A Self-Configuring Agent-Based Document Indexing System

L. Peng, R. Collier, A. Mur, D. Lillis, F. Toolan, J. Dunnion

Department of Computer Science,
University College Dublin (UCD), Belfield, Dublin 4, Ireland.
{ peng.liu, rem.collier, mur.angel, david.lillis,
fergus.toolan, john.dunnion}@ucd.ie

Abstract. This paper describes an extensible and scalable approach to indexing documents that is utilized within the Highly Organised Team of Agents for Information Retrieval (HOTAIR) architecture.

1 Introduction

This paper describes the HOTAIR Search Engine architecture, an *extensible* and *scalable* architecture for the discovery, retrieval and indexing of documents from multiple heterogonous information sources. Within the HOTAIR architecture, extensibility is engendered through the design of an architecture that provides support for: (1) the plugging in of multiple retrieval strategies such as the Vector Space Model [5] and the Extended Boolean Model [6]; (2) the ability to rapidly and seamlessly integrate diverse sources of information. This requires the use of an open infrastructure that is able to dynamically adapt its configuration.

2 The HOTAIR Indexing System

The HOTAIR Document Indexing System has been implemented using Agent Factory [2], a cohesive framework that delivers structured support for the development and deployment of multi-agent systems, which are comprised of agents that are autonomous, situated, social, intentional, rational, and mobile [1].

A diagrammatical overview of the agents that make up the system architecture is presented in figure 1. The actual number of agents that exist at any time varies depending upon the demand on and the resources available to the system. In addition, these agents are deployed over a number of different agent platforms that reside on different physical machines.

The creation of agents is a service that is provided by the Platform Manager (PM) system agent. Each agent platform contains a PM, which is responsible for handling requests to create more agents. Upon receipt of a request, a PM negotiates with its counterparts to decide which on machine(s) the requested agent(s) should be created.

If there are insufficient resources to create all of the requested agent(s), then the PM agents can either refuse or partially fulfil the request.

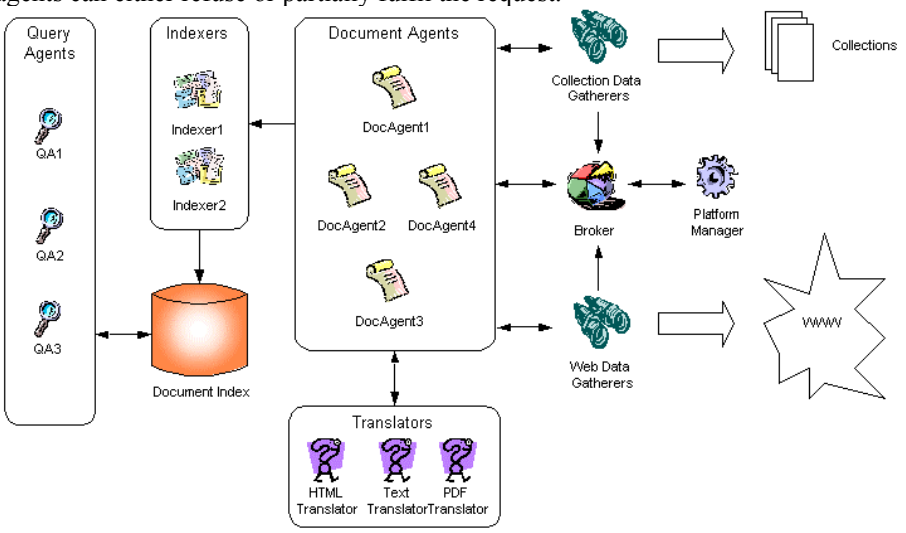


Fig. 1. The HOTAIR Document Indexing System

3 Experiments: Impact of Document Agents on Indexing Speed

The hypothesis for our experiment is that a specific number of Document Agents (DA) is anticipated for optimal indexing speed. The approach taken in the evaluation of this hypothesis was to configure the HOTAIR architecture to index four document collections with different features. (Figure 2)

To perform the experiment, a simplified version of the HOTAIR architecture was constructed, which consisted of: one *Data Gatherer* (DG), which is charged with the task of analyzing information sources; one *Indexer*, responsible for indexing documents and one *Broker,* which is responsible for monitoring the status of the DGs. The Broker can ask the local AMS (Agent Management Service) agent to create DAs. When significant disparities exist, the Broker re-assigns some existing DAs to different DGs. A fixed number of DAs encapsulate the workflow of the system, that is, they know how to get a document indexed.

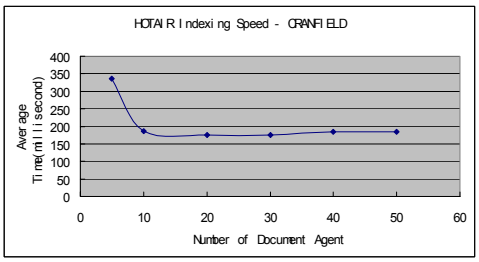
Dataset	No. of documents	Average no. of terms per Doc	Coefficient of Variation
Cranfield	1400	95.18	50%
LISA	6003	46.58	45%
Med	1033	83.72	56%
Time	423	326.61	92%

Fig. 2: Table of the four collections used in the experiment

The figure 3 shows how the number of agents affects the indexing speed. Every graph plots the mean indexing speed for each document collection.

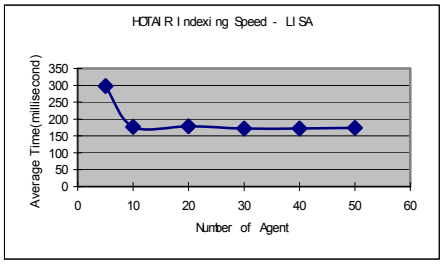
Cranfield, Lisa, and Med performance increases up to a point, and then slowly decreases after that point. Their optimal speed is approximately 170 milliseconds per document.

Cranfield Result Set



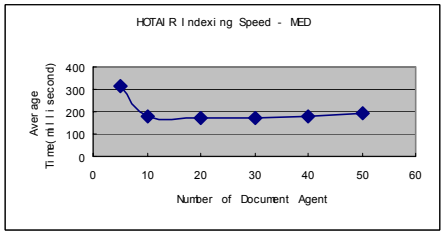
(a)

LISA Result Set



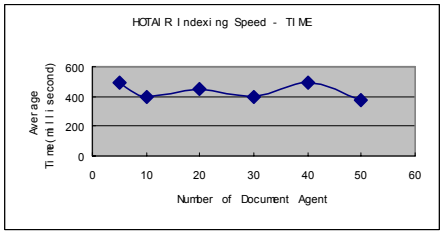
(b)

MED Result Set



(c)

TIME Result Set



(d)

Fig. 3: Graphs illustrating the experimental result for the four collections.

These results shows that increasing the number of DAs has the effect of improving performance up to a limit that corresponds to the speed at which the Indexer agent is able to index documents. Once this limit is reached, adding more DAs has the effect of degrading the indexing speed. The performance of the architecture is worse as it processes the first bundles of documents.

In contrast, the results generated for the TIME collection do not follow this pattern; the mean indexing speed of each document bundle fluctuates wildly, and there is no obvious correspondence between the number of DAs and the indexing speed. On closer inspection, it was felt that this incoherence resulted from a combination of the high level of variation in the number of terms in the documents of the collection (the coefficient of variation for this collection is 92% versus 45-55% for the other collections) and the relatively small number of documents, which makes this variation more marked.

In summary, the results for the Cranfield, Lisa, and Med collections support our hypothesis, namely that the number of DAs does have an impact on indexing speed.

As indicated earlier, the optimal speed of the architecture is bounded by the speed at which the Indexer agent can index documents. This speed is proportional to the size of the document that it is indexing.

4 Conclusions / Future Work

This paper presents an agent-based document indexing system for the HOTAIR architecture. This architecture is able to dynamically reconfigure itself to reflect changes in demand through either the creation of additional DAs or through the cloning of Indexer or Translators agents.

It is our intention that, ultimately, this reconfiguration will be driven by built-in metrics for evaluating performance. However, in an effort to validate the architecture, we present the results of a set of experiments that seek to evaluate whether the number of Document Agents has an impact on the speed at which documents are indexed. These experiments have shown a general pattern of behaviour that supports this hypothesis.

It would seem sensible to assume that, once the optimal number of DAs has been reached for a given indexer, and then performance can only be improved by adding another indexer. Ultimately, we envisage that it will be possible to implement some form of mathematical model that can be used to estimate the number of DAs and Indexers required based on the available resources. The built-in metrics would then be used to make small adjustments to the community of agents based on the actual performance of the system.

References

1. Collier, R., Agent Factory: A Framework for the Engineering of Agent-Oriented Applications, PhD Thesis, Dept. Computer Science, University College Dublin, 2001.
2. Collier, R., O'Hare, G. M. P. Lowen, T. D., and Rooney, C. F. B., *Beyond Prototyping in the Factory of Agents*, In Proc. 3rd Int. Central and Eastern European Conference on Multi-Agent Systems (CEEMAS), Prague, Czech Republic, 2003.
3. Doorenbos, R. B., Etsioni, and Weld, D.S.: *A Scalable Comparison-Shopping Agent for the WWW*, in W.L. Johnson and B. Hayes –Roth (eds). Proc, Proceedings of the First International Conference on Autonomous Agents pp. 39-48, Marina del Rey, CA, USA. ACM Press, 1997
4. FIPA, The FIPA 2000 Specifications, FIPA Website URL: http://www.fipa.org
5. Salton, G. and Lesk, M.E.: *Computer evaluation of indexing and text processing*. Journal of the ACM, 15(1):8-36, January 1968
6. Salton, G., Fox, E. A., and Wu, H.. *Extended Boolean information retrieval*. Communications of the ACM, 26(11):1022-1036, 1983