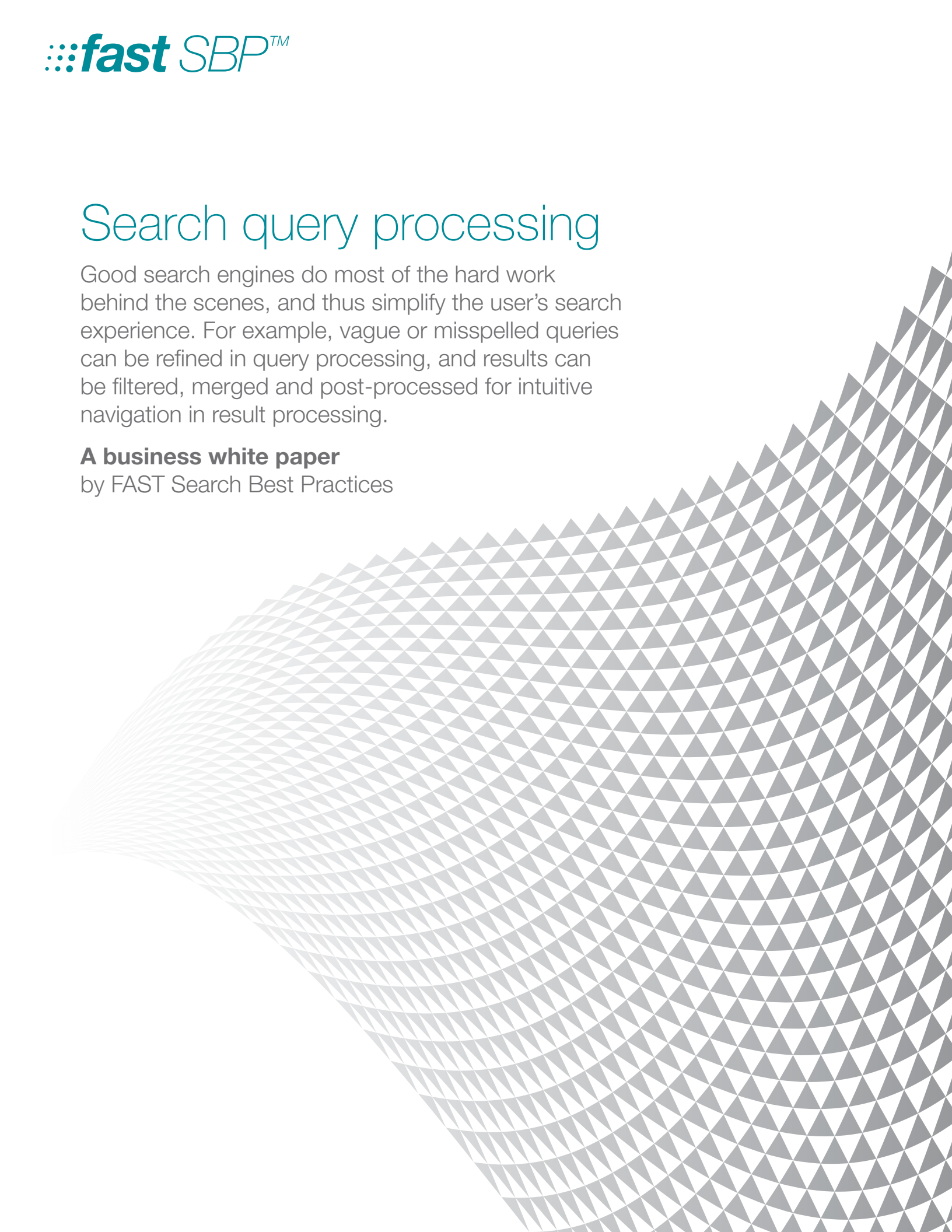
fast SBP™

# Search query processing

Good search engines do most of the hard work behind the scenes, and thus simplify the user's search experience. For example, vague or misspelled queries can be refined in query processing, and results can be filtered, merged and post-processed for intuitive navigation in result processing.

**A business white paper**

by FAST Search Best Practices



## 5 things you should know about query processing

1. QR processing, or query and result processing, is the application of algorithms to the original query and/or to the raw results returned by the search engine
2. The goal is to analyze human language to identify the context of the searcher's intent in order to return the most relevant set of results
3. Loading the system beyond its query rate capacity will create a backlog of queries
4. The query rate capacity is limited mainly by the number of search rows, with multiple search rows providing a linear scaling of QPS
5. OEMs should leverage the query API of the search system rather than relying on HTTP

Searching is becoming increasingly complex. Queries now include single words, phrases and questions, and whole passages and documents. In some cases, the right result can be a single document or answer. In most cases, the correct result is an array of relevant information, strengthened by precise navigation to related information and topics that can help the searcher discover other insightful results or get a more complete answer.

To deliver consistently superior results, you must understand the exact intent of the query. You must also know what information is available, how it relates to the query, and where it is located. Accomplishing these goals requires a mixture of technologies, each with complementary strengths.

In search, true success comes from understanding what the user is asking from their query. Some user queries are simply stated, while others are stated in a Boolean format ("apples AND oranges OR bananas"), or presented as whole paragraphs, passages, or documents with a request to "find similar" information. So the search platform must have a range of tools in order to accurately understand what is being asked.

Two applicable technologies are Natural Language Processing (NLP) and linguistic analysis. NLP interprets queries posed as questions, phrases, etc., in part by identifying and stripping out terms that don't contribute to the relevance of the results. Linguistic tools include capabilities that circumvent word-sense ambiguity – for example, distinguishing between the color orange and the citrus fruit. Search applications use these technologies to analyze human language, identify a searcher's intent, and return the most relevant results.

## Enhancing search with QR processing

The challenge with information retrieval revolves around two basic problems: 1) getting a good query from search users with the aim of helping them craft better questions, and 2) presenting "easy-to-judge" results to minimize what the user has to read through. For example, are a title and the first few sentences of an article a satisfactory result?

Q: My company has an integrated search platform that connects multiple content sources from files systems, ERP data, and corporate applications. How can I ensure that employees can access only the information they're permitted to access?

A: To begin with, you should look to use the underlying security principles or models for each employee, role, application, etc. and make this data available to the search application. You can also configure multiple rank profiles and relevancy models that will surface only the permitted content for the respective employee roles or groups.

Basic search cannot always figure out which words are most important. In the query "How do we replace our Social Security cards?" is "social" more important than "security?" Phrases are not always obvious; do "social" and "security" form a phrase? Boolean formatted queries are not always clear; is it "social" AND "security" or "social" OR "security?" And the query could have an "unstated" question – the user may just want everything about "social security."

Turning every search request into a well-understood query requires analysis of ambiguous types of queries as well as alternate complementary analysis capabilities.

## Enterprise search engines should analyze queries along these four dimensions:

Orthographic -- checking for typos, official variants (e.g., German/Dutch spelling), etc.

Morphologic -- including all forms of a given word via linguistic normalization (lemmatization)

Syntactic -- entity or phrase extraction, anti-phrasing, removing word-sense ambiguity (orange color vs. fruit), etc.

Semantic -- applying a combination of general and specific thesauri and ontologies, automatic phrasing, etc., to understand the intention of the query.

In order to effectively analyze the search query and deliver appropriate results, search applications rely on a key component known as the "'query and result processing" engine (see diagram on next page). Fundamentally, QR processing is the application of algorithms to the original query or to the raw results returned by the search engine. In general, queries from the user come into the query processing and transformation subsystem. This framework takes the original query, analyzes it, transforms it with, for example, corrections of spelling mistakes ("Nisan Macra" will be corrected to "Nissan Micra"), and then sends the query to the search engine.

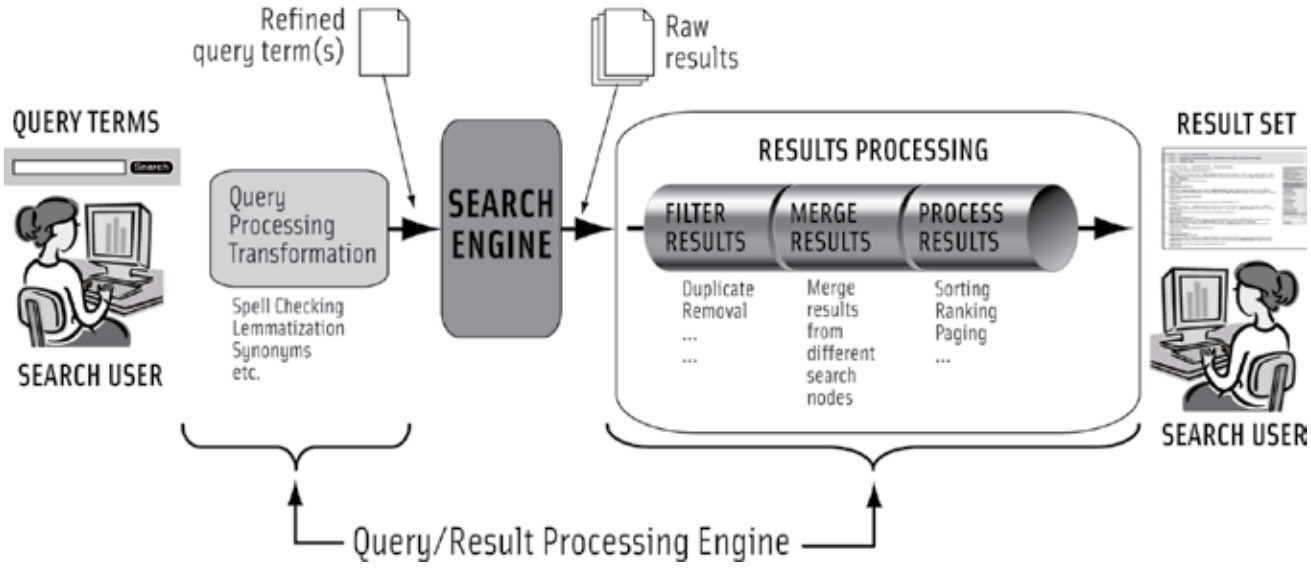
The node in the search matrix that receives the query performs its retrieval operation and returns its results to the results-processing subsystem. The raw results are passed to the results-processing subsystem (which performs duplicate removal), results merging (from different search nodes), sorting, rank ordering, etc. All results are then sent to the search user.

In general, QR processing is measured in terms of QPS (queries per second) and by the perceived relevancy of the results. These measurements are affected mostly by the following: the hardware used in the search matrix, the limits imposed by licensing (e.g., how many search nodes), the linguistics features available in the index, the query's complexity, and the query-specific features invoked.

Use of the QR processing stage depends on the complexity of your content, your business model, and your search goals. The depth and quality of processing that the search platform performs on content directly affects the speed and quality of the query results.

When leveraging QR processing, search customers will typically start with the functionality that comes standard with the search application. Over time, they will introduce more complex processing capabilities to deal with the mix of different data sources. Linguistic tools such as spell checking, lemmatization, and query re-writes can then help to improve the relevancy of the results.

### Elements of query and results processing



Hidden within the search application is the relevancy model that can be tuned to meet the search needs of particular users or aligned to the business environment. (This is discussed in more depth in the "Relevancy" white paper.) In this environment, simple and advanced queries would be supported, ranging from simple keyword searches to Boolean operators.

Depending on the profile of the user, organizations may wish to use the appropriate security and filtering features to ensure that the user gets access only to the information that he or she is permitted to view. For example, if employees in research and development search the company intranet for "remuneration packages," they would likely get back only documents that describe HR policies in general, and not be able to access information on employees' salaries.

**Linguistic optimization tools can be applied both on the index side and at query time. The choice of where to use these tools comes down to performance, practicality, and flexibility considerations.**

Advanced use of QR processing capabilities would see organizations employing custom query transformations, which would automatically modify the query in order to improve recall or precision. An example would be geo-encoded searches, where the user is looking for results within a preferred distance of a particular location. Another example: automatic acronym transformation, where "IBM" would also return results for "International Business Machines" and I.B.M.

It is also possible to request that a query term string is transformed in case there are no returned hits from the original query. In this case, the modified query term string is returned so that a result page may inform the user of the performed transformation.

On the results-processing side, advanced scenarios would try to improve the usability of the search interface, especially if thousands of results are returned. In this case, an organization may opt to cluster results or analyze or sort them. For example, the query "BMW dealership" on an Internet Yellow Pages site might be set up to return the categories and number of hits – for instance, "BMW dealers (12), BMW garages (42), BMW repair shops (17)," etc.

In an enterprise environment, organizations may blend results from multiple systems (this is known as a federated search) so that users can get results from corporate file systems, or from the intranet, extranet, or Internet. In this case, the user interface becomes critical to simplifying the viewing and consumption of information.

Q: I need very high QPS on my e-commerce site. How can I achieve this?

A: Query rate capacity is limited mainly by the number of search rows – multiple search rows will provide linear scaling of QPS. The number of query and result servers is important when using result processing features. You should limit the result-side features (dynamic duplicate removal, result clustering, etc.) to optimize for the number of QR servers needed.

Alerting functionality is becoming popular with seasoned search users who prefer to have information pushed to them rather than searching for it each time. Here, multiple filter conditions (or triggers) are matched to an incoming stream of fresh data in the form of events such as news articles, stock quotes or other documents processed by the search application. Alerts or filtered content streams are provided in real-time and converted to appropriate applications or end-user devices via XML.

## Different industries, different solutions

It is apparent that many facets of an enterprise search system need reasonable consideration before they are used in business-critical environments. So what does query and results processing mean for your business?

An investment bank will have vastly different knowledge discovery or search needs compared to an oil and gas manufacturer. A bank needs tightly integrated security models to ensure that employees on either side of "Chinese Wall" (equity research teams and corporate finance teams, for instance) are granted access to only the information they have permission to receive. Oil and gas manufacturers also need to consider security, but not on the same level as the bank.

In an e-commerce environment, the business model focuses on sales volumes, so e-commerce providers have to pay particular attention to the processing and presentation of results. If a search was for "PDA", for example, linguistic processing would be performed ("PDA" is translated as "Personal Digital Assistant") before the search application returns results. The results would be categorized by PDA brand, category, price, type, color, availability, etc., to allow simple navigation. Information/entity extraction would be used to ensure that consumers can complete a purchase in as few clicks as possible. High QPS and performance will have to be considered in this environment. If consumers have to wait more than five to 10 seconds for a results page, they might defect to a competitor.

OEM integrations of search need to consider how much linguistic processing to perform on queries, and the level of results processing necessary for different user scenarios. OEM applications should utilize the standard functionality of the search system and interface directly with it, to leverage the administration and reporting tools that will allow configuration and tuning as needed.

## Understanding the impact of QR processing

Different search contexts call for different response profiles, and different enterprise objectives dictate different response parameters. However, most search engines use a fixed-ranking relevancy model, which is acceptable only when the search solution is used in the context it was designed for. It's far better –consistently superior – to integrate linguistic and result- processing capabilities into a holistic and adaptive approach.

The holistic aspect means applying linguistic analysis across the board – documents, queries, results, and navigation – to maximize the contribution that such technology makes. The adaptive aspect means using the right components, leveraging industry terminologies, and tuning the ranking model to match the type of search application – from broad uses like a general information portal to specific solutions like shopping sites.

As noted on the previous page, the main objectives of QR processing are to turn a potentially bad query into a good query, and to present the best results in the form that's most helpful to the user. The search provider can develop modules for specific query analysis and results processing in order to offer a personalized user experience. The front-end search can be configured to select particular rank profiles and present the results in certain ways if the user group is known. For example, a medical application might process and display results differently for doctors compared to the results offered to nurses or to medical students.

It's interesting and worthwhile to note that the search consumer may not always be a person; it may be another IT application. In that case, the search application owner will manage and maintain the query and the results processing stages, with specific frameworks for post-processing that allow the results to be presented in formats that are consumable by other IT systems.

It is rare to get a perfect search environment from the outset, so business managers will have to monitor the functional specifications, and over time refine the QR processing stage (using control and administration tools) to meet the evolving needs of the user base.

Trade-offs associated with QR processing include the points at which processing is performed – in the core of the search engine, in the QR stage, or on the client side. Typically, it is most efficient to offload as much of the processing to the search platform as possible rather than trying to post-process all of the results. This becomes particularly important if you are federa-ting results from other content sources or applications.

The filtering of results, such as removing duplicates or limiting the results, can be carried out per the terms of the query (using the search application's query language and constraints), or it can be done by the client-side application. For example, searching a sports category may limit only the documents written in English or Spanish.

To achieve the best results, in terms of relevancy and speed, it is worth letting the search application do the work to return as few highly relevant results as possible.

## Mini case study
### Enterprise storage vendor uses search as a competitive differentiator

**Who**
Worldwide provider of enterprise storage solutions

**Challenge**
To provide value-added features and functions to storage solutions using advanced search capabilities

**Solution**
Provision of unique "charge-back" model for storage managers in large organizations. Ability to handle large content volumes (50 million-plus documents) and provide high-availability configuration, integrated administration, and security.

**Technology**
Advanced query/results processing features, advanced administrative APIs, and integration with third-party data and custom code.

## Guidelines and recommendations

Generally, the metrics used to evaluate QR processing include QPS (queries per second), the number of results returned per query, and the relevancy of these results as judged by real users. The query load performance is measured as the maximum QPS number that the system can process with acceptable response times.

The QR processing overhead and query response time need to be quantified – both perceived and actual volumes. The algorithms applied at query and response time need to be measured for speed and efficiency, especially relative to the time that the core engine spends in search.

Among the mistakes most commonly made when organizations develop search applications are the failure to understand the query processing stage, and the attempt to perform too many calculations across a large result set during result processing. Loading the system beyond its query rate capacity will generate a backlog of queries, which creates higher query latencies and possibly disruption of service. Depending on where the system bottleneck is, the backlog may cause timeouts and retransmits. This in turn generates even higher loads on the search service, markedly degrading service.

The query rate capacity is limited mainly by the number of search rows, with multiple search rows providing a near-linear scaling of QPS within the search engine. It is also constrained by the number of servers used for query and results processing, which becomes especially important when using result-processing features. It is recommended that you plan for increasing query loads and upgrade the system accordingly.

It is crucial to understand the performance of the query API on the client side. In some situations, the query API can perform a substantial amount of parsing and processing – for example, it can include result clustering and navigators with multiple buckets. Search users should understand this and design the client API appropriately.

Q: My CMS application features a search engine, but it currently displays lists of results. How can I improve the usability without degrading performance during query/result processing?

A: Consider using the analytical capabilities of the search solution – entity extraction on unstructured content and/or dynamic drill-down for structured data. This will allow you to provide navigational search. To protect performance, ensure that the search platform carries out as much of the processing as possible, and try to minimize post-processing of results. Increasing hardware (if appropriate) can offset the additional load on the system.

Finally, the feature set has a strong impact on the effective QPS. Result-side features such as dynamic duplication removal, clustering of results, and result-side navigators will add substantially to the load on the query/result servers. Increasing the level of hardware can offset the load on the system. Other features like deep navigation and full wildcard support will also add load to the search nodes.

## The fundamental steps for improving search

It's best to take a phased approach when developing your search application. This will allow you to identify what works well and to isolate areas for improvement. It is useful to leverage the standard query processing options such as synonym expansion and automatic query rewriting.

With QR processing, it is advisable to do as much as you can in the core of the search application. Increasing the number of search rows allows you to enhance the speed of processing and boost scalability in a near-linear fashion. Ideally, search providers should offload as much of the processing to the search platform as possible rather than trying to post-process all of the results, which can create latency issues. Here, the system should use deep navigators and avoid result-side (shallow) navigators; that way, the search tools will be less likely to return more results than is necessary per query.

When returning results, it is useful to understand the result volume, since most search applications can return the top N results or the entire result set. This is a trade-off between speed and satisfying the need for all information. The more data returned, the more time it takes to stream it back to the search client.

It is vital to understand the impact of relevancy. Most users and applications will need only a small subset of the entire result set if the relevancy model is adapted to their needs. Organizations are urged to understand the impact and cost of features that operate on an extended result set – features such as results clustering.

In an ideal environment, the best performance and consistency will come from having all content indexed by the search application. This isn't always possible, however, so users should be aware of the impacts of federated or blended searches such as mixed relevancy or throughput.

From an OEM perspective, it is advisable to leverage the query API of the search system rather than relying on HTTP. Generally, the query APIs and connectors will provide a rich and robust wrapper around the underlying HTTP interface to the search engine. This makes it easier to work with the search application and provides additional capabilities such as error checking and an administration interface for reporting.

### Frequently asked questions

Q: What's a rank profile?

A: The relevancy of a document with respect to a query is represented by a ranking value. A rank profile concept enables full control of the relative weight of each component for a given query (e.g., How important is the title relative to the body of the article?). This enables individual relevance tuning of different query applications.

Q: Can I return information that resides outside of the search application?

A: Federating or blending results is possible, but you have to consider the consistency of the result relevance and throughput. These issues can be solved with additional hardware and by tuning of the relevancy model and the associated rank profiles.

Q: What is query transformation?

A: It refers to the analysis and subsequent rewriting of a query – typically linguistic transformations such as lemmatization and spell checking. If need be, you can also plug in custom query transformation stages.

Q: What is results transformation?

A: This is the algorithmic processing of search results. It includes result-set reordering (e.g., duplicate removal), adding navigation information (e.g., clustering/drill-down), and result content conversion or reformatting.

Q: Can I pass results from the search system to a third-party application?

A: Search systems should be able to return results in the format that you require – text and/or XML. You have to understand the downstream consumer's needs so that your application returns information in a suitable form.

Q: Is it possible to customize the QR processing stage?

A: It is possible to augment and enrich the frameworks associated with the QR processing stage for your application needs. Typically, this will require custom modifications. It's advisable to seek expert advice to design and build the right solution in the shortest time.

## About FAST SBP™ (Search Best Practices)

SBP consulting is a highly focused transfer of search knowledge and experience from FAST to its prospects and customers. SBP workshops aim to help enterprises realize the full potential of search, by creating optimal strategic, functional and technical roadmaps, delivered in the form of business model, solution and architecture designs.


**Fast Search & Transfer**
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





SWP.008.B.01.011206