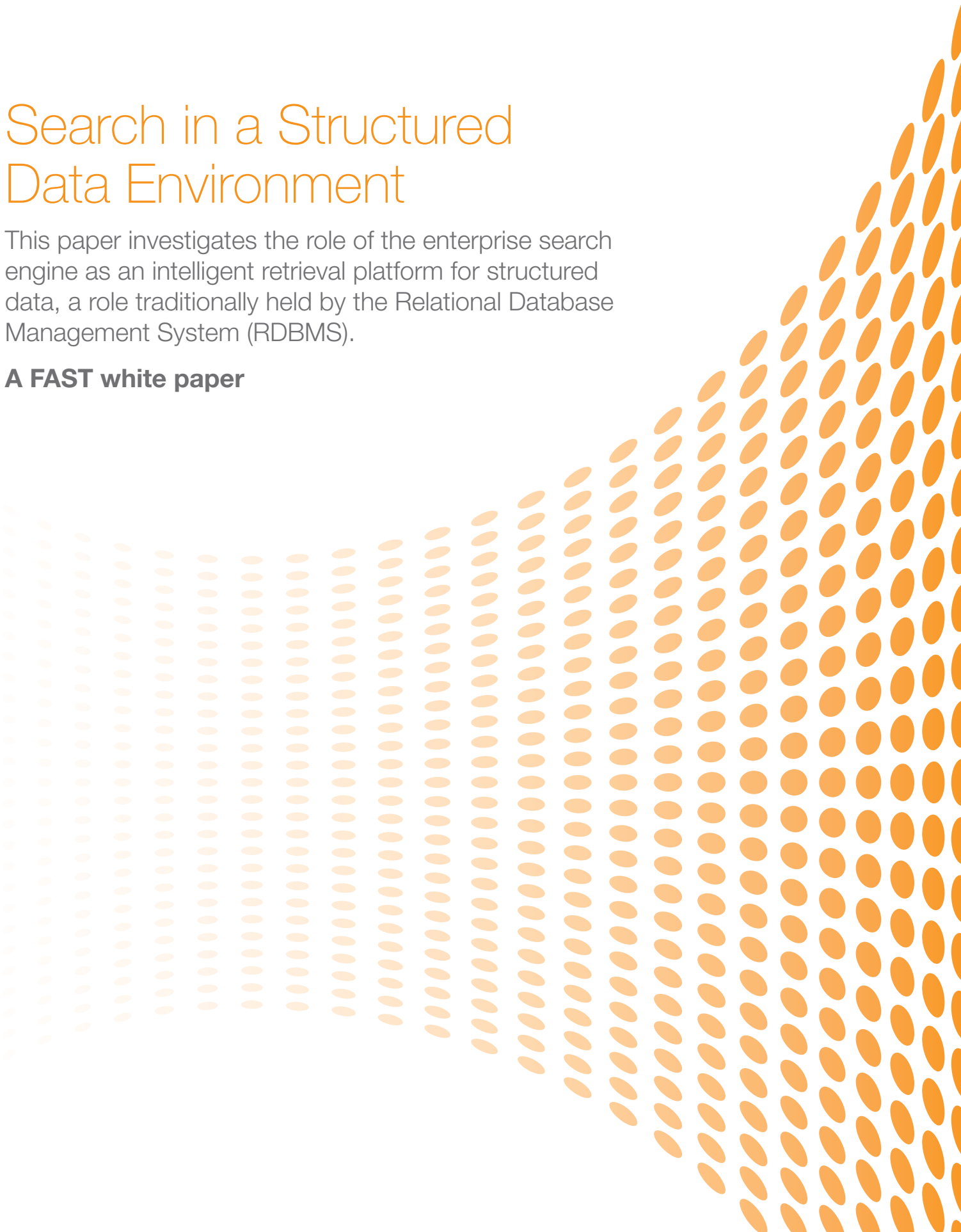
fast

Search in a Structured Data Environment

This paper investigates the role of the enterprise search engine as an intelligent retrieval platform for structured data, a role traditionally held by the Relational Database Management System (RDBMS).

A FAST white paper



Introduction

It has long been held as convention that intelligent exploration of corporate operational systems requires a commitment to Business Intelligence and Data Warehousing technologies. Recent demands, however, have compelled organizations to look for ways to enhance their technical portfolio with capabilities similar to those found at public Web search portals or with their current Content Management Systems. Generally, the demand arises from extreme requirements in performance and scalability, a desire to incorporate unstructured data, or adopting more effective analytics via concept and context-based textual mining.

The Structured World

Every once in a while an industry develops a peculiar ability to sustain two seemingly identical technologies in fundamentally incompatible ways. Think of the media industry with the computer and the television. Both involve a source of electronic information, a vehicle for transmission, and a device for visual display, and yet how long did it take before we could share the same monitor? Record a show on the same machine as the family photos?

Now consider Online Analytical Processing (OLAP) and Search. Both include a central data store; mechanisms for extracting data from multiple sources; logic for intelligent, ad-hoc retrieval; and visualization paradigms to organize the results. They even have parallel jargon: knowledge and data management, text and data mining, search and query, document and row, and so on. And they are entirely incompatible with each other – or were.

Our interaction with structured content (mainframes, databases, applications, etc.) can be divided into two basic groups: one that focuses on creation and management, and the other on intelligent, on-demand extraction and analysis of information. We are all quite familiar with the RDBMS as the de facto standard, but the extraction and analysis process, the traditional prevue of the OLAP environment – Extraction, Transformation, Load (ETL), data warehouse, data mart, and business intelligence – is beginning to strain the relational model. Why is it so hard to support natural language queries? How is meaning extracted from textual content? Why can't the system be more real-time? How can it provide better analytics? Why is the whole process so expensive to manage?

The RDBMS vendors are just now releasing new versions of their database engines embedded with the beginnings of search technology. Perhaps they "peered over the wall" at the search vendors to see how they've been solving these problems. If they did, they would have been surprised to see the search vendors staring right back at them, for they have been mining both structured and unstructured content for some time now, and doing so in ways quite different than the RDBMS.

Search Technology Today

Today's enterprise search technologies are no longer limited to free or unstructured text. They provide core indexing services that focus on extreme performance and scalability for retrieval and analysis of both structured and unstructured content. The binary notion of "either in or out" that forms the basis of the relational model is being replaced with advanced frameworks that provide fuzzy matching and ranking schemes to separate value from noise, and advanced analytics to compute contextual concept relationships across billions of records, all on the fly.

The net affect of these enabling technologies is the appearance of new infrastructure patterns with redefined expectations:

- Finding meaning in the textual components of structured content;
- Federating both structured and unstructured content (documents, emails, Web content, rich media, etc.) in one common extraction and analytics environment;
- Mirroring database content in a search engine to improve query response time and quality of results;
- Handling extreme volumes of data in an efficient and cost-effective manner;
- Adopting the "search and navigate" paradigm for ad hoc querying to improve the user experience with an approachable, natural language and faceted interface;
- Significantly reducing maintenance costs, especially for handling updates;
- Embedding search technology in packaged applications (ERP, CRM, etc.) and storage and document management systems to provide a common extraction and access layer for both structured and unstructured content.

To discover how the search engine is better equipped to handle these expectations, let's take a closer look at the comparative advantages.

The Search Alternative

Scalability and Distributed

Architectures Traditional enterprise architectures gain performance primarily through increased use of CPU and memory. This is scalability through more grid iron and its costs increase exponentially with demand. RDBMS technologies adopted this model from the beginning, which is understandable since it was the prevailing solution when the industry was new and growing. They have recently begun a move towards more distributed models, but the journey must be difficult; to be truly multi-dimensionally distributed means a basic rewrite of your core.

New enterprise architectures are distributed from the ground up, balancing demand for CPU, memory, and disk to reflect the realities of commodity hardware, which is any collection of off-the-shelf, inexpensive computers, typically "Lintel/Wintel", networked together in a grid. Since performance is gained through simply adding another computer to the grid, costs are linear in growth. This is scalability through grid computing, and IT organizations are beginning to adopt this model for all their enterprise requirements. Second generation enterprise search engines embraced this model from inception.

The approach becomes more and more attractive the more extreme the performance requirements. Consider the most demanding data warehouses today with data sizes in the tens of terabytes or billions of rows. In some cases the RDBMS simply can't handle the volume. The market even found room for a high-end (and expensive) dedicated hardware/software data warehousing solution. Now add the migration to online customer services where query traffic can hit peaks of thousands of queries per second and you have a problem on your hands.

Enterprise search engines have been solving scalability problems for some time now. Although rare, data volumes in the range of petabytes and trillions of rows, and query volumes in the thousands of queries per second, exist today. You might question how this can be solved simply by a better distributed architecture and you would be correct to do so. There is another factor involved in the equation we have not mentioned: the transaction, possibly the greatest drag on performance there is. It is obvious why an RDBMS requires transactional awareness: for data management integrity. But in the world of pure retrieval, it gets in the way. Search engines have no transactional overhead because they simply don't support it.

True Ad-hoc Querying and Relevancy

The relational model is exactly that – a model. The fact that there is a schema at all means the RDBMS works better for some queries over others. The issue is not so much who has the fastest query but who has the slowest. For example, the difference between a 20 and 50 millisecond delay is not discernable to the user, but a 30 second response time is. Variance is often far more important than average query time.

The problem lies with the fact that a schema is counter to the notion of a purely ad-hoc query environment. The industry devised new denormalized designs, such as star and snowflake schemas, and even introduced multi-dimensional database engines to alleviate the problem as much as possible, but while these technologies were good for queries generally focused on sales, financial, and production analyses, they were no good for real-time processing, heavy analytics, historical comparisons, complex updates (e.g. data corrections), textually-centric content, or environments requiring a large number of dimensions. A truly unbiased model is one that has no schema at all (i.e. hyper-denormalized to one table). This is the index of an enterprise search engine.

The second problem is the language itself. SQL is designed to query systems where the results support a binary inclusion model. In other words, data is either part of the answer or not. The order of the data itself in the result set has no relevant meaning other than a preference for sorting and summation. This model leaves out the whole world of relative inclusion, where data has a graduated score of relevance in the result list.

The omission is understandable; business intelligence focuses almost exclusively on financial and other numerical data, really to support monetary trends to measure an organization's health or assess risk. However, a lot of information is left on the table when one elects to ignore the textual content or relegates it to simple identifiers for qualifying these analyses. Imagine integrating sentiment scores from news feeds and online trading sites into portfolio analytics. Imagine finding the one email that would prove an important position in a legal deposi-

tion. All of this is standard functionality of the enterprise search engine.

User Interaction Model

The popularity of the Web search vendors (Google, Yahoo!, and Microsoft) has had an interesting effect on the expectations of users in the enterprise market. The simplicity of the search bar, the power of navigators, and the convenience of simple ranked results are now basic requirements. "Why can't our stuff be as simple as Google?" is a common cry. It is not surprising this is the case, since what better an audience to test the acceptability of a user interface than the world at large?

The net effect is a shift in user interaction design (UI is more about "interaction" these days than "interface") as it relates to information retrieval, exploration, and analysis. The new model recognizes the need for an integrated approach to querying and exploring content. The interaction is a combination of techniques that all resolve to the same action: asking for information and getting results. How intelligently the system allows you to ask your question measures the quality of the interaction.

There are some basic required paradigms. The search box is the most common; enter a query and get results. The important point here is the query's lingua franca. The interface should support natural language style queries (SQL is a programming language, not a user language) that understand the variations and ambiguities of the user's native tongue. This is not a trivial exercise as we all know, but the inclusion of advanced linguistic techniques such as spell checking, phrase detection, grammatical inflections, stop word recognition, thesauri, etc. provides an acceptable solution.

The navigation paradigm is an alternative way to generate a search query. It has become quite common in business intelligence environments that support variable reports; every drop down list is an example. But navigation really shines in search environments where lists of terms reflecting concepts or entities in the content are displayed dynamically and in real time. Users click on them as a way to scope the results down to a smaller list – "navigating" in a sense through the content. Entities can be database table fields, terms or categories derived from taxonomic or ontological libraries, or concepts automatically mined from the source, such as proper names, geographic locations, email addresses, phone numbers, etc.

Navigation can also occur through graphics. The pie chart and histogram are simple tools for segmenting content, and clicking on one of their components is a natural way to revise a query. But there are more advanced techniques that focus on managing massive amounts of content, for example, scatter plots, spider diagrams, and heat maps. The pivot button is a common interface in business intelligence environments.

Regardless of the techniques, the basic model is this: providing both a search box and several smart navigation approaches offers the best interface for query and exploration, and an intelligent balance of the two is the most efficient interaction model for intelligent extraction of information. A good second generation enterprise search engine supports all of these capabilities out of the box. The RDBMS vendors are just beginning to incorporate some of them in their latest versions.

More Content, Better Access

There is an often repeated statement that 80 percent of an organization's content resides in unstructured repositories (email, documents, etc.), while only 20 percent resides in databases. The content management vendors use this statistic often. While true, it may also be the case that the 20 percent in the database has 80 percent of the importance. The argument is largely academic, because the real answer is to make the entire 100 percent available.

You can't find something that isn't there, so why not make sure it is in the corpus of searchable content? This is simple insurance against the "not knowing what you don't know" problem that inevitably inflicts the worst pain on an organization. Furthermore, the integration of structured and unstructured content allows for new investigative approaches. Imagine monitoring the efficacy of an advertising campaign by tracking product sales and market intelligence in a coordinated fashion.

We are usually clear as to where our structured content resides, but unstructured content exists everywhere: email, documents, content management systems, Intranet sites, the Web, media (image, audio, video), even (and perhaps most important in the context of this article) embedded in the database fields themselves. Consider the common description or details field in a table. There is text in there worthy of all the capabilities of the search text mining process.

A typical online career site is a good example where this comes into play. While it is nice to imagine every job provider nicely fills out each field in the online form, in reality, some of them leave fields blank (e.g. company name, salary, city, contact) and simply include their standard description. How then does the job seeker find this listing? The search engine provides the answer by mining the description to extract the field content as entities.

Finally, we often think of content as primarily historical, a snapshot of activity that occurs over a period of time (which may be as recent as the last 24 hours). This is such a standard assumption the batch orientation of the data warehousing market requires that it be true. But it is not always true. What if I'm monitoring stock transactions and I wish to get answers before the trader executes the manager's requests? What if my job is to look for patterns on the newswire? The data in these scenarios is streamed in real time, not digested in batches, and it should be supported as such. Note the importance of latency here. Streamed content should be searchable with minimal delay, often less than a second.
Second generation enterprise search engines support alerts and sub-second latency. In an OLAP environment this would be considerably more difficult.

Cost of Maintenance

In a typical OLAP environment you will find one or more data warehouses, several operational data stores and/or data marts, assorted ETL technologies, and a myriad of business intelligence repositories that define reports, queries and other assorted meta-data. Creating and maintaining these components requires a host of engineers, DBAs, and knowledge workers (who we pretend do not need to understand relational concepts but really do if they build reports or queries, even graphically). Now add to this environment meta-data management, keyword and acronym dictionaries, multiple languages, thesauri, etc. and you can see how the management costs can proliferate.

Search engines, especially those with a history of servicing the Web, are designed for the widest audience, consumers who know absolutely nothing about technology. The knowledge worker does not exist in the consumer market, so the ad hoc report at its most simplest is merely a search bar that returns ranked results and sometimes dynamically generated navigators to assist in exploration. The management process, as you might imagine, is also much easier, if for no other reason that there is one component to install, configure, and manage rather than several. And note that a DBA is no longer required.

The "single component" comment requires some explanation. Recall that data marts and operational data stores were created to overcome the limitations of the relational model for handling ad-hoc querying – avoiding the "killer query" – and because the RDBMS was never able to efficiently support high volume data updates and query traffic at the same time. The search engine has no such limitations.

Change Management

If I ask for all sales data in the last quarter and break it down by geography, then by product, I know a fairly straight forward star schema will do the trick. After all, this is why we invented dimensions in the first place. But what if I wish to compare the results to the same data three years earlier? We all know the problem of maintaining history in a data mart, so we usually don't. This means a lot of design work. Not with a search engine, however, since search technology has always had to support ad-hoc querying against massive volumes of information.

Now let's say I would like to include in my report a field not in the data mart. The process would require adding it to the model, creating the stored procedures, etc. to extract the content from the warehouse and update the field. While this is similar for the search engine (adding the field to its profile and defining its extraction logic), what if the field does not exist in the warehouse either? Now the process is considerably more involved. The problem is exacerbated if we have a considerable number of stored procedures that ensure integrity with deletion and update control (a common practice). Again, not so with a search engine index as it does not use a relational model and the data warehouse and data mart are the same entity.

In general, the cost to keep a search engine in line with either data changes (insert, update, or delete) or profile changes is far less than the cost to do so in an OLAP environment. And as more functional demand is placed on the system, the savings accumulate.

Conclusion

Enterprise search technology may not have been around as long as its RDBMS counterpart, but it has been much more aggressive in its intelligent retrieval of information from both structured and unstructured content. Perhaps the reasons are serendipitous: they flourished just around the same time as the market began moving from grid iron to grid computing architectures, when data volumes grew at an enormously accelerated rate, and when the Web search vendors really brought the advantages of search to a common light.

Whatever the reasons, for organizations to compete effectively, report accurately, defend aggressively, or most any other activity, the key to their success is their ability to get the right information to the right people at the right time (and at the right cost); from any source, internal or external to the organization. The enterprise search engine is the only technology that can effectively deliver.

About FAST

FAST is the leading developer of enterprise search technologies and solutions that are behind the scenes at the world's best known companies with the most demanding search problems. FAST's solutions are installed in more than 3500 locations.

FAST is headquartered in Oslo, Norway and Needham, Massachusetts and is publicly traded under the ticker symbol 'FAST' on the Oslo Stock Exchange. The FAST Group operates globally with presence in Europe, North America, the Asia/Pacific region, South America, the Middle East and Africa. For further information about FAST, please visit www.fastsearch.com.

For any feedback or questions related to this paper, please contact us at feedback@fastsearch.com.

FAST™
www.fastsearch.com
info@fastsearch.com

Regional Headquarters

The Americas
+1 781 304 2400

Europe, Middle East & Africa (EMEA)
+47 23 01 12 00

Japan
+81 3 5511 4343

Asia Pacific
+612 9929 7725

FWP.002.01.040606