Knowing Sooner Rather Than Later:

Cost Analysis of Low-Latency Data in Enterprise Data Warehousing

Richard Hackathorn and Jack Garzella

Version 11 – January, 2007

1. Executive Summary 2

2. Managing Your Business in a Global Economy 3

The Time-Value Curve 4

3. Architectures for Data Acquisition 6

Case A – Daily Batch Updates 6

Case B – Intra-Day Batch Updates 7

Case C – Continuous Batch Updates 7

Case D – Continuous Stream Updates with Extra-Database Transform 8

Case E – Continuous Stream Updates with Intra-Database Transform 9

4. Cost Assumptions 10

Sizing of Data Warehouse 10

Platform for Transaction Systems 10

Platform for Database Warehouse 10

Platforms for Development and Disaster Recovery 11

5. Cost Estimations 12

Case A – Daily Batch Updates 12

Case B – Intra-Day Batch Updates 12

Case C – Continuous Batch Updates 12

Case D – Continuous Stream Updates with Extra-Database Transform 13

Case E – Continuous Stream Updates with Intra-Database Transform 13

Summary of Costs 14

6. Recommendations ..... Error! Bookmark not defined.

7. Appendix A – Cost Estimation Table 17

8. Endnotes 18

BOLDER Technology, Inc.

1. Executive Summary

Knowing about your business sooner, rather than later, provides a competitive edge.

In today's rapidly changing global economy, a critical issue facing every corporation is the freshness of their business data. Knowing about your business sooner, rather than later, provides a competitive edge and enables you to better manage through unexpected situations.

This is an issue debated by both business executives and Information Technology professionals because of the close intertwining of business issues and technical factors.

This study explores the costs of the freshness of business data within the enterprise data warehouse. In particular, we focus the study on the time delay in acquiring data from transaction systems into the data warehouse. The intent of this educational whitepaper is to enable IT professionals to ask the right questions and for business executives to understand what is technically possible.

There is a widespread belief that near real-time data is too expensive for most business applications. This belief is no longer valid in many situations because of rapid evolution of data acquisition technology. This study concludes that the cost of continuous stream data acquisition is similar to upgrading current ETL technology to mini-batch processing.

With the trend toward operational applications closely linked into the data warehouse, business requirements for low-latency data will continue to increase. This paper provides practical insights into how evolve your enterprise architecture to support those future requirements.

2. Managing Your Business in a Global Economy

Business happens fast in the global economy. And each month, it is happening faster. Broadly defined, Business Intelligence (BI) is a critical capability that enables a corporation to manage its business based on the facts of its business activity. In the past, BI focused on strategic planning and historical reporting. This will remain essential, but the focus of BI has shifted to business operations.

Recent studies have shown that there is a significant business requirement for data 'freshness' that is less than 24 hour. Technically this is referred to as *data latency*, which is the time interval from a business event (such as, an order placed by customer) to business response (such as, shipping the order to the customer). There is a growing requirement that data latency be measured in minutes to support critical operational processes.

What do we do with what we know – now?.

The implication is that there is business value when we can do something different in our business today, rather than waiting until tomorrow or next week. The key question is: What do we do with what we know – now?

Let's consider an example. A large retailer releases a special promotion for a limited time and monitors the sales lift of their campaign. For instance, there were many special promotions during the holidays for one or two days.

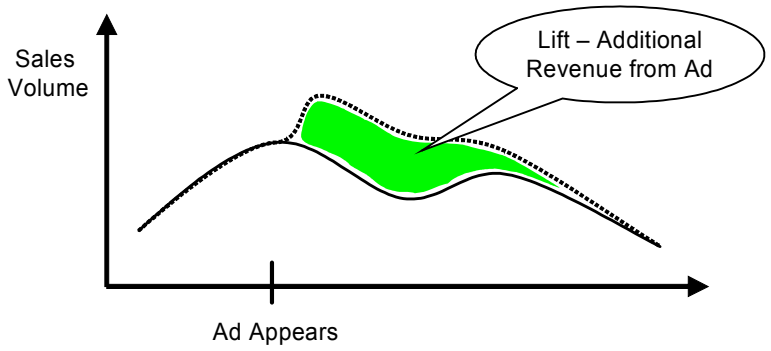

Figure 1: Life in Sales Volume

As shown in Figure 1, the vertical scale is sales volume in dollars per time interval, while the horizontal scale is the timeline. The solid curve is the historical sales for the item that is being promoted. The dotted line is the actual sales volume.

If this is a traditional campaign in which material is mailed to prospects, the timeline may be a week. What do we do with this analysis? Probably adjust the next campaign that will be launched next week or month for a similar item.

If this is an email or web campaign (in which a promotion window pops up on the home page), the timeline is maybe just a few hours. And, what do we do with what we know about the sales volume? The answer depends on when we obtain this analysis?

For traditional BI, we would obtain the analysis the following day. In this case, nightly, the data warehouse would be loaded with the data from today's sales. We would obtain this analysis the following morning. So, what would we do? Probably adjust the pop-up promotions for the next day in an attempt to increase the resulting lift. Note that there is nothing we can do about yesterday's campaign. It is history!

For operational BI, we would want to obtain the analysis as soon as technically possible. The data warehouse would be loaded continuously by streaming updates from the sale system so that the data is ready for analysis within minutes of the sale event on the website. We would watch the analysis as a dashboard display that is constantly refresh minute by minute.

So, what would we do differently? There are now many more options. Within an hour or less, we could determine whether the promotional campaign was successful as compared with similar campaigns. If successful, we could let the campaign run as is. If the campaign is too successful with an unusual high volume, we may have set the price too low. An immediate price increase may boost overall profitability with similar sales volume. If the campaign is faltering with a volume that is historically too low, we could lower the price, change the messaging or offer free shipping. Note that all of these options are intended to improve the current campaign.

By incorporating low-latency data into business processes, there are more decision options available.

By incorporating low-latency data into business processes, there are more decision options available. We have actually changed those processes (or at the very least increased their frequency). The full business value of low-latency data is realized when the corporation willing to change their business processes by managing to a new set of standards and enabling their staff with new skills.

The Time-Value Curve

The principle of the business impact of low-latency data is illustrated by the Time-Value Curve, as shown in Figure 2.

The horizontal scale is the time from a business event to an action event. The vertical scale is the business value of responding with an action to the business event. In general, business value decays with time. The decay could be days or weeks; however, in our global economy it is often measured in minutes.

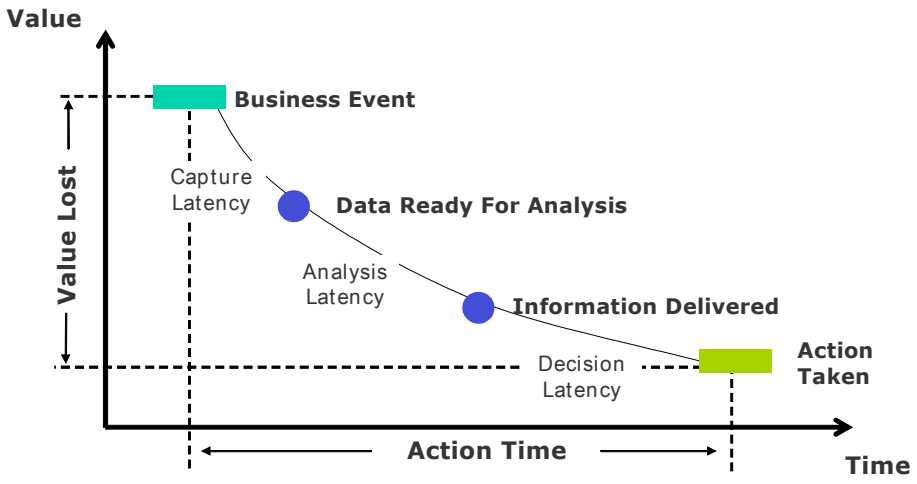

Figure 2: The Time-Value Curve

There are three steps in processing business information. First, we capture the data and acquire it into the data warehouse so that the data is ready for analysis – *capture latency*. Second, we analyze the data, packaged the reports properly, and deliver to the person (or program) that will make a decision – *analysis latency*. Third, we make a decision and take an action – *decision latency*.

This whitepaper concentrates on the first step involving capture latency. In particular, we describe the architectures that are commonly used in the industry to acquire data into the data warehouse and then perform a cost analysis of these architectures.

3. Architectures for Data Acquisition

From discussions with managers of several large-scale data warehouse installations, we determined that there are five general architectures[1] for data acquisition (DA), which capture (or extract) data from transactional systems (the source systems), move, transform, and load this data into the data warehouse.

Often the label Extract-Transform-Load (ETL) has been used. However, technology to acquire data has expanded considerably in recent years. Data propagation technology is often used under the label of Enterprise Application Integration (EAI), and Data federation technology is more recently used with the label Enterprise Information Integration. Regardless, these architectures perform the function of acquiring data from transactional systems into the data warehouse.

For this whitepaper, we assumed a 'typical' enterprise data warehouse (EDW). In particular, these architectures are for a company with annual revenues over a billion dollars and thousands of customers and products, operating in a generic product/service industry. The EDW is mature having an enterprise-wide scope and has been in production for several years. It has daily feeds from a hundred tables derived from several internal source systems. These tables are transformed into a similar number of tables within the data warehouse. There are several hundred users who depend on the EDW for their daily work. Hence, this EDW is mission-critical to the business.

The key factor that drives the architecture is the latency (time delay) to acquire the changed data from the transaction systems, process it, and flow it into the data warehouse. As a transaction is executed in the operational transaction system, new data is generated and committed to the operational databases. At some time, this changed data must be extracted from the source system, transformed (filtered, merged, etc.), and loaded into the data warehouse. We assumed that acquisition latency clusters into four categories: daily updates, intra-day updates, continuous batch updates, and continuous stream updates. The primary difference among these categories is whether data acquisition is performed as periodic batch updates or as continuously stream updates.

The architectures for each case are described below.

Case A – Daily Batch Updates

This case is the base case for a typical EDW architecture and assumes traditional ETL processing for daily batch updates, along with various weekly/monthly/quarterly updates. The daily update is executed during a 'batch window' (a period during which the system is relatively idle).

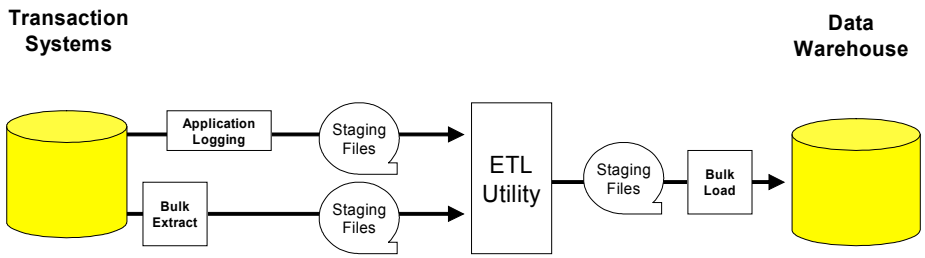

Figure 3a: Daily Batch Updates

The figure shows two ways of extracting data from source systems using custom application logging or bulk extract utility. This extracted data is staged for ETL processing, which is then staged for bulk loading into the EDW.

Case B – Intra-Day Batch Updates

Using the same ETL architecture as Case A, this case performs intra-day updates that are more frequent than daily, occurring several times during the business day. Generally, intra-day updates are every four to six hours (or four to six times per day). The source systems must be able to support data capture during normal business hours while executing the normal transaction load. Many transaction systems are not designed to handle those extracts during peak demands on the source systems and on the network traffic. Significant hardware changes are often required to avoid performance degradations.

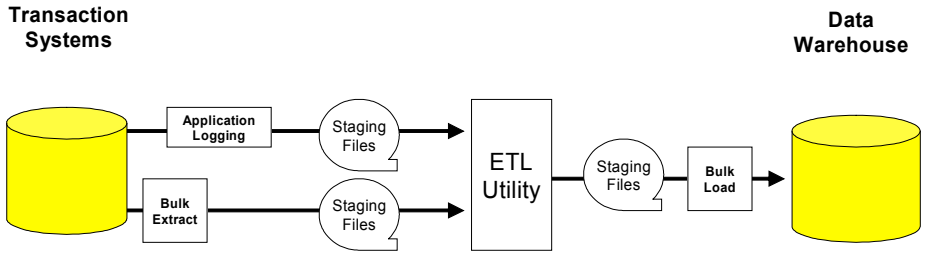

Figure 3b: Intra-Day Batch Updates

Case C – Continuous Batch Updates

Using a variation of the ETL architecture, this case uses change-data capture techniques to stream the data from source systems. Instead of bulk extract, continuous batch updates captured as transactions are committed and are then processed in mini-batches of 20 to 100 transactions. The objective is to flow the changed data into the data warehouse within 10-30 minutes.

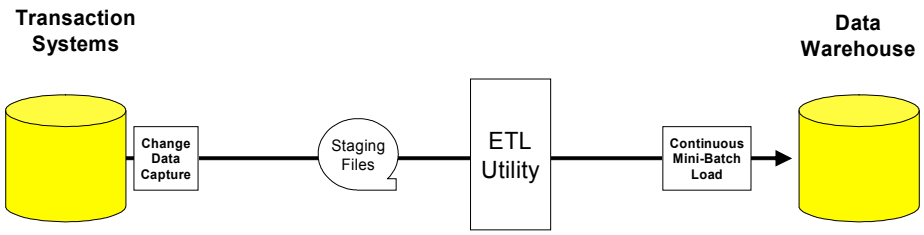

Figure 3c: Continuous Batch Updates

The data extraction from the source systems is performed using change-data capture techniques, which could have a major impact on legacy systems using older file management techniques. Data is supplied through application logging or from the database transaction log cache to ensure minimal performance impact to the source system. ETL processing is performed using mini-batches into the EDW. We assume that the ETL tool efficiently supports 'mini-batches' to stimulate continuously stream processing.

Case D – Continuous Stream Updates with Extra-Database Transform

This case streams changed data directly from the source systems into EDW staging tables within seconds (i.e., near real-time). Continuous stream updates are driven by the commit events of transactions in the source systems. Then, transforms are performed using ETL processing externally to the database (extra-database transforms). A possible problem is that delays of several minutes may occur in the extra-database transform because the data must be copied into external files, processed by the ETL utility, and then reloaded into user tables.

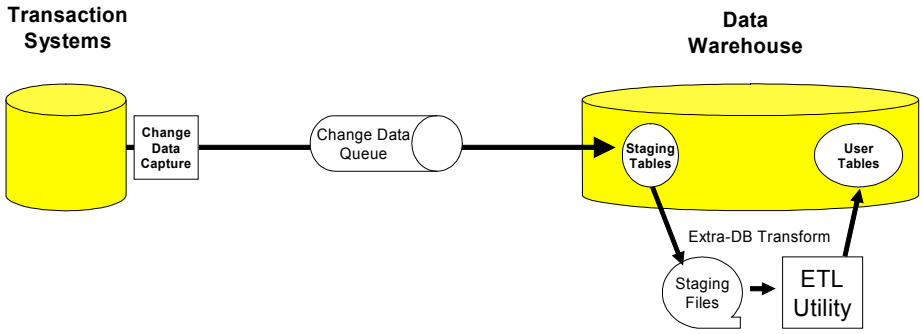

Figure 3d: Continuous Stream Updates with Extra-Database Transforms

Case E – Continuous Stream Updates with Intra-Database Transform

As in Case D, this case also streams changed data directly from source systems into EDW staging tables in near real-time. Transforms are then performed internally to the database (intra-database). The data latency is often quite small. One large retailer had a latency averaging 11 seconds with delays of 30 seconds to two minutes during peak loads, across 180 data feeds. A possible problem is that intra-database transform places an increased workload on the EDW.

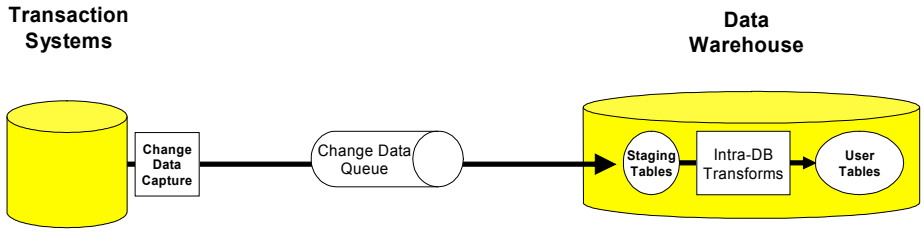

Figure 3e: Continuous Stream Updates with Intra-Database Transforms

4. Cost Assumptions

The following assumptions were made about the data acquisition architectures.

Sizing of Data Warehouse

We assumed that the EDW would contain approximately five terabytes of user data in one hundred tables, generating about 100 GB per day in constantly changing raw data (date, numbers, expanded textual fields, etc.). There is a single data center so that data flows are within the internal LAN without the need for WAN services, such as OC3 lines.

As this data flows from the source systems to the data warehouse, it may have to be stored in one or more times in flat files or other formats. We assumed that the standard ETL architecture would store such data twice, once before the cleanse/transform processing and once after. This intermediate data flow and storage affects the cost of storage capacity and network bandwidth.

Platform for Transaction Systems

We assumed that the transaction systems were independent of the DA architecture and, therefore, associated costs are not included in the cost comparison. For all cases, transaction systems are assumed to be an 8-way server if non-Intel platform or a 4-way server if an Intel platform. Daily batch extracts are performed during a batch-window during periods of low business activity (generally at night), thus no additional cost was required within these systems. In addition, we assumed that current changed data capture technology was sufficiently efficient (only 3% to 5% performance loading) that no additional costs were required in the source systems.

The weakest assumption was in Case B, where batch extracts are being performed four to six times per day during normal transaction loads. Most transactional systems will not allow batch data extracts to occur during peak daytime loads. This implies that the cost of Case B is conservative and an additional cost item for upgrades to the source systems may have to be added in some situations.

Platform for Database Warehouse

We assumed that the data warehouse was also independent of the DA architecture and, hence, associated costs are not included in the cost comparison. The data warehouse is typically an 8-way server with DB/2 and Oracle or a 4-node Teradata system.[2] We assumed that the bulk loads to the EDW were performed during

periods of low activity. And, we assumed that the trickle feed into the EDW did not impose a significant impact on EDW performance. Finally, we assumed that the in-database transforms in Case E could be performed at a low priority so as to not impact EDW performance.

Platforms for Development and Disaster Recovery

As shown in Figure 4, there are multiple platforms for the typical enterprise data warehouse. We have focused on the production version. However, there are often two (or more) EDW platforms that must be kept updated with data. The development EDW must have a realistic subset of data to adequately test new processes. The disaster recovery EDW must also be kept current.

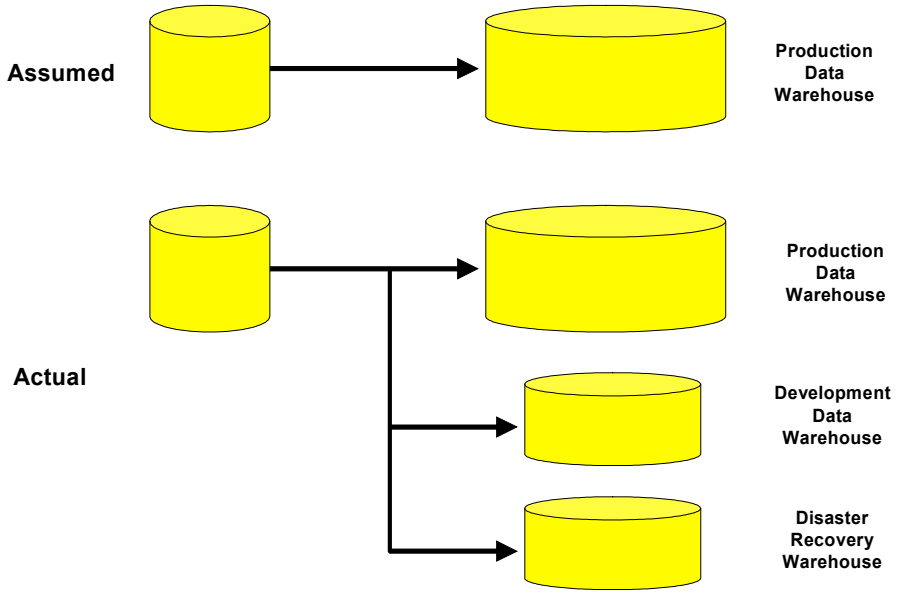

Figure 4: Assumed versus Actual Architectures

In this analysis, we did not include these extra platforms. In actual situations the cost of these additional platforms must be included, which is likely to magnify the cost differences.

5. Cost Estimations

Based on the above assumptions, we built a detailed spreadsheet that listed the significant cost components of the five data acquisition architectures, as shown in Appendix A.

Case A – Daily Batch Updates

This case is the base case with daily batch updates using traditional ETL processing. The major expenses for initial development are:

- ETL tool licensing: $500K for a full suite from a third-party data integration vendor.
- Network hardware: $280K to support the flow of 500 GB/day. This amount was difficult to estimate because of all the factors involve with data center networks. We did assume that the network was a LAN internal to the data center and that a WAN was not required. The same estimate was used for all cases so that the differences should not be biased, which is typically not the situation in actual implementations.
- Managed storage: $135K for 9 TB at $15 per GB, fully loaded

The major expenses for annual operations are:

- Labor: $270K for two persons in development and operations.
- Software maintenance: $100K as 20% of initial cost.

The total expenses are $1,025K initially and $469K annually.

Case B – Intra-Day Batch Updates

Based on Case A, this case performs updates three to six times per day. As incremental to Case A, the additional expenses for initial development are:

- Additional ETL tool licensing: $250K for a second license with a 50% discount from the original cost since it is a redundant server for intra-day activity versus a fully engaged server

The additional expenses for annual operations are:

- Labor: $120K for one person in operations, and $75K for a half person in on-going development

The total expenses are $1,339K initially and $726 annually.

Case C – Continuous Batch Updates

Based on Case A, this case uses change-data capture techniques to stream the data from source systems. As incremental expenses to Case A, the additional expenses for initial development are:

- Data capture software: $100K for a utility on 4-cpu server, or the same expense in custom developed software.
- ETL software: $250K licensing for redundant server
- Training & Professional Services: $50K for expertise to configure the initial system, and $30K for three-weeks of training for two persons

The additional expenses for annual operations are:

- Labor: $150K for an additional developer, and $120K for an additional operations person

The total expenses are $1,513K initially and $831K annually.

Case D – Continuous Stream Updates with Extra-Database Transform

This case streams data directly from the source systems to the EDW and then performs transformations using ETL processing externally to the database (extra-database transforms). The major expenses for initial development are:

- Data capture software: $100K for a utility on 4-cpu server, or the same expense in custom developed software.
- Extra-Database transform utility: $500K for full-function ETL utility
- Network hardware: $280K to support the flow of 500 GB/day, as in Case A.
- Labor: $75K for professional services, and $75K for initial development

The major expenses for annual operations are:

- Software maintenance: $100K for data transform utility, and $20K for data capture
- Labor: $150K for an additional developer, and $120K for an additional operations person

The total expenses are $1,084K initially and $455K annually.

Case E – Continuous Stream Updates with Intra-Database Transform

This case also streams data directly from source systems to the EDW but then performs transformations internally to the database (intra-database transforms). The major expenses for initial development are:

- Data capture software: $100K for a change-data extract utility on 4-cpu server, or the same expense in custom developed software.
- Intra-database transform utility: $90K for a SQL-generator utility.
- Network hardware: $280K to support the flow of 500 GB/day, as in Case A.

- Labor: $75K for professional services, and $75K for initial development

The major expenses for annual operations are:

- Software maintenance: $18K for data transform utility, and $20K for data capture
- Labor: $150K for an additional developer, and $120K for an additional operations person

The total expenses are $643K initially and $367K annually.

Summary of Costs

The cost summary of the five cases is shown below.

Architecture		Initial Expense		Annual Expense		3-Year Cost	
			Incremental From A		Incremental From A		Incremental From A
Case A	Daily Batch Updates	$ 1,025		$ 469		$ 2,432	
Case B	Intra-Day Batch Updates	$ 1,339	$ 314	$ 726	$ 257	$ 3,516	$ 1,084
Case C	Continuous Mini-Batch Updates	$ 1,513	$ 488	$ 831	$ 362	$ 4,005	$ 1,573
Case D	Continuous Stream Updates with Extra-DB Transform	$ 1,084		$ 455		$ 2,449	
Case E	Continuous Stream Updates with Intra-Data Transform	$ 643		$ 367		$ 1,744	

The initial and annual costs are shown in the first two columns. For Cases B and C, the incremental costs from Case A are shown and used to calculate the total costs. The third column is the three-year system cost (i.e., initial + 3*annual) for the five cases.

Note that these scenarios and costs are 'typical' and are probably not valid for any particular business. The authors encourage readers to use this model as a template to evaluate their actual costs. Examine the labor costs closely as the need for special skills are often underestimated, especially for Cases A-B-C.

6. Conclusions

This section concludes by suggesting recommendations for the preferred data acquisition architecture, dependent on the following factors:

Factor 1: **< 24 hours?** Will the business require data latency of less than 24 hours? The implication is whether there is or will be business requirements for more than daily updates.

Factor 2: **< 4 hours?** Will the business require data latency of less than four hours? The implication is whether there is or will be business requirements for very current data, which can not be supported by current ETL technology.

Factor 3: **New EDW?** Is this an established EDW installation with daily ETL updating (instead of a new installation)?

Factor 4: **Mini-Extracts?** Can the source systems support mini-batches extracts during peak periods? Can the current transactional systems handle the increased workload from frequent batch extracts? Note that this may require transaction processing be quiescent during the extracts.

Factor 5: **Mini-Loads?** Can the ETL utility support mini-batch loads into the EDW during peak periods? Can the ETL processing be efficiently decomposed into small batches?

Factor 6: **Intra-DB?** Can intra-database transforms be support on the EDW database platform without impacting query performance? Does the EDW platform have sufficient processing capability to perform adequately?

Decision tree below shows our recommendations, given these factors.

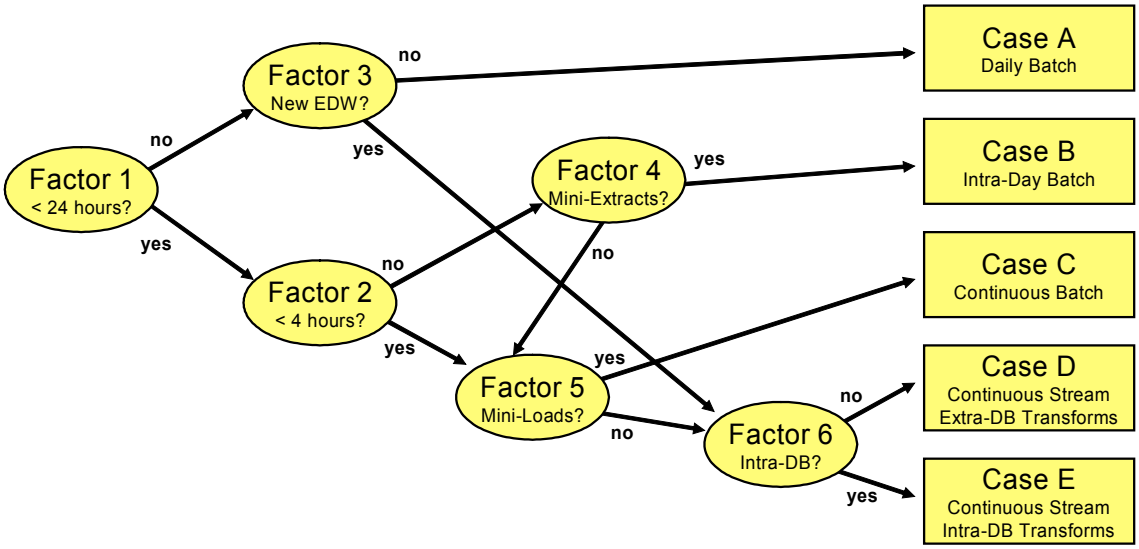

Figure 5: Recommended Architectures for Data Acquisition

The decision tree suggests the following key recommendations.

1. Continue with Case A if there is no business requirement for data with latency of less than 24 hours.
2. Adopt Cases D or E if you are developing a new EDW. They have the lowest system cost while having the extra benefit of supporting near real-time updates.
3. Adopt Cases D or E if there is a business requirement for data with latency less than 4 hours.[3] The system cost of Case E is the same as the incremental cost of Case C. Further, the system cost of Case D is roughly the same as Case C since the ETL data transformation utility is already owned.
4. Adopt Case B if there is no business requirement for data with latency less than 4 hours. However, carefully examine the performance impacts of frequent mini-batch extracts upon current transaction systems.

- - - - -

The accepted industry wisdom about low-latency data is rapidly shifting.

In conclusion, technology advances are changing the data acquisition architectures for enterprise data warehousing. The accepted industry wisdom about low-latency data is rapidly shifting. In many situations, an architecture that supports continuous stream updates with near real-time data feeds has become the low-cost alternative.

The cost analysis showed that the typical architectures for Cases A-B-C have high costs when pushed toward near real-time latency. Further, performance impacts of mini-batch processing on legacy systems are often prohibitive as the requirement for data latency drops to less than four hours.

... innovative business processes enabled...by 'smart & fresh' data.

It is exciting to see the emergence of innovative business processes enabled by a new generation of enterprise applications driven by 'smart & fresh' data. This is changing the whole complexion of Business Intelligence in enterprises, as they cope with the rigors of the global economy.

7. Appendix A – Cost Estimation Table

Architecture		Initial Development Expense		Annual Operating Expense	
Case A - Daily Batch Updates					
	server hardware	$ 30	4-cpu Intel server	$ 6	20% of initial purchase
	software licensing	$ 500	ETL tool licencing	$ 100	20% of initial licencing for maintenance
	storage hardware	$ 135	200 GB/day kept 45 days, $0.015/MB or $15/GB	$ 27	20% of initial purchase
	network hardware/configuration	$ 280	need 500 GB/day = 5 MB/s within center	$ 56	20% of initial purchase
	training	$ 30	3 wk class for 2 p at $15/p	$ 10	1 wk class for 2 p at $5/p
	professional services	$ 50	8 wk at $25/month		
	development labor			$ 150	1 p at $150 fully burdened
	operations labor			$ 120	1 p at $120 fully burdened
	TOTAL	$ 1,025		$ 469	
Case B - Intra-Day Batch Updates (incremental costs from Daily Batch Updates)					
	server hardware	$ 30	second 4-cpu Intel server	$ 6	20% of initial purchase
	software licensing	$ 250	ETL tool licencing (50% less for 2nd licence)	$ 50	20% of initial licencing for maintenance
	network hardware/configuration	$ 28	20% additional network support	$ 6	20% of initial purchase
	professional services	$ 6	1 wk at $25/month		
	development labor			$ 75	0.5 p at $150 fully burdened
	operations labor			$ 120	1 p at $120 fully burdened
	source system costs	-tbd-			
	NET INCREASE	$ 314		$ 257	
	Expenses from A	$ 1,025		$ 469	
	TOTAL	$ 1,339		$ 726	
Case C - Continuous Mini-Batch Updates (incremental costs from Daily Batch Updates)					
	server hardware	$ 30	4-cpu Intel server	$ 6	20% of initial purchase
	software licensing	$ 100	data capture software	$ 20	20% of initial licencing for maintenance
	software licensing	$ 250	ETL tool licencing (50% less for 2nd licence)	$ 50	20% of initial licencing for maintenance
	network hardware/configuration	$ 28	10% additional network support	$ 6	20% of initial purchase
	training	$ 30	3 wk class for 2 p at $15/p	$ 10	1 wk class for 2 p at $5/p
	professional services	$ 50	8 wk at $25/month		
	development labor			$ 150	1 p at $150 fully burdened
	operations labor			$ 120	1 p at $120 fully burdened
	NET INCREASE	$ 488		$ 362	
	Expenses from A	$ 1,025		$ 469	
	TOTAL	$ 1,513		$ 831	
Case D - Continuous Stream Updates with Extra-DB Transform					
	server hardware	$ 30	4-cpu Intel server config mgt, capture, ETL transformation	$ 6	20% of initial purchase
	software licensing - data movement	$ 100	data capture software	$ 20	20% of initial licencing for maintenance
	software licensing - transformation	$ 500	data transformation utility	$ 100	20% of initial licencing for maintenance
	storage hardware	$ 16	150 GB/day trail files (plus staging) for 7 days plus $0.015/MB or $15/GB	$ 3	20% of initial purchase
	network hardware/configuration	$ 280	need 500 GB/day = 5 MB/s within center	$ 56	20% of initial purchase
	training	$ 8	1 wk class at $8/wk on transform design		
	professional services	$ 75	3 mo at $25/month		
	development labor	$ 75	3 mo at $150/yr for 2 p	$ 150	1 p at $150 fully burdened
	operations labor			$ 120	1 p at $120 fully burdened
	TOTAL	$ 1,084		$ 455	
Case E - Continuous Stream Updates with Intra-DB Transform					
	server hardware	$ 10	2-cpu Intel server w min disk for config mgt, capture, load	$ 2	20% of initial purchase
	software licensing - data movement	$ 100	data capture software	$ 20	20% of initial licencing for maintenance
	software licensing - transformation	$ 90	intra-database transform utility	$ 18	20% of initial licencing for maintenance
	storage hardware	$ 5	100 GB/day trail files w 50% compression for 7 days, $0.015/MB or $15/GB	$ 1	20% of initial purchase
	network hardware/configuration	$ 280	need 500 GB/day = 5 MB/s within center	$ 56	20% of initial purchase
	training	$ 8	1 wk class for 2 p at $8/wk on transformation design		
	professional services	$ 75	3 mo at $25/month		
	development labor	$ 75	3 mo at $150/yr for 2 p	$ 150	1 p at $150 fully burdened
	operations labor			$ 120	1 p at $120 fully burdened
	TOTAL	$ 643		$ 367	

8. Endnotes

[1] In reality, corporations will have some mixture of these architectures depending on their history, environment, and industry.

[2] Since each Teradata node has two processors, all platforms have eight processors.

[3] Note that Cases D and E can be mixed with Cases A-C to add near real-time data feeds if only a few data feeds require intra-day latency.

- - - - -

We appreciate the support from GoldenGate Software to pursue this study.

Dr. Richard Hackathorn is president and founder of **Bolder Technology, Inc.** (BTI) in Boulder, Colorado. BTI is a thirteen-year old consulting and education firm specializing in the Information Technology industry.

Richard has over thirty years of experience in the IT industry as a well-known technology innovator and international educator. He has pioneered many innovations in database management, decision support, client-server computing, database connectivity, data warehousing, and web farming. He founded MicroDecisionware Inc. (MDI), an early vendor of database connectivity products that was acquired by Sybase in 1994.

Richard has published numerous articles in trade and academic publications, presented regularly at leading trade conferences, and conducted professional seminars in eighteen countries. He writes for the Business Intelligence Network and has written three professional texts, entitled Enterprise Database Connectivity, Using the Data Warehouse (with W.H. Inmon), and Web Farming for the Data Warehouse.

For twelve years, Richard was a professor at the Wharton School of the University of Pennsylvania and at the University of Colorado. He received his B.S. degree in Information Science from the California Institute of Technology and his M.S. and Ph.D. degrees in Information Systems from the University of California, Irvine.

He can be contacted at richardh@bolder.com

Mr. Jack Garzella Is President of JMG Software Engineering, a firm specializing in IT and data architectures, data warehousing and management consulting with regard to technology. Jack recently managed the building of Overstock's active data warehouse that supports marketing, finance and merchandising for over 300 users. Many applications were built one the data warehouse including their CRM Email marketing system, reporting and dashboards, partner billing as well as others. It was one of the first operational data warehouse with over a hundred near real time data feeds.

Previously, Jack was with Teradata where he led the Application Solutions and Professional service teams. Prior to Teradata, Jack was VP of IT at MatchLogic running their CRM and analytics teams. Jack has also worked for Oracle and other marketing and system integration firms over the last 20 years. Jack received his B.S. in Computer Science from Purdue University and has continued is professional development with specialized courses in Product management, business administration and general IT management

He can be contacted at via email at jack@jmgse.com