
# DOE SBIR Phase II Final Report

## "Distributed Relevance Ranking in Heterogeneous Document Collections"

## *Executive Summary*

This report contains the comprehensive summary of the work performed on the SBIR Phase II project ("Distributed Relevance Ranking in Heterogeneous Document Collections") at Deep Web Technologies (http://www.deepwebtech.com). We have successfully completed all of the tasks defined in our SBIR Proposal work plan (See Table 1 - Phase II Tasks Status). The project was completed on schedule and we have successfully deployed an initial production release of the software architecture at DOE-OSTI for the Science.gov Alliance's search portal (http://www.science.gov).

We have implemented a set of grid services that supports the extraction, filtering, aggregation, and presentation of search results from numerous heterogeneous document collections. Illustration 3 depicts the services required to perform QuickRank™ filtering of content as defined in our architecture documentation. Functionality that has been implemented is indicated by the services highlighted in green.

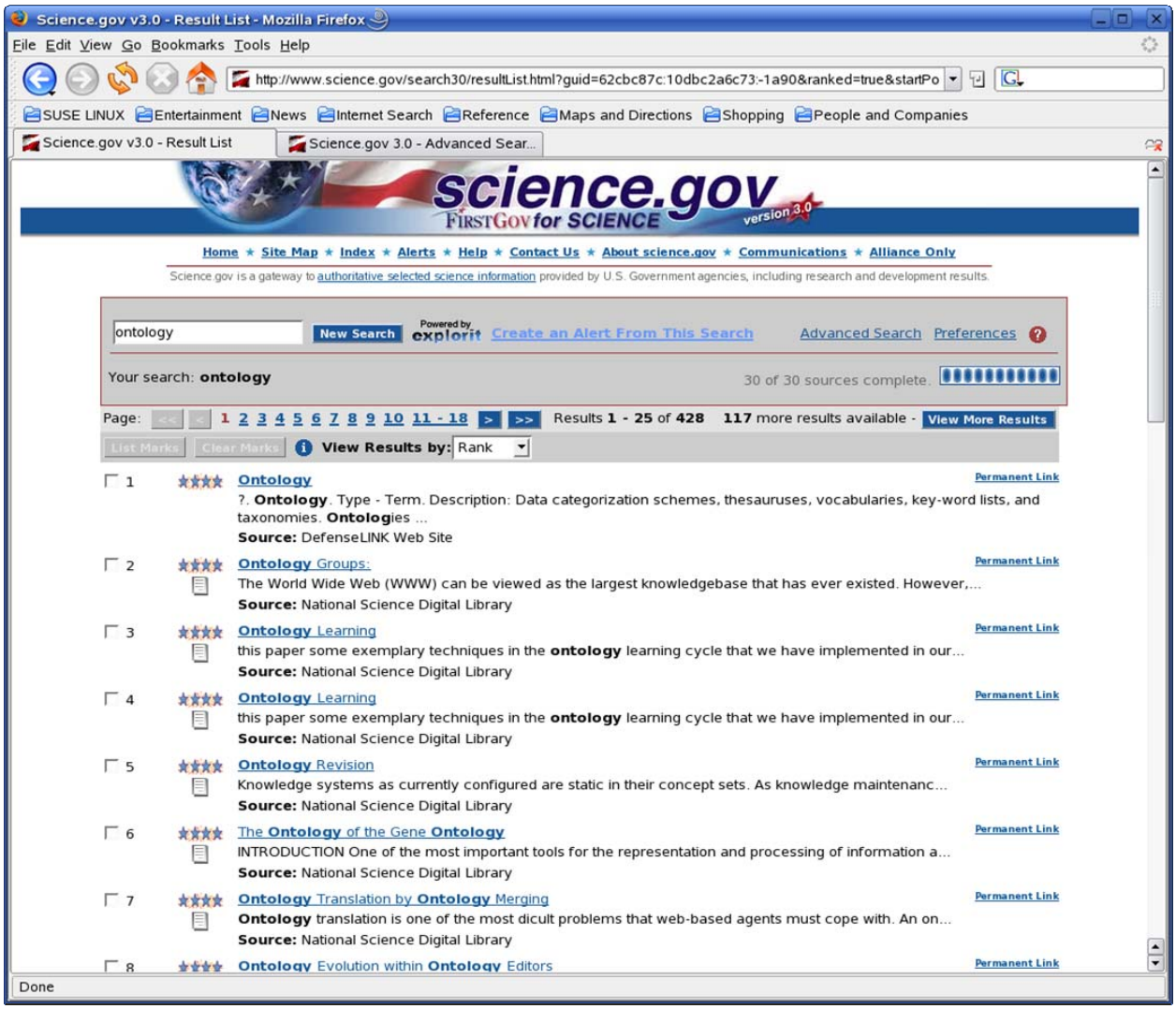
We have successfully tested our implementation in a multi-node grid deployment both within the Deep Web Technologies offices, and in a heterogeneous geographically distributed grid environment. We have performed a series of load tests in which we successfully simulated 100 concurrent users submitting search requests to the system. This testing was performed on deployments of one, two, and three node grids with services distributed in a number of different configurations. The preliminary results from these tests indicate that our architecture will scale well across multi-node grid deployments, but more work will be needed, beyond the scope of this project, to perform testing and experimentation to determine scalability and resiliency requirements.

We are pleased to report that a production quality version (1.4) of the science.gov Alliance's search portal based on our grid architecture was released in June of 2006. This demonstration portal is currently available at http://science.gov/search30 . The portal allows the user to select from a number of collections grouped by category and enter a query expression (See Illustration 1 - Science.gov 3.0 Search Page). After the user clicks "search" a results page is displayed that provides a list of results from the selected collections ordered by relevance based on the query expression the user provided.

Our grid based solution to deep web search and document ranking has already gained attention within DOE , other Government Agencies and a fortune 50 company. We are committed to the continued development of grid based solutions to large scale data access, filtering, and presentation problems within the domain of Information Retrieval and the more general categories of content management, data mining and data analysis.

*Illustration 1 - Science.gov 3.0 Search Page*


We provide uniform access methods to these heterogeneous collections where both access method and format of the resulting content vary across collections. In addition we provide efficient application of a variety of content filters and aggregation of content. Finally we address the more general problems of flow control, performance, usability, and administration of large scale distributed solutions to federated search and relevance ranking.




## *Grid Service Layers*

The system architecture is partitioned into a set of layers with each layer responsible for a specific set of functionality. These layers include; User/Usability, Work Flow, Filter, Data Access, and Data Source. These layers are depicted in Illustration 2 - High Level Architecture Diagram.

*User/Usability (Research Assistant)*

The User/Usability layer contains the services that interact with the user and involve issues of usability. It interacts with services in the Work Flow layer to turn search requests into jobs and to receive aggregated search results from all queried sources.

*Work Flow (Ranking Conductor)*

Services in the Work Flow layer involve jobs creation, scheduling and management. The work flow of the overall search is managed. Sources are optionally selected and search results are aggregated. This layer interacts with the User/Usability layer to receive job input (query string plus an optional list of collections to search) and to present it with aggregated results. This layer also interacts with the Data Access layer that queries content sources.

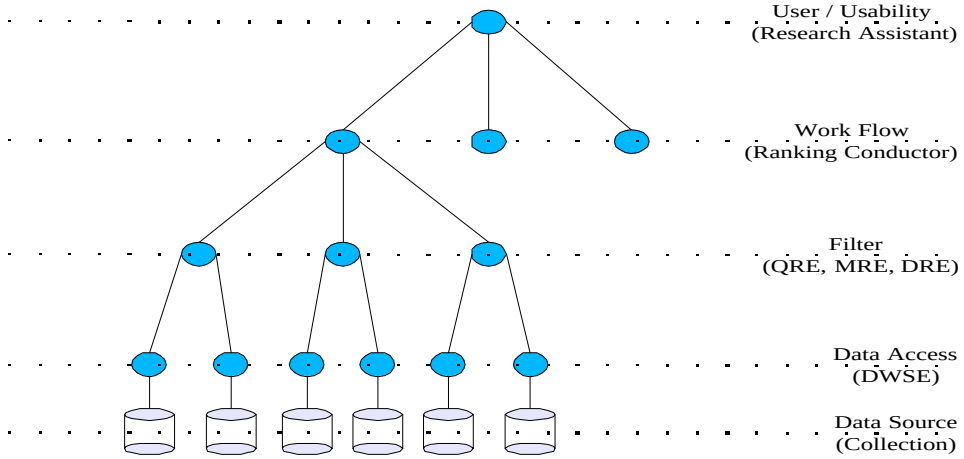



*Illustration 2 - High Level Architecture Diagram*

*Filter*

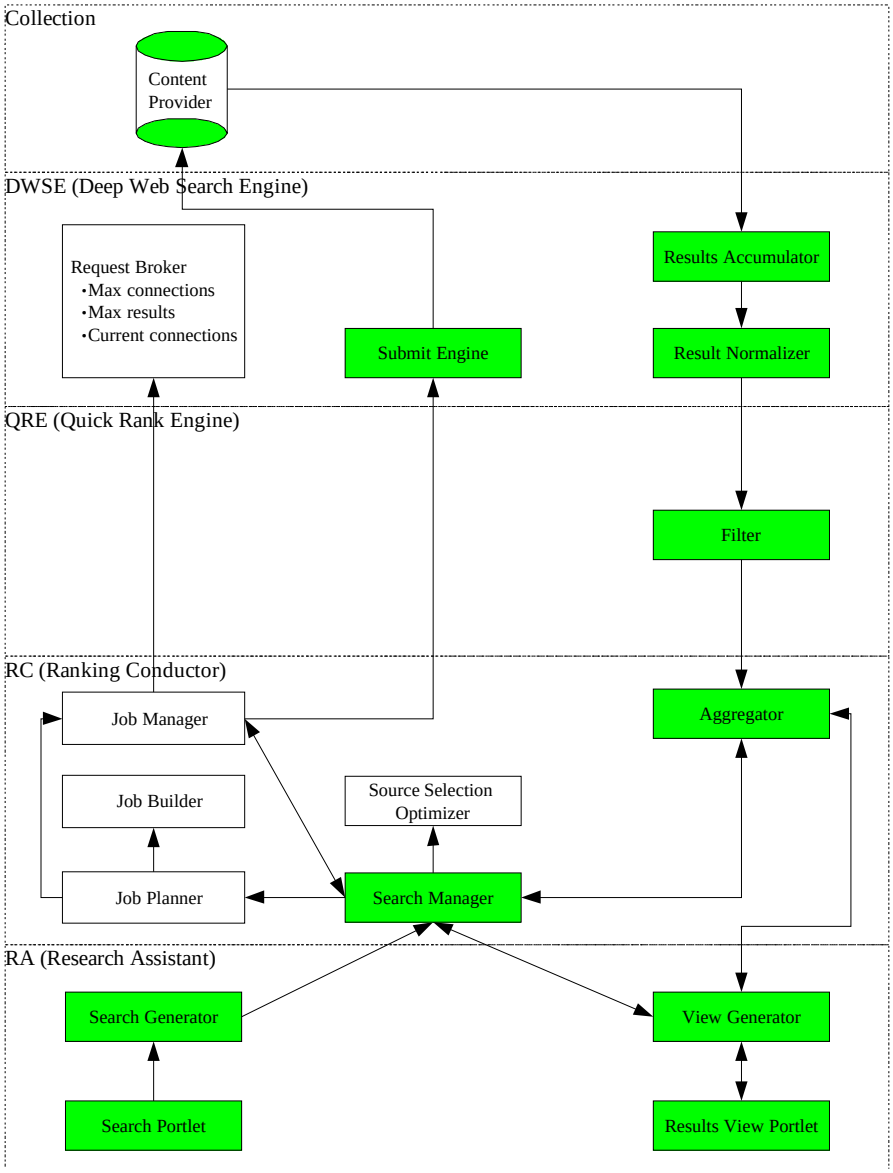
The Filter layer contains, in this SBIR, ranking engines. In the general case it may contain other data processing filters. Quick Rank, Meta Rank, and Deep Rank are processed via their own filters. As the processing flow of Quick Rank is different than that of Meta Rank we document the two in separate flow diagrams. The Filter layer receives normalized search results from the Deep Web Search Engine layer, filters the content, then passes it on to Results Aggregator service in the Work Flow layer.

*Data Access (Deep Web Search Engine)*

The Data Access layer is responsible for direct data access with a data source. It determines first that the data source is not overloaded with search requests then interacts directly, through the Submit Engine, with the data source. Search results are accumulated and normalized, then sent to the appropriate content filter in the Filter layer.

*Data Source (Collection)*

The Data Source layer consists of the deep web content accessed by the system. It is only accessed by the Data Access layer, whose submit engines know how to access the collections. The Data Source layer returns results to the Results Accumulator in the Data Access layer.




## *Ranking Algorithms*

During the course of this SBIR we developed three unique ranking algorithms. Each of these algorithms is implemented as a grid Filter Service which takes a normalized result set as input, assigns some relevance metadata to each result based on its contents, and provides this relevance

metadata as output.

Quick Rank is designed to quickly rank documents based solely on the information available in the terse results from a deep web search engine. It ranks documents based on title and snippet. The highest rank is given to documents the earlier the search terms occur in the title. Higher ranks are also assigned to documents that match the search terms and contain shorter titles. Additionally, a higher rank is assigned to top-level web pages. Quick Rank ranks each document outside of the context of the rank of any other document. The grid services involved performing a Quick Rank search are depicted in Illustration 3 - Quick Rank Service Interactions. Services that have been implemented as of this time are highlighted in green.

Meta Rank operates on documents containing highly structured text (meta-data.) Unlike Quick Rank, Meta Rank goes beyond Quick Rank in that it retrieves documents to which the results list points. Documents suitable to Meta Rank processing include some web pages, short articles, and short reports. Meta Rank identifies zones in a document in order to assign weights based on location of search terms. Documents are retrieved and indexed into a collection for searching.

The most sophisticated and resource intensive ranking algorithm developed during this project is Deep Rank. This algorithm retrieves the complete document text for each record returned from the collections. These documents are put into a normal form that can then be applied to relevance and semantic analysis filters. The quantity of data and complexity of the computation required to perform this level of search make Deep Rank a good candidate for a grid-based solution.

## *Overview of Accomplishments*

*Highlights*

1. Operational deployment of a multi-node production grid to a Fortune 50 corporation.
2. A fully operational prototype was developed that successfully validated the architecture detailed in our Architecture and Requirements documentation. This prototype was built upon the production quality Apache Axis technologies and was composed of a number of production quality services that performed the work in a fully-realized implementation of the software architecture.

1. Operational deployment of a fully operational grid implementation of the core functionality needed to realize “Distributed Relevance Ranking in Heterogeneous Document Collections” has been successfully developed, tested, deployed and used by thousands of users. This implementation supports the extraction, filtering, aggregation, and presentation of search results from numerous heterogeneous document collections. Illustration 3 depicts the services required to perform ranking and filtering of content as defined in our architecture documentation. Functionality that has been implemented is indicated by the services highlighted in green.

2. We have successfully tested our implementation in a multi-node grid deployment. We have tested deployment of the implemented grid services in a number of different configurations across multiple node grids within the DWT offices.

3. We have performed a series of load tests in which we successfully simulated 100 concurrent users submitting search requests to the system. This testing was performed on deployments of one, two, and three node grids with services distributed in a number of different configurations. The preliminary results from these tests indicate that our architecture will scale well across multi-node grid deployments.

4. We have implemented sophisticated Boolean search expression support into the ranking and filtering services. This enhancement allows complex Boolean logic to be applied across all collections regardless of the individual collection's level of Boolean support.

5. The search results interface now supports the display of incremental results. Results stream into the Aggregation services from numerous lower-level services. The user is informed of the number of new results available and is given the option to fetch these additional results from the Aggregation service. This is all accomplished through an intuitive web page based interface.

6. An asynchronous web services interface has been implemented that allows services to be defined to run either synchronously or asynchronously depending upon the specific requirements of the service.

7. A mechanism has been implemented to allow for the creation and management of persistent grid services. Each service may create one or more persistent resources each with a specified TTL. Resources may be persisted beyond their initial TTL through keep alive messages, or by adjustment of the TTL value. The abstract Web Resource may be extended to perform any unit of work required and to persist state on any required data.

8. A workflow management service has been implemented that evenly distributes the work for a search request amongst each of the nodes configured.

9. Science.gov 3.0 (Explorit version 1.4) released ( www.science.gov/search30 ).

*Implemented, Tested, and Deployed Grid Services*

1. A Submit Engine service that transforms a normalized search request into a form suitable for the underlying collections. The transformed request is submitted to the underlying collection, results are retrieved, normalized, and passed on to additional services for further processing.

2. The QuickRank™ relevance ranking algorithm has been ported to a Services based implementation and has been enhanced to support sophisticated indexing and application of complex Boolean logic.
3. MetaRank™ has been implemented to allow search of rich metadata from heterogeneous collections.

4. An Aggregator service that merges the results from numerous instances of the QuickRank™ filter service into one comprehensive result set.

5. A Collection Manager Service that allows each service within the deployment to discover the document collections available for searching and extract relevant information on each collection.

6. A Service Registry service that allows each service within the deployment to discover the types of services deployed and the locations of each service within the deployment.

7. A Resource Manager Service that allows for the creation and management of stateful and persistent grid services. Asynchronous service execution is also supported by this implementation.

8. A Search Manager Service that initiates and monitors the overall state of each search within the deployment.

9. A Results Generator service that calls upon the Aggregator service to determine the number of results available for retrieval and to retrieve these results. It also makes use of the Resource Manager Service to display the overall progress of the current search.

10. Search Generator service that provides the functionality necessary for a user to initiate a search request. The service calls upon the Collection Manager service to extract meaningful information on the document collections currently available, presents the available collections to the user, processes the users input (selected collections and query expression), and submits the users request to the Search Manager service.

## *Future Work*

Among the major goals accomplished by this SBIR was the development of initial implementations of the QuickRank™, MetaRank™, and DeepRank™ filtering services. The initial implementations of these services have validated the use case requirements but require a deeper analysis and evaluation to improve their usability for all content collections that are currently supported by Deep Web Technologies content access functionality. Although we have developed initial versions of these services, more work is needed to enhance the production quality, scalability, and resiliency of these services.

## *Phase II Tasks Table*

The table below lists each task item defined in our Phase II proposal and the expected time line for their completion. Items that have been completed are highlighted in green, items that are currently in development are highlighted in red, and items to be addressed in the future are highlighted in blue.

| ***Task*** ***Month*** | ***1*** | ***2*** | ***3*** | ***4*** | ***5*** | ***6*** | ***7*** | ***8*** | ***9*** | ***0*** | ***1*** | ***2*** | ***3*** | ***4*** | ***5*** | ***6*** | ***7*** | ***8*** | ***9*** | ***0*** | ***1*** | ***2*** | ***3*** | ***4*** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Create Research Assistant detailed design documents | | | | | | | | | | | | | | | | | | | | | | | | |
| Build infrastructure to support core Research Assistant functions | | | | | | | | | | | | | | | | | | | | | | | | |
| Build Deep Web Search Engine | | | | | | | | | | | | | | | | | | | | | | | | |
| Build QuickRank Search Engine | | | | | | | | | | | | | | | | | | | | | | | | |
| Grid-enable Deep Web Search Engine and Results Repository | | | | | | | | | | | | | | | | | | | | | | | | |
| Build and Test MetaRank Engine | | | | | | | | | | | | | | | | | | | | | | | | |
| Re-architect ed Distributed Explorit and deploy Science.gov 3.0 | | | | | | | | | | | | | | | | | | | | | | | | |
| Build and test DeepRank Engine | | | | | | | | | | | | | | | | | | | | | | | | |
| Build and test Source Selection Optimizer and supporting databases | | | | | | | | | | | | | | | | | | | | | | | | |
| Add sophistication to Ranking Conductor | | | | | | | | | | | | | | | | | | | | | | | | |
| Build tests and deploy grid-based demonstration system. | | | | | | | | | | | | | | | | | | | | | | | | |

| ***Task*** ***Month*** | ***1*** | ***2*** | ***3*** | ***4*** | ***5*** | ***6*** | ***7*** | ***8*** | ***9*** | ***0*** | ***1*** | ***2*** | ***3*** | ***4*** | ***5*** | ***6*** | ***7*** | ***8*** | ***9*** | ***0*** | ***1*** | ***2*** | ***3*** | ***4*** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Document, test and deploy science.gov 4.0 | | | | | | | | | | | | | | | | | | | | | | | | |

*Table 1 - Phase II Tasks Status*