

Knowing Sooner Rather Than Later: Cost Analysis of Low-Latency Data in Enterprise Data Warehousing

Richard Hackathorn and Jack Garzella
February 2007

1. Executive Summary	2
2. Managing Your Business in a Global Economy	3
The Time-Value Curve	5
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	13
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. Conclusions.....	15
7. Appendix A – Cost Estimation Table.....	17
8. Endnotes	18

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 organization is the freshness of its business data. Knowing about your business sooner, rather than later, provides a competitive edge and enables you to better manage through unexpected situations.

With the rising importance of operational applications linked into the enterprise data warehouse, business requirements for low-latency data within the data warehouse will continue to increase. This study provides practical insights into how to evolve your enterprise architecture to support those future requirements.

Sponsored by GoldenGate Software, the study explores the cost factors of acquiring low-latency data from transaction systems into the enterprise data warehouse. The study explores five typical architectures for data acquisition and estimates the cost factors for each. Although subjective estimates were used in this study, they are based on first-hand experiences with several large systems in production today.

The objective of this study is to educate information technology professionals to ask the right questions and for business executives to understand what is technically possible. The objective is *not* to form firm conclusions based on our estimates. Instead, we suggest a framework and method for evaluating the tradeoffs in real situations. The Excel spreadsheet used in this study is available from the authors to be modified to your actual configuration. You can then draw conclusions that are valid for the unique situation of your company.

The belief that near real-time data is too expensive may be no longer true for many situations with current technology.

Low-latency data is data about your business that is less than one day since the underlying business event has occurred. Most businesses operate on the basis of data that is one day old or greater. Some people believe that low-latency data is too expensive for most business applications. This belief may no longer be valid in many situations because of rapid evolution of data acquisition technology.

Based on our cost estimates, this study concludes that data acquisition using continuous stream technology is cost effective related to traditional technology while offering a simpler architecture and better ability to support 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. Going forward, these applications will remain essential, but BI has expanded to include minute-by-minute business operations.

Recent studies have shown that there is a significant business requirement for data ‘freshness’ that is less than 24 hours. This low-latency data is increasingly available through online reports or alerts. This is referred to as *action time*, 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.

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? In other words, what can we do differently to improve our business if we have proper information available immediately?

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, the retailer ran special promotions during the holidays over one or two days.

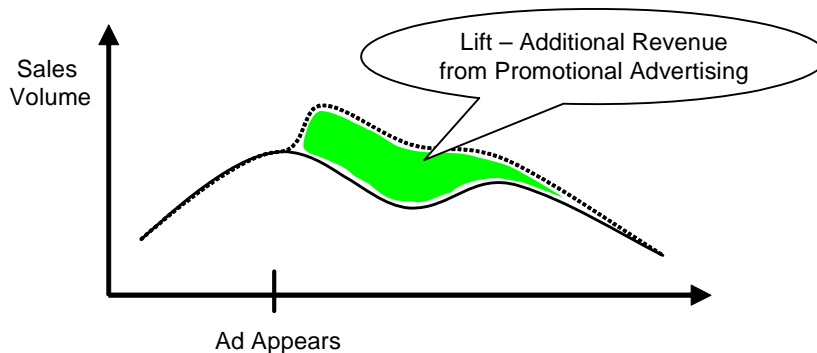


Figure 1: Lift in Sales Volume

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.

*What do we do with
what we know – Now?*

If this is a traditional campaign in which material is mailed to prospects, the timeline may be a week. The analysis would probably be used to adjust the next campaign, which would be launched next 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 just a few hours. In either case, the question is 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, the data warehouse would be loaded nightly with the data from today's sales. We would obtain the analysis the following morning. So, what would we do? Probably adjust the pop-up promotions for the next day to try to increase the resulting lift. Note that there is nothing we can do about yesterday's campaign. It is history!

If we were controlling a business process like a manufacturing assembly line, 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 refreshed minute by minute.

This illustrates how low-latency data not only speeds up the execution of a business process, but changes the business value altogether. With data loaded once per day, a two-day campaign can only be adjusted once. With continuously fresh data, the retailer can adjust constantly for the maximum returns.

The manager now has many more options. Within an hour, 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 unusually 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.

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. When the organization is willing and able to change their business processes by managing to a new set of standards and enabling their staff with new skills, the full business value of low-latency data can then be realized.

By having continuously loaded data available for BI functions, users have the opportunity to manage the business to its full potential maximizing operations, revenue and profitability.

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



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.¹ The decay could be days or weeks; however, most operational tasks are measured in minutes.

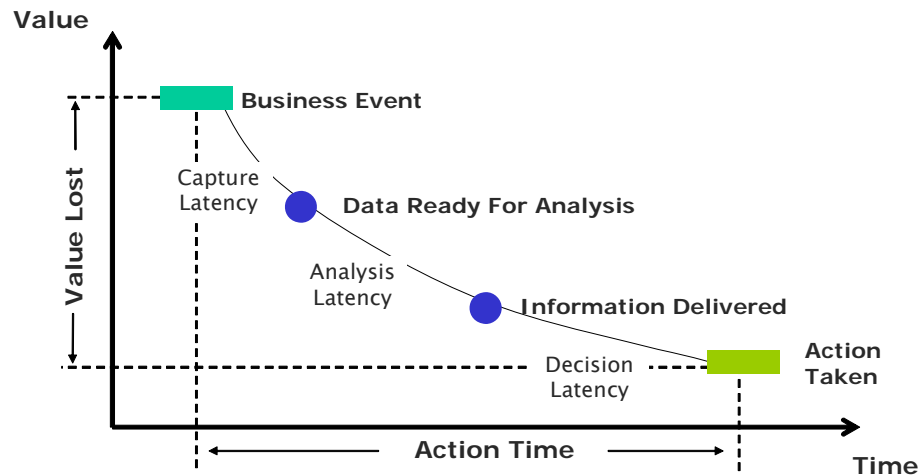


Figure 2: The Time-Value Curve

There are three steps in processing business information. First, we capture the data and insert it into the data warehouse so that the data is ready for analysis – *capture latency*. Second, we analyze the data, package 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*. The end-to-end interval from event to action is the sum of these three latencies and is called the *action time*.

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 and load data into the data warehouse, and then perform a cost analysis of these architectures.

Three Steps for BI Data:

1. Capture the Data
2. Analyze the Data
3. Make a Decision

3. Architectures for Data Acquisition

From experiences with several large-scale data warehouse installations, we synthesized five typical architectures for data acquisition. Specifically, all of these architectures capture (or extract) data from production systems (the transactional systems), move, transform, and load this data into the data warehouse.

Data acquisition is often labeled as ‘Extract-Transform-Load’ or simply ETL. However, underlying technology to acquire data has expanded considerably beyond ETL in recent years. For instance, Enterprise Service Bus technology is in vogue to support Service-Oriented Architectures. Regardless, these technologies perform data acquisition – acquiring data from production systems into the data warehouse.

We assume a ‘typical’ enterprise data warehouse² for a company with annual revenues of more than a billion dollars with thousands of customers and products, operating in a generic product/service industry. The Enterprise Data Warehouse (EDW) is mature, having an enterprise-wide scope and has been in production for several years. It has daily feeds from one hundred tables derived from several internal production systems. These updates are transformed and loaded into a number of tables within the data warehouse. There are one or two 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 load it into the data warehouse. As a transaction is executed in the production system, new data is generated and saved into the production databases. At some time, this changed data must be extracted from the production system, transformed (reformatted, cleaned up, and merged with related data), and loaded into the data warehouse.

The primary ways to acquire data are: daily updates, intra-day updates, continuous batch updates, and continuous stream updates. The primary difference in these methods is whether we use periodic batch processing or continuously stream the data into the data warehouse.

We will describe the architectures for each case here.

Case A – Daily Batch Updates

This case assumes traditional ETL processing with daily batch updates, along with various weekly/monthly/quarterly updates. The daily update is executed during a ‘batch window’, or a period during which the system is relatively idle.

Underlying technology to acquire data has expanded considerably beyond ETL in recent years.

Data Acquisition Methods:

1. *Daily Batch Updates*
 2. *Intra-Day Batch Updates*
 3. *Continuous Mini-Batch Updates*
 4. *Continuous Streaming Updates*
-



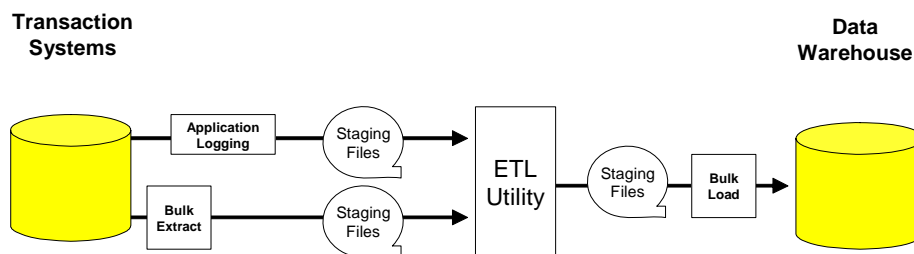


Figure 3a: Daily Batch Updates

The figure shows two ways of extracting data from production 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 occur every four to six hours.

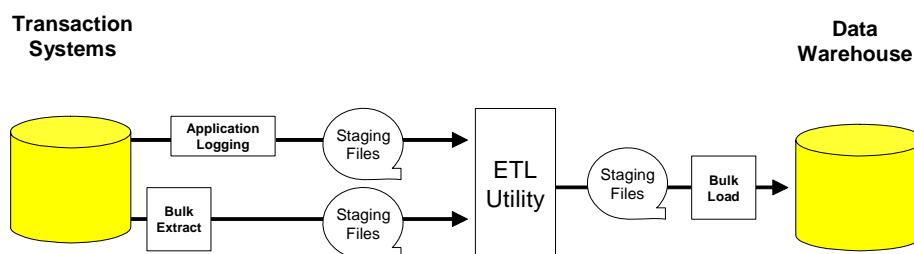


Figure 3b: Intra-Day Batch Updates

Many legacy transaction systems cannot handle the intra-day extracts, making Case B not possible.

A key concern is whether the production systems can support data capture during normal business hours while executing the normal transaction load. For many production systems, infrastructures were not designed to supply extracts during peak demands, requiring possible hardware upgrades to avoid performance degradations.

Case C – Continuous Batch Updates

Note that extra network bandwidth is required for moving the files.

Using a variation of the ETL architecture, this case uses change-data capture techniques to stream the data from production systems. Instead of bulk extract, production updates are collected over 10-15 minutes and processed as mini-batches of 20 to 100 transactions. This allows the changed data to flow into the data warehouse within 10-30 minutes.

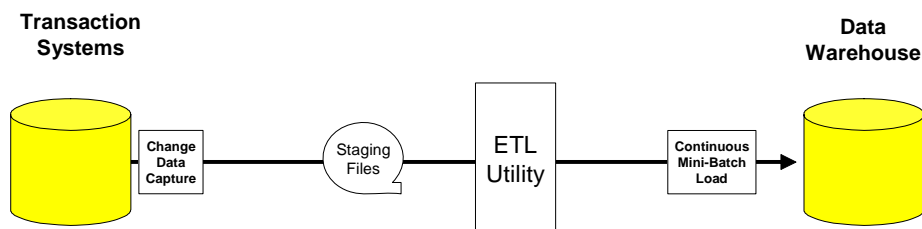


Figure 3c: Continuous Batch Updates

Many legacy transaction systems cannot detect changed data. Therefore, Case C may not be possible without application redesign to log changes in data.

The data extraction from the production 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 recovery log to reduce the performance impact to the production system. ETL processing is performed using mini-batches into the EDW.

There are two caveats for this approach. First, some legacy production systems using older file management techniques will have difficulty performing change-data-capture activities. Second, older ETL tools may not be able to process the groups of transactions that arrive erratically due to timing differences or improper sequencing of data.

Case D – Continuous Stream Updates with Extra-Database Transform

This case streams changed data directly from the production systems into EDW staging tables within seconds (i.e., near real time). By watching the recovery logs for completed transactions, continuous stream updates flow directly into staging tables within the data warehouse. At a later time, transforms are performed using ETL processing outside the database (extra-database transforms).

In Case C as opposed to Case B, we get the change data capture directly from the database logs so that only committed data is extracted from the transaction systems.

Using continuous load and data capture to a data warehouse is a significant change to most legacy data warehouse loading techniques but has proven to work well in many cases.

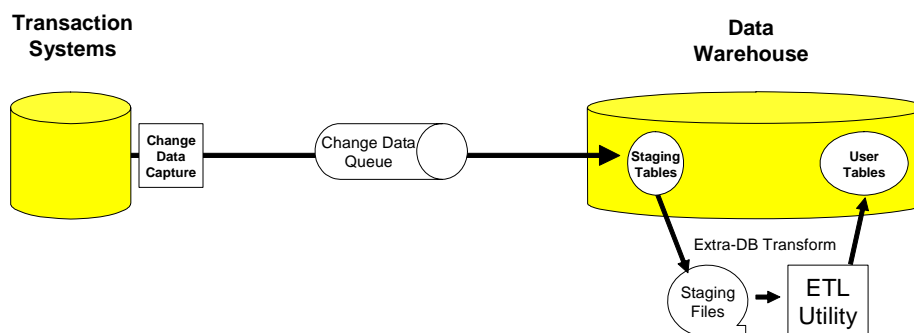


Figure 3d: Continuous Stream Updates with Extra-Database Transforms

A caveat 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. The loading process is expensive, especially since the data is loaded twice. However, extra-database transform may be necessary when the logic for cleansing and merging is complex.

Case E – Continuous Stream Updates with Intra-Database Transform

As in Case D, this case also streams changed data directly from production systems into EDW staging tables in near real time. Transforms are then performed inside the database (intra-database).³ The data latency is often quite small. For example, a large retailer had a latency averaging 11 seconds with delays of 30 seconds to two minutes during peak loads. This was accomplished while receiving incoming data from 180 production applications.

Care must be taken in this type of architecture but has been proven to work in many data warehouses.

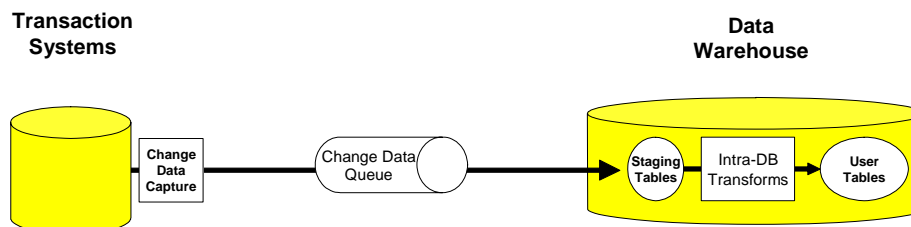


Figure 3e: Continuous Stream Updates with Intra-Database Transforms

A caveat is that intra-database transform places an increased workload on the data warehouse platform, which could reduce performance for critical reporting and querying.

4. Cost Assumptions

To develop cost models for these cases, the following assumptions were made about the data acquisition architectures.

Sizing of Data Warehouse

The EDW contains 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 production systems to the data warehouse, it may have to be stored 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 usage affects the cost of storage capacity and network bandwidth.

Platform for Transaction Systems

For all cases, the production transaction systems are separate from the data acquisition architecture and, therefore, associated costs are not included in the overall cost comparison. Transaction systems are assumed to be an 8-way server if non-Intel platform or a 4-way server if an Intel platform.

For Case A, daily batch extracts are performed within a batch-window during periods of low business activity (generally at night), thus no additional cost was required with these systems. In addition, current changed data capture technology is efficient (only 3% to 5% performance loading) so that no additional costs were required by the production systems.

For Case B, the weakest assumption in this paper occurs where batch extracts are being performed four to six times per day during normal production workload. 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 production systems may have to be added in some situations.

Platform for Database Warehouse

The data warehouse is also separate from the data acquisition architecture and, hence, its associated costs are not included in the cost comparison. The data warehouse is typically an 8-way server with DB/2, Oracle, or a 4-node Teradata® system.⁴ Bulk data loading into

Key Assumptions:

1. 100 GB per day of data
 2. Single data center
 3. Flat file extracts are stored twice, for extract and post cleansing before batch loads.
-



the EDW is performed during periods of low activity. And, we assume that trickle feed into the EDW does not impose a significant impact on EDW performance. Finally, in-database transforms in Case E are performed at a low priority so they don't impact concurrent EDW users.

Platforms for Development and Disaster Recovery

As shown in Figure 4, there are multiple platforms for the typical EDW. 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.

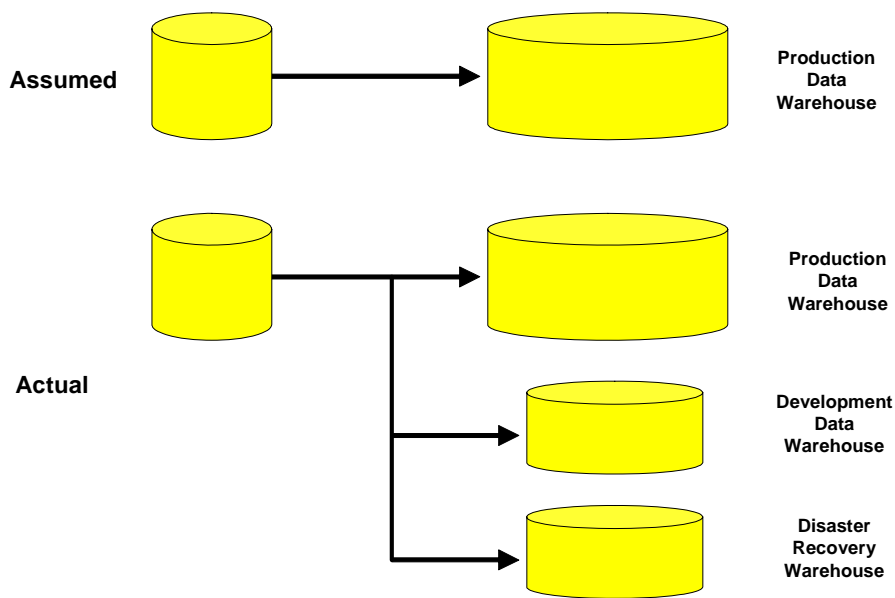


Figure 4: Assumed versus Actual Architectures

In this analysis, we did not include these extra platforms or their associated license, consulting, and other costs. In actual situations, the cost of these additional platforms must be included, which is likely to magnify the cost differences to favor Cases D and E.

5. Cost Estimations

Using the above assumptions, a cost analysis was developed for each of the data acquisition methods. A detailed Excel spreadsheet was built to compile the cost components, as shown in Appendix A.⁵

Remember that the objective of this study is to illustrate an approach for doing a cost analysis for your company. The spreadsheet is available from the authors to be modified to your actual configuration. You can then draw conclusions that are valid for your unique situation.

We encourage the reader to take our costing templates and modify them to reflect their actual environment.

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 involved with data center networks. The network was assumed to be 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 with network management capability.

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 since dual ETL processes are required as this is now critical for intra-day updates. A 50% discount from the original cost is assumed for 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.

In all cases, we assumed:

1. 100 source tables
 2. 100 GB/day of data from the transactional system.
 3. 100+ tables in DW after transformation is done.
-



Case C – Continuous Batch Updates

Based on Case A, this case uses change-data capture techniques to stream the data from production 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. This is a very conservative number if custom development is needed to capture the changed data.
- ETL software: \$250K licensing for redundant server.
- Training and Professional Services: \$50K for expertise to configure the initial system, and \$30K for three-weeks of training for two persons. This expense covers either third-party vendor training on a product or internal training of operation staff on the customer software and operations changes needed to support Case C.

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 production 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 production systems to the EDW, but then performs transformations internally to the database

Cases D and E assume that a third-party tool is used exclusively to collect the changed data and stream it into the EDW via SQL inserts or messaging techniques.



(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. As in Case D, this case exclusively uses third-party software for the change data capture.
- Intra-database transform utility: \$90K for a SQL-generator tool.
- 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

Here is the cost summary of the five cases.

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.

For Case E, implementations that were studied showed lower costs for doing intra-database transforms.

6. Conclusions

This section concludes by recommending the data acquisition architecture that is preferred under certain conditions. NOTE: Because this is based on our cost estimates for a ‘typical’ EDW, these recommendations may not be valid for any particular company. However, the following factors are key in determining the lowest cost solution:

- 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 cannot 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 production systems support mini-batch extracts during peak periods? Can the current transactional systems handle the increased workload from frequent batch extracts? Note that this may require that 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 supported on the EDW database platform without impacting query performance? Does the EDW platform have sufficient processing capability to perform adequately?

This decision tree shows our recommendations, given these factors.

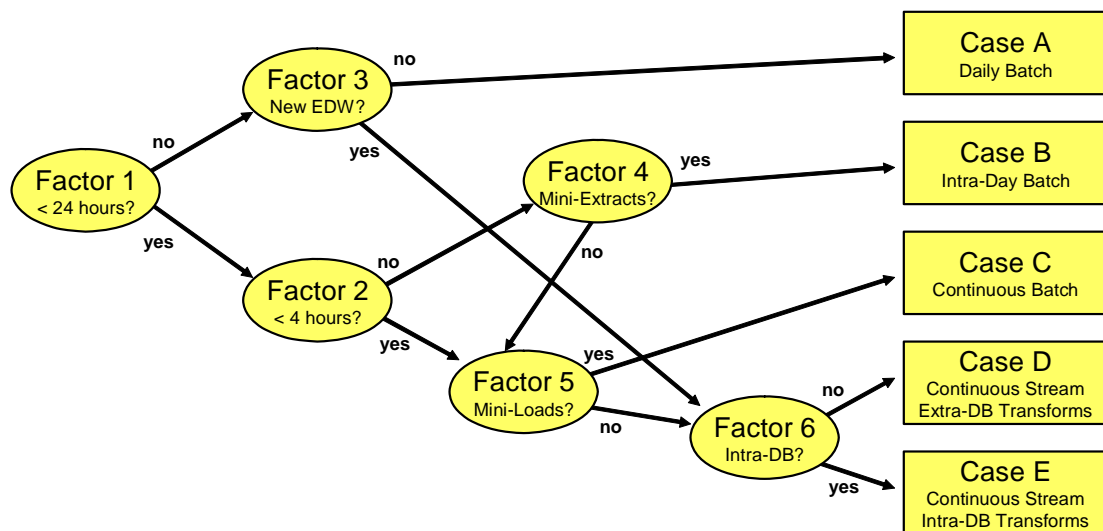


Figure 5: Recommended Architectures for Data Acquisition

Near real-time latency is not always more expensive than typical ETL architectures, depending on these factors. In particular, the decision tree suggests the following recommendations to achieve lowest cost.

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 providing 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.⁶ 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.

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

Driven by 'smart & fresh' data...innovative business processes are emerging.

Driven by 'smart and fresh' data, it is exciting to see the emergence of innovative business processes enabled by a new generation of enterprise applications. As your company is coping with the rigors of the global economy, you are changing the character of Business Intelligence in enterprises.



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 This decay assumed that, the longer you wait to take an action, the less potential business value you will achieve. Further, the value declines more rapidly closer to the business event. This may be valid in many situations. More work is needed to validate this assumption by documenting real decision situations. See the article *Real-Time to Real-Value* in *DM Review*, January 2004.

<http://www.dmreview.com/master.cfm?NavID=55&EdID=7913>

² In reality, corporations have a mixture of architectures depending on their history, environment, and industry.

³ This is sometimes called Extract-Load-Transform instead of ETL.

⁴ Since each Teradata node has two processors, all platforms have eight processors.

⁵ The XLS file for the Cost Estimation Table is available for download to model the costs of actual systems configurations.

http://www.b-eye-network.com/files/knowning_sooner.xls

⁶ Note that Cases D and E can be mixed with Cases A-C to add near real-time data feeds if only a few feeds require intra-day latency.

- - - -

We appreciate the support from GoldenGate Software to pursue this study, along with granting access to their innovative customers.



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 more than 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 has built five data warehouses and been involved in more than 30 data warehouse projects as a consultant. Jack recently managed the building of Overstock's active data warehouse that supports marketing, finance, and merchandising for more than 300 users, as well as ran IT operations for Overstock. Many applications were built on 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 more than one 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, as well as other marketing and system integration firms over the past 20 years. Jack received his B.S. in Computer Science from Purdue University and has continued his 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

EB-5226 > 0207 Teradata is a registered trademark of NCR Corporation.

