Using Distributed Problem Solving to Search the Web

Amir Langer
School of Engineering and Computer Science
Hebrew University, Jerusalem, Israel
amirl@agentsoft.com

Jeffrey S. Rosenschein
School of Engineering and Computer Science
Hebrew University, Jerusalem, Israel
jeff@cs.huji.ac.il

1. INTRODUCTION

The huge growth of the Internet in recent years encouraged the ermegence of information retrieval tools that assist the user in accessing data on the Web. These tools have been epitomized by many popular search engines, such as AltaVista and Google. Current Web search engines consist primarily of information databases and groups of agents that update this database while browsing through the Web. So far, Search Engines suffer from poor precision and poor recall.

We propose a different approach to gathering information on the Web, by viewing the search problem as a distributed problem that requires cooperation among different agents in order to arrive at a solution. This approach differs radically from the approach of current search engines ([4], [5]) by being distributed, rather than parallel. Our system is composed of autonomous agents cooperating to achieve a better result, rather than spiders feeding a large amount of information into a database. This idea also differs from the meta-crawler ([7]) approach, because in our proposed method, the different agents cooperate and exchange their partial solutions in order to arrive at one global solution instead of just merging all results into one list.

In Section 2 we elaborate on the intuitions and reasons why we chose to use a DPS algorithm for the Internet search problem. In Section 3 we discuss the analogy between the Device Vehicle Monitoring Testbed and the Internet search problem, and present an outline of our proposed system. In Section 4 we mention our current prototype and several points for future work.

2. USING A DPS APPROACH

The Internet is probably the most natural environment of all for distributed applications, and therefore a DPS system fits it well. Different agents with different limited views of the Internet are capable of cooperation, at the end of which a better solution could be achieved than one giant database with a single view of what's out there, no matter how much faster it receives the information. This is mainly because the limited view of a certain topic / a specific area on the Web that every agent has, is much more accurate and up-to-date than the view of the same topic / area by the giant database. For example, our bookmarks file is a very limited view of the web, but if we queried it for some information it possesses, the results would be much more accurate and up-to-date than querying a search engine. The only problem with this approach is that our bookmark's view of the web is extremely limited. But what would happen if many users would cooperate by sharing their "bookmarks"? Cooperation among several autonomous entities is what is implied by a DPS approach.

3. SYSTEM OUTLINE

From the point of view of the user, let's assume that you're looking for information about the island of Java. This is exactly the sort of query for which every syntactic-based search engine would return lots of information about the Java programming language, Java coffee, etc. In order to find the information you want you might use several search engines, follow several links, and after some filtering, even find a few sites you might like. Then, you would probably save those "good" sites in your bookmarks. When you look for more sites, you would probably prefer to exchange your acquired knowledge with other Java enthusiasts around the world and get immediate links to related sites, rather than to start searching all over again. Note that as the number of persons who exchange this knowledge increases, you don't need to hold information about Java to receive Java-related sites from the system. You could get the Java-related sites in exchange for your knowledge about "shareware sites", for example. The system would also continue to provide good quality knowledge, as long as more knowledge of that quality is poured into it, where this "it" is a totally virtual concept, just like the Internet itself. You could view this solution as an attempt to enhance the recall rate of many small Web indexes that have good precision rates (many small "Yahoos", if you like, with a very limited views of the Web, cooperating to create a much larger virtual Web index). Therefore, our algorithm should know how to automatically create a small database with specific expertise based on a small portion of the web, and how to integrate all these many databases into one large system.

We create this good precision rate database by finding clusters of related pages (using either the hyperlinks between them, or an occurrence of words, in those pages), and then finding a topic for these page clusters via user intervention. The communication mechanism between small databases is drawn from the DVMT.

3.1 Analogy to DVMT

DVMT was designed as a framework for Distributed Problem Solving algorithms ([1], [3]). It includes a simulation of a network constituted of several problem-solving nodes. Each node applies simplified signal processing knowledge to acoustically sensed data to identify location and track patterns of vehicles moving through a two-dimensional space. The data that is used by the sensors is the location of the signal and its frequency. The DVMT architecture is based on the Hearsay-II architecture [2], with knowledge sources and abstraction levels appropriate for vehicle monitoring. The goal in the DVMT system was to construct a high-level map of vehicle movement by cooperation among different sensors, where each had a limited area it can sense or different fields of expertise. In our system, we view each agent as a sensor with a limited view of the Web, where instead of trying to chart vehicle movement we attempt to find groups of related pages and a topic that relates to a specific group.

3.2 System View of WWW

In the same way that location and frequency data is used by DVMT nodes, our agents view the Web according to two different signals, a Context Signal (a prominent word appearance on a specific page) and a Hyperlink Signal (a prominent hyperlink from one page to another). This produces two graphs of the Web. The first graph is produced from the Hyperlink Signals and is the straightforward WWW graph of pages as nodes and hyperlinks as edges. The second is a graph constructed from the Context Signals by viewing the pages again as nodes and defining an edge between nodes *x*, *y* if for a specific word *w* exists Context Signals *(x,w)* and *(y,w)*. The belief parameter of both signals is based on the prominence of the link or word on the page (this could be the size of the text or image, its position on the page, etc.).

This system view of the Web allows us to define goals, and plans, and to use the same architecture as described in the DVMT system. We use a planning mechanism by generating sub-goals and solving sub-problems. This planning process can be distributed among several agents and is initiated by the signals picked up by the sensors (lowest-level goals). From these achieved goals (hypotheses), the planners plan, attempting to achieve higher goals until we get a group of pages that are linked to a specific topic – The Topic Group goal.

The Context and Hyperlink Signal types are a result of the information we get from the basic lowest level signal — The Page Signal (a Web page and its URL). The URLs can be inserted into the system as an initial input or as a side effect of finding a Hyperlink Signal to an as yet unknown URL. This is a low level goal that is always achieved.

Higher-level goals define a higher abstraction level. The goals direct the agent to construct collections (= groups) of Web pages according to context or hyperlinks. These groups are equivalent to the 'track' goals of the DVMT. The goal driven planner goes higher and higher in the abstraction level with each goal, and one of the agents' planners should arrive at the highest level goal which is the Topic Group.

The Topic Signal is specific to our domain. It is used to define a specific word or sequence of words as a topic of a specific page. Defining the topic as a signal enables us to integrate the phase of creating the topics or concepts into the agents' process and in this way create topics or link topics to existing pages while the system is already running. This enables a much more dynamic mapping of the web. The signal data is generated by the user and used as a trigger for the agent planner that would now create a plan to achieve the goal of a group of Web pages which correspond to this topic — i.e., achieve a Topic Group for that topic.

Higher abstraction levels are: a) Two Context Signals belong to the same Context Group if both are defined by the same context; b) Two Hyperlink Signals belong to the same Hyperlink Group if there is a short path (links) between the pages in the signals.

The Group goal is a higher level goal used to merge hyperlink and Context Groups into one collection of pages. A Group is formed from a Context Group and a Hyperlink Group that have a large correlation between their elements (Web pages). The Topic Group goal is actually the main goal of the whole system, i.e., to build a group of Web pages that are related to a specific Topic Signal. The agent's planner tries to achieve this goal by building a Topic Group from a Group of pages and a topic that is related to a large number of pages in that Group. (In other words, there is a Topic Signal of a specific topic defined in a large number of pages in the group.) Note that this end result returns us not only the pages that were defined as connected to the topic, but all "close" pages in terms of close nodes in the Internet graph, and in terms of similar context. That context is not necessarily the actual topic words, so the results are not based on any syntactic pattern.

The system is organized according to specific interest areas for each of the agents. This interest area may consist not only of part of the WWW space (a limited number of hosts, for example), but also part of the solution space. (Agent1 may be involved only in creating signals while agent2 may not extract information from the Web at all, and would limit itself to constructing higher level goals out of these low-level signals.) Because our view of the Web is based on low-level signals and higher levels of abstraction of those signals, we can organize our system of agents according to the levels with which every agent deals.

In our domain, the intervention of an expert or an expert agent is highly important. We intend to enable the user to add her own signals or even groups to the list of achieved goals—hypotheses (with a high belief rate). This is how user bookmarks are supposed to get integrated into the system. The user should also set all the Topic Signals of all topics she would like the agent to explore.

4. CURRENT AND FUTURE WORK

We built a prototype that explores the potential of our approach. Currently we test using two agents who communicate and classify pages from different portions of the web. The current prototype builds graphs without weights on the edges and does not have a planner, but results are already proving to be quite promising.

There are a variety of directions in which to expand the research described in this paper. These directions include the following: a) Adding weights to the graphs edges is bound to give a more accurate view of the Web; b) Using a planner to direct the agents' actions. Extensive work done on Partial Global Planning and Generalized Partial Global Planning by Durfee [3], Decker [7], and others may be most relevant here; c) Enabling expert intervention – drawing data from real bookmark folders; d) Testing on a much larger number of agents.

5. REFERENCES

[1] E. H. Durfee, V. R. Lesser & D. D. Corkill, "Coherent cooperation among communicating problem solvers", in Readings in Distributed Artificial Intelligence, Morgan Kaufmann Publishers, San Mateo, 1988, pp. 268–284.

[2] L. D. Erman, F. Hayes-Roth, V.R. Lesser and D.R. Reddy, "The Hearsay-II speech understanding system: Integrating knowledge to resolve uncertainty", Computing Surveys, Vol. 12, pp. 213–253, June 1980.

[3] E.H. Durfee & V. R. Lesser, "Using Partial Global Plans to Coordinate Distributed Problem Solvers", in Readings in Distributed Artificial Intelligence, Morgan Kaufmann Publishers, San Mateo, 1988, pp. 285–293.

[4] AltaVista: http://www.altavista.com

[5] Yahoo: http://www.yahoo.com

[6] MetaCrawler: http://www.metacrawler.com

[7] K. S. Decker and V. R. Lesser, "Generalizing the Partial Global Planning Algorithm", International Journal of Intelligent Cooperative Information Systems, Vol. 1(2), 1992, pp. 319–346.