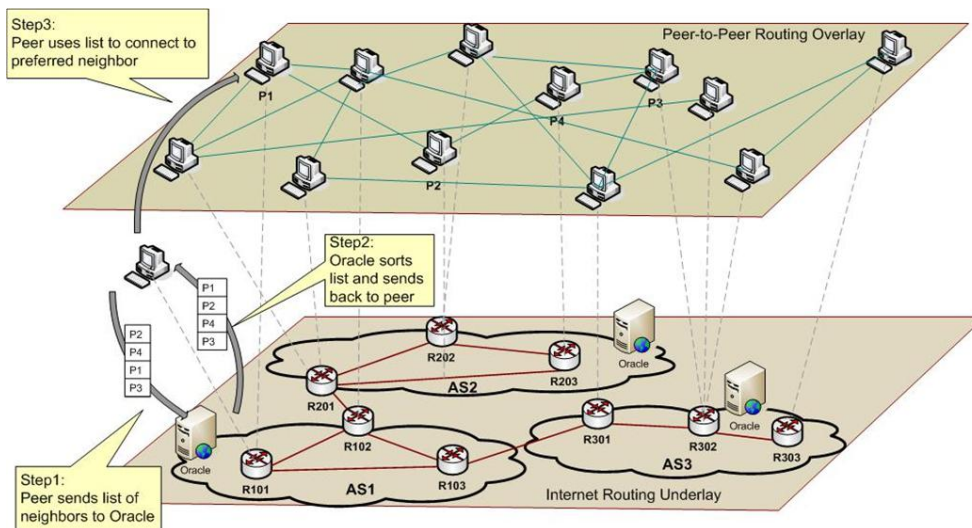
## 2 The Oracle as a Network Service

The Oracle service is an Internet-based decision facilitator that helps end-systems decide between alternative choices. It eliminates the need for end-systems to infer the best choice by themselves.

As an ISP-hosted service that has extensive knowledge of the network, the Oracle can, amongst other things, help end-systems avoid bottlenecks and congested links. End-systems/networks that use the Oracle service are thus able to establish more efficient communication links with each other, in benefit to both the systems/networks and the ISP who is hosting the service. It can be used by any internet-based system, including P2P systems. We show how the Oracle service benefits both P2P systems and the ISP in sections 2.1 and 2.2 respectively.

### 2.1 Benefits of the Oracle to P2P Systems

P2P systems can use the Oracle service to setup more efficient overlay topologies and to select best source(s) from among multiple potential sources to download content from. These occur in the following ways:

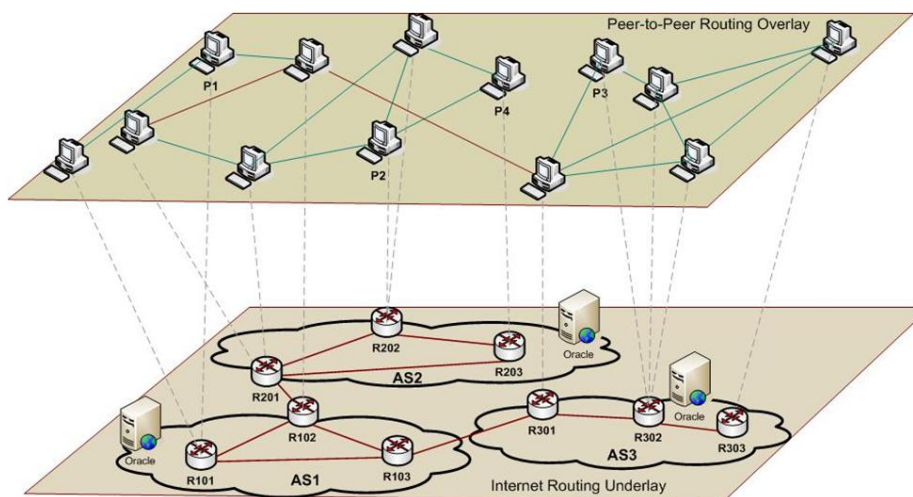


**Figure 1:** Peer uses Oracle to optimally select neighbors when joining P2P system

### 2.1.1 Effect on P2P Topologies

The Oracle helps P2P systems establish localized topologies and manageable traffic flows in benefit to both the P2P systems and the ISP. It occurs as follows:

1. Before joining a P2P overlay, peers can query the Oracle server by sending it a query containing a list of potential neighbors.
2. The Oracle server can then use its knowledge of the state of the network to rank the list of addresses according to certain metrics and send back to the peer.
3. The peer then uses this ranked list to select appropriate neighbors to connect to.



**Figure 2:** Localized overlay established with the help of the Oracle

Figure 1 is an illustration of this process and shows the three steps a peer takes to join an Oracle-aided P2P overlay. The outcome of this process is shown in Figure 2; a localized and more efficient P2P overlay topology that also matches the underlay topology to a very large extent.

### 2.1.2 Effect on Content Download

Peers in the P2P no longer need to measure path performance themselves and can take advantage of the knowledge of the ISP to boost performance and avoid bottlenecks.

1. After a successful search, the peer can query the Oracle server again with a list of potential peers to download from.
2. The Oracle server sorts this new list according to appropriate metrics and sends back to the querying peer.
3. The peer can then use the list to select the most appropriate source to download from.

In principle, the Oracle helps the P2P overlay localize its content transfer traffic.

## 2.2 Benefits of the Oracle to the ISP

ISPs also benefit from the Oracle service. It enables them influence the neighborhood selection process of P2P networks to e.g. ensure locality of traffic and also regain the ability to manage/control traffic that traverses their backbone.

## 3 The Oracle Protocol

The Oracle protocol defines the format and set of messages that are exchanged between an Oracle client and an Oracle server or an Oracle server and another Oracle server.

### 3.1 Definitions

- **Autonomous System (AS):** A single network or a collection of networks under the same administrative control.
- **Autonomous System Number (ASN):** A globally unique number, used to identify an AS and exchange exterior routing information with other (neighboring) ASes.
- **Client:** An Internet end host that requests for Oracle services by creating and sending queries.
- **Server:** An Internet system that accepts and processes Oracle queries.
- **Metric:** Criteria used by server to process/rank IP addresses in a query.
- **Database** A storage system for structured metric data or records that are retrievable by the Oracle server.
- **Query:** A message to request for a particular Oracle service.
- **Response:** An answer to an Oracle query.
- **Error:** Indicates non-conformity.
- **Failure:** A system malfunction.

## 3.2 Scope of the Specification

This document defines only messages that are exchanged between a client and a server or between two servers. It does not deal with messages exchanged between an Oracle server and an Oracle database. These are out of scope of the document.

## 3.3 Design Goals

The main design goals of the Oracle protocol are simplicity, scalability, high performance and extensibility.

## 3.4 Design Choices

Two major concerns of the Oracle design are scalability and performance. A server (or a cluster of servers) should be able to handle a very large number of client requests simultaneously and yet, suffer no penalties in its performance. To address these concerns, important design choices need to be made, including;

- the type of transport protocol to use, User Datagram Protocol (UDP) or Transmission Control Protocol (TCP),
- the option to create a new protocol from scratch or to adapt an already existing one.

### 3.4.1 Choice of Transport Protocol

We decided to use UDP instead of TCP because of the following reasons:

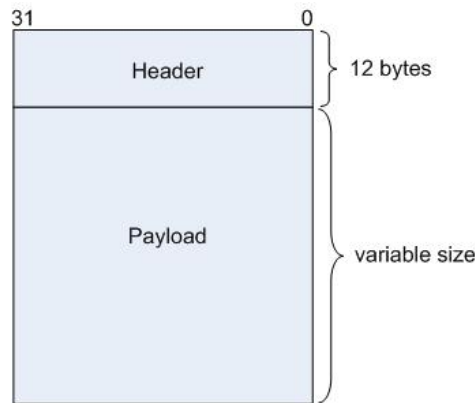
- UDP is faster and more efficient for applications like the Oracle that do not need guaranteed delivery. It thus avoids the overhead involved in checking each packet and correcting those that either arrive out of order, are duplicated or get missing.
- UDP has a stateless nature that is quite useful for servers that need to process small-sized queries from a very large number of clients.
- UDP is preferable for time-sensitive (or very fast) applications that would rather have packets dropped than delayed.

### 3.4.2 Choice of Approach

Instead of creating a new protocol from scratch, we decided to take advantage of an already existing one. In principle, we see a lot of functional similarities between the Domain Name System (DNS) and the Oracle service. The DNS is not only distributed and very scalable, but can also effectively handle many queries from a large number of clients simultaneously. We thus adapted the DNS protocol.

## 4 Oracle Messages

Oracle messages consists of a header and a payload section as show in Figure 3. A typical Oracle packet has a constant header size of 12 bytes and a variable payload size.



**Figure 3:** An Oracle packet

There are generally two types of Oracle messages; the query message and the response message. Both messages have the same header format and differ only in their payload format. Details of the header format and both the query and response messages are given in the sections below.

### 4.1 The Oracle Header Format

The Oracle protocol has a header format that is in many ways, similar to that of the DNS protocol. As shown in Figure 4 below, it is also 12 bytes long but differs in its content. Each of field that constitutes the protocol header is briefly described below.



**Figure 4:** Oracle protocol header

#### Identification

The Identification field is 16-bits long. It is used to correlate queries and responses. The value is generated by a client when creating a query and the same must be used by the server when responding to the query.

#### Flags

The Flags is a 16-bit field that consist of smaller fields ranging from 1-bit to 4-bits. Amongst other functions, they are also used to communicate different kinds of messages from clients to servers and vice versa.



**Figure 5:** Header flags

The Flags consist of the following components and values:

- QR - Query/Response (1-bit): Identifies the message as either a query or a response.

Set as 0 for a query

Set as 1 for a response

- Opcode - Operation Code (4-bits): Describes the type of messages.

0 = COQ (Client/Oracle Query)

1 = OOQ (Oracle/Oracle Query)

2 = STAT (Server status request)

3 = Reserved

4 = Notify

5 = Update

6 = OOQ-BW (Oracle/Oracle Query for bandwidth)

7 = OOQ-D (Oracle/Oracle Query for Delay)

8 = OOQ-BWD (Oracle/Oracle Query for bandwidth + Delay)

9 - 15 = Reserved

**COQ** is set when an end host is querying an Oracle server for ranking.

**OOQ** is set when an Oracle server is querying another Oracle server for ranking [2].

**STAT** is used to request status information from the server.

**Reserved** is an unused bit.

**Notify** is used for server notification (e.g. between primary/secondary Oracle servers).

**Update** allows records in the Oracle database to be selectively added, deleted or updated.

**OOQ-BW** is set when an Oracle server is requesting bandwidth information from another Oracle server.

**OOQ-D** is set when an Oracle server is requesting delay information for all IP addresses in the query from another Oracle server.

**OOQ-BWD** is set when an Oracle server is requesting both bandwidth and delay information for all IP addresses in the query.

While COQ and OOQ generally default to IP address ranking, OOQ-BW, OOQ-D and OOQ-BWD are used to demand more specific information.

- Version - Oracle protocol version number (3-bits).
- OV - Size of the option+value field in the response (2-bit).  
Set as 0 when no options and values are included, i.e., 0-bit (default).  
Set as 1 for 32-bit  
Set as 2 for 64-bit
- IN - Internet address type in payload (1-bit).  
Set as 0 for IPv4  
Set as 1 for IPv6
- AD - Authenticated Data: Used in server/server communications to indicate fully authenticated data in the response packet.

- Rcode - Return Code (4-bits):
  - 0 = No error - No error occurred.
  - 1 = Format error - Construction error in query prevents server from responding.
  - 2 = Server Failure - Server problems prevents it from responding to query.
  - 3 = IP address error - IP address is either invalid or not used on the Internet.
  - 4 = Not implemented - Server does not support type of query received.
  - 5 = Service refused - Server refused to process query for policy-reasons.
  - 6 - 15 = Reserved

### # Unsorted IPs (Query)

This is a 16-bit field that indicates the total number of (unsorted) IP addresses in the query payload. Although it theoretically allows as much as 65 thousand IP addresses to be sent, we argue that a better approach would be to limit the number to an amount that prevents packet fragmentation from ever occurring. Based on this argument we recommend a maximum of 100 IPv4 or 25 IPv6 addresses only.

### # Sorted IPs (Response)

This 16-bit field gives the total number of (sorted) IP addresses in the response. This number must never be greater than the total number IP addresses that was sent in the query, but may be less.

### TTL

The 16-bit Time-To-Live field is used to indicate the number of seconds that the query/response could be cached by the server/client respectively. For queries arriving at the Oracle server, this value could be reset to zero, i.e. the server decides whether it wants to cache a query or not.

### Message Length

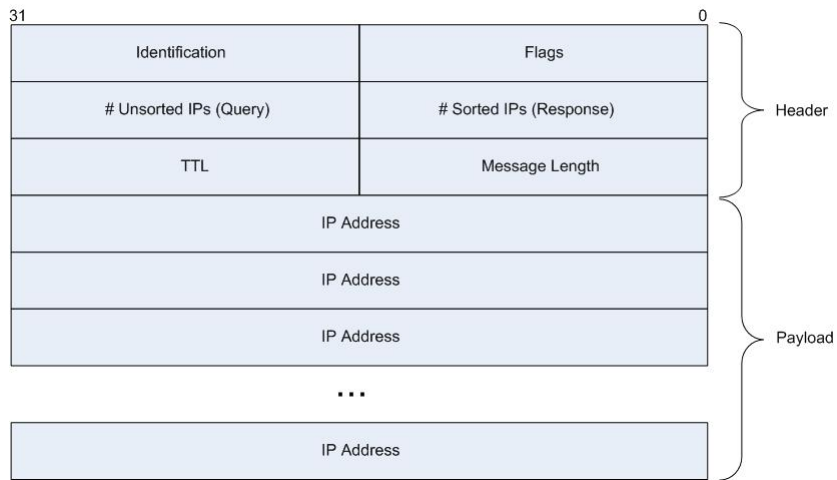
This field specifies the total size of the message (header + payload) in bytes. The message size should always be less than or equal to the Maximum Segment Size (MSS). Considering a Maximum Transmission Unit (MTU) of 1500 bytes, and subtracting the transport (TCP/UDP) and IP header overheads, results in a Maximum Segment Size (MSS) of 1460 bytes for IPv4 or 1440 bytes for IPv6. It is recommended to maintain this limit to avoid packet fragmentation during transmission. This in turn, limits the maximum number of IP addresses that should be sent in the query or response.

## 4.2 The Query Message

The payload of the query message consists only of IP addresses. These can either be 32-bit IPv4 or 128-bit IPv6 addresses, but not a mixture of both. Presently, only end-host IP addresses are used in the payload. This would be extended in the future to include prefixes as well.

A standard (default) Oracle query is that which is sent from an Oracle client to an Oracle server. It has an Opcode value of 0.

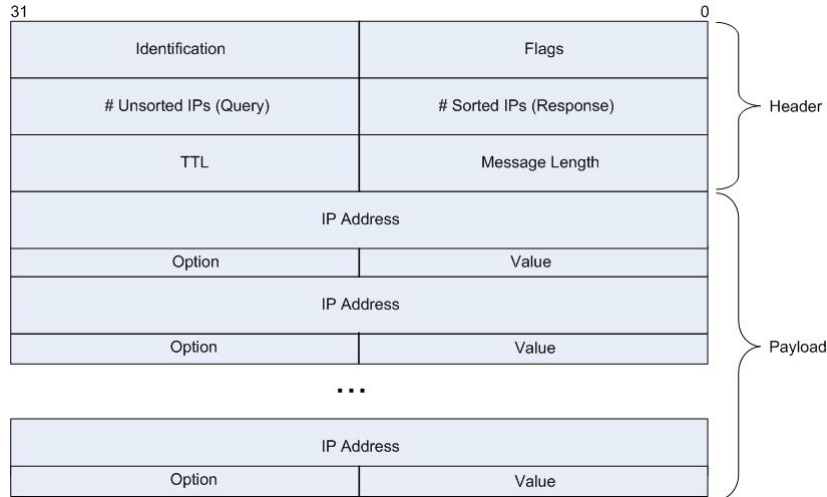
An Oracle server can also use the query message format to query another Oracle server. In this case, the Opcode value is set to either 1, 6, 7 or 8, depending on the type of information or service the querying Oracle server needs from the responding Oracle server.



**Figure 6:** Query message format

### 4.3 The Response Message

The response message is similar to the query message in structure, but contains additional option and value fields per IP address, as shown in Figure 7. These fields are used to send additional information per IP address from one Oracle server to another, such as in a global coordinate system.



**Figure 7:** Response message format

A global coordinate system is a system made up of Oracle servers that are controlled by different ISPs. The additional information supplied by the Option and Value fields can help the receiving server make better decisions before responding to client-generated queries. The OV bit is set to 0 when no Options and Values are sent. Else, a 32-bit or 64-bit Option+Values information per IP address is sent with the response when the OV bit is set to 1 or 2 respectively.

The order of the IP addresses in the response payload is very important. The first IP address indicates highest priority or preference and the last one, least priority or preference.

## 5 Entities of the Oracle System

An Oracle system consists of Oracle clients, Oracle servers and Oracle databases. These three entities implement the Oracle protocol in order to interact with each other. Although each entity independently plays an important role in the overall system architecture, only the client and the server are generally visible from an end user's perspective. The Oracle database is generally visible only from the server's perspective and should be completely transparent to all end users or clients.

### 5.1 The Oracle Client

To use the Oracle service, a client must generate a standard Oracle query message and send to the Oracle server. It can also cache a copy of the sent packet. Figure 8 summarizes the steps a client undertakes, to generate and send a query.

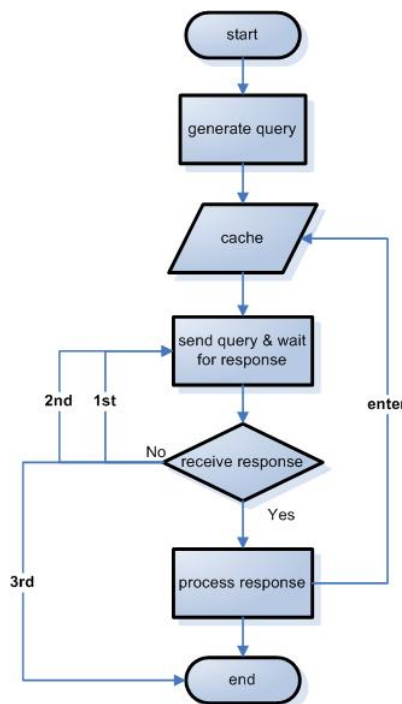


Figure 8: Flow diagram - generating and sending a query message

After sending a query, the client should set an external timer, and then wait for the response from the server. A timeout value of 400 milliseconds is recommended. If the timer times out without the client receiving a response, it should re-send the same request one more time and reset the timer once again. After a maximum of two unsuccessful repetitions, the client should terminate the process. It must then clear the cache before attempting to contact another server.

### 5.2 The Oracle Server

When a server receives a query from a client it must respond using the appropriate Rcode values mentioned in Section 4.1.

Before it responds to any query, the server must contact the Oracle database for the most up-to-date information. If need be, the Oracle server can in turn query other Oracle servers to request needed or up-to-date information.

A server that is responding to another Oracle server must authenticate the data and set the AD bit to indicate that the response contains only authenticated data. This applies to all Oracle servers, irrespective of who is hosting them (i.e. the same or another ISP, such as in a global coordinate system [2]. Authentication establishes a kind of trust between the Oracle servers.

### 5.3 The Oracle Database

The Oracle database is a dynamic database that stores information such as ISP prefixes and other topological in the form of records. These records are partly static, such as the ASN of an IP address, and also partly dynamic, such as the current delay or level of utilization between any two Prefixes/IP addresses. These metric information are used by the server to rank IP addresses.

The database is kept current by means of updates/edits to its records, so that it is able to reflect the current state of the network at all times. Considering the dynamic nature of the state of the Internet, and the fact some data do change frequently, while others change more slowly, updating the database immediately a change is noticed might not be the best approach. A minimum and reasonable time interval, based on the Oracle provider's own assessment, could be used to establish a balance.

## 6 Security Considerations

The Oracle system must ensure that data is exchanged between Oracle servers in a secure manner. The AD bit in the protocol header is used to indicate authenticated data. Details of this aspect and further security considerations will either be treated in a separate document or be included in future versions of this document.

## 7 Extensibility

This is the first draft of the Oracle Protocol specification document and is by no means complete. Issues such as details of the communications between an Oracle server and its database and between two Oracle servers will be addressed in future versions of the document.

## References

- [1] Vinay Aggarwal, Anja Feldmann, Christian Scheideler. *Can ISPs and P2P systems co-operate for improved performance?* ACM SIGCOMM Computer Communications Review (CCR), 37(3):29-40, July 2007
- [2] Vinay Aggarwal, Anja Feldmann, Roger Karrer *An Internet Coordinate system to enable collaboration between ISPs and P2P systems* In Proceedings of the 11th International ICIN Conference, (Location: Bordeaux, France), October 2007
- [3] Vinay Aggarwal, Obi Akonjang, Anja Feldmann. *Improving User and ISP Experience through ISP-aided P2P Locality.* In Proceedings of 11th IEEE Global Internet Symposium 2008 (GI '08), (Location: Phoenix, AZ, USA), IEEE Computer Society, Washington, DC, USA, 2008