

## An analysis of the 'build vs. buy challenges for GridServer reporting and chargeback

There are several approaches to build a reporting solution for Tibco (DataSynapse) GridServer using traditional reporting or BI tools, the most common of which would be to create reports directly using the GridServer reporting database. It is of our opinion that commercial tools such as Evident Clearstone for GridServer maintain the following advantages over this approach, for the reasons identified below:

1. Product development efforts are spread across many clients. In the twelve months ending September 2008, the reporting portfolio has grown from ~12 to over 40. Over ten man-years went into development of Evident ClearStone's first release (aka DataSynapse 'VersaVision.')
2. Support costs are known and consistent, and little customer support is required. The vendor is responsible for testing and troubleshooting.
3. The source of the data is optimized for data mining and queries in high transaction volume environments. Such data is maintained in a robust and open data warehouse that is optimized for scalability, performance, and extensibility. The data model is published for use with third party tools.
  - a. By the end of 2008, vendor tools will be deployed across over 100 grids.
4. The product is designed to be used by various operating groups, with are user-friendly and easily accessible reports based on highly refined and processed data.
5. The level of effort associated with adding support for chargeback models has been consistently under-estimated by other grid users.
6. The product platform is designed for cross domain analysis outside of the grid services environment (adding system level information, data caching, etc.)

Users who are considering an internal build should be aware of several technical challenges, including the following:

1. The GridServer reporting database schema is not ideal for data mining or analysis for the following reasons:
  - a. The database is not an optimized for sophisticated SQL queries. It lacks additional dimensions like application, business unit mappings, timezones, SLA's, etc.
  - b. The database does not use indexes, as it is optimized for storage of real-time events, and not their retrieval for reports or analysis.
  - c. The database schema is not designed to take advantage of database partitioning features, such as Oracle tablespaces or MS SQL Server file groups.

- d. Querying tasks for a specific service session involves a table scan of the task table (which can be voluminous). The TASK table does not contain the proper index to support this type of query.
  - e. Querying for a specific engine's performance over time also would not utilize any indices, since the ENGINE\_STATS table does not contain the proper index to support this type of query.
  - f. By way of example, consider a basic report to determine the running task count for each hour of the day per grid service. The GS reporting database does not provide this type of summary. Instead, task records start time and end time are provided, leaving the rest to the reporting tool to, for instance, determine which tasks ran concurrently during each hour, summarizing the information by grid service and hour, etc. This requires complex SQL or ETL's (Evident ships with over 200 ETL's).
  - g. A well designed data warehouse designed for reporting and analysis should provide summary tables that across dimensions like JOBS, TASKS, Time of Day, Day of week, etc. Without these summaries, it is difficult to perform common tasks such as comparisons of current trends with history, what if analytics for Capacity Planning, etc.
  - h. Contention for reading/writing to database table will exist in medium/heavily loaded environments. This may cause problems for GS writing records to the database or query performance latency.
  - i. Most GS reporting databases only keep data for a few days to weeks (depending on the GS configuration). Therefore, for longer term reports there may not be sufficient data retention.
  - j. Even if there was sufficient data retention, the data is "raw and granular." The data is stored at a granular level without any summarizations. Therefore producing hourly, daily, weekly, or monthly reports will involve significant database CPU time to perform aggregations, data analytics, etc.
  - k. For heavily loaded environments, the volume of data produced may be overload some reporting tools. Without a proper data warehouse design, database partitioning, and database ETL implementation, it would be very difficult for a reporting tool to scale.
  - l. Relying solely on datetime functions in the database (i.e. MS SQL Server) to determine task durations is not 100% reliable due to known precision issues with MS SQL Server.
  - m. Not all GridServer data can be trusted. There are anomalies in GS reporting database that will produce erratic results which affect report accuracy.
2. Reporting Tool Concerns
- a. In a multi-grid environment, there will be multiple GS reporting databases. Cross grid reporting will be challenging since the data resides in separate databases.
  - b. The reporting tool also must be able to run reports across one or more databases.

- c. Automating scheduled of reports.
- 3. Service Mapping
  - a. For service oriented reports, there may be a need to group multiple grid services into a common application for report presentation. This capability allows users to view consolidated reports. To support this in a traditional reporting tool, the grid service names or service types would have to be hard coded in the reports or queries, requiring additional effort for each new report or service.
  - b. If the customer requires organizational reporting on grid utilization, this logic has to be designed and developed. This then enables mapping grid activity to users based on the grid service, engine allocation, broker, etc.
  - c. Grid chargeback/billing requires the ability to define pricing models, producer/consumer mapping and business models, all correlated with grid usage. This was a one year development effort for initial release.
  - d. SLA reporting based on grid job/task performance: A reporting tool would “hard-code” these business rules into the queries and reports for each SLA policy by service, which implies customizing reports for each different grid service and SLA. It is possible that users will want notification of SLA violations. Reporting tools are not designed for monitoring and alerting.
- 4. Other dependencies
  - a. There is configuration data that would be helpful to obtain outside of the GS database, specifically, information about engines and grid services that is available via web services APIs. Most reporting tools don’t support querying data via web services APIs. Also, this information is dynamic, requiring the development of an additional collection mechanism in order to correlate this with the GS database.
  - b. The reporting schema can change across multiple releases, thus having another layer of complexity when there are multiple versions of the GRID, and the solutions having to provide multi-grid reporting.