连续数据集成调研报告

北京大学计算机系数据库实验室

林子雨

报告内容

- 数据仓库相关概念区分
- 主动数据仓库需求
- 数据集成方式
- 数据集成技术
- 变化数据捕捉技术

数据仓库相关概念区分

- 实时数据仓库(Real-time Data Warehouse)
- 及时数据仓库(Right-time Data Warehouse)
- 主动数据仓库(Active Data Warehouse)

数据仓库相关概念区分

- **实时数据仓库(Real-time Data Warehouse)**

  Michael Haisten首先提出实时数据仓库的概念：

- 在数据仓库中保持两类数据，静态数据和动态数据

  - 静态数据：满足用户的查询分析要求

  - 动态数据：为了实时性，可以实时更新，并做相应转换，满足用户对“最后一分钟”数据的实时请求

  其他定义……

总结：实时数据仓库是这样一个系统，只要行为发生，数据就变得可用，就能从中获得信息。

数据仓库相关概念区分

- **及时数据仓库(Right-time Data Warehouse)[9]**
  - 数据更新周期介于“实时和每天一次”之间
  - 在特定的商务问题提出时，就能马上给出答案
  - 从及时数据仓库中得到的答案，能够帮助组织做出带来巨大收益的决策
  - 为了回答这些事先设计的特定的商务问题，需要在数据仓库中预先存储该商务问题所需的集成的数据(比如一天一次或15分钟一次)

数据仓库相关概念区分

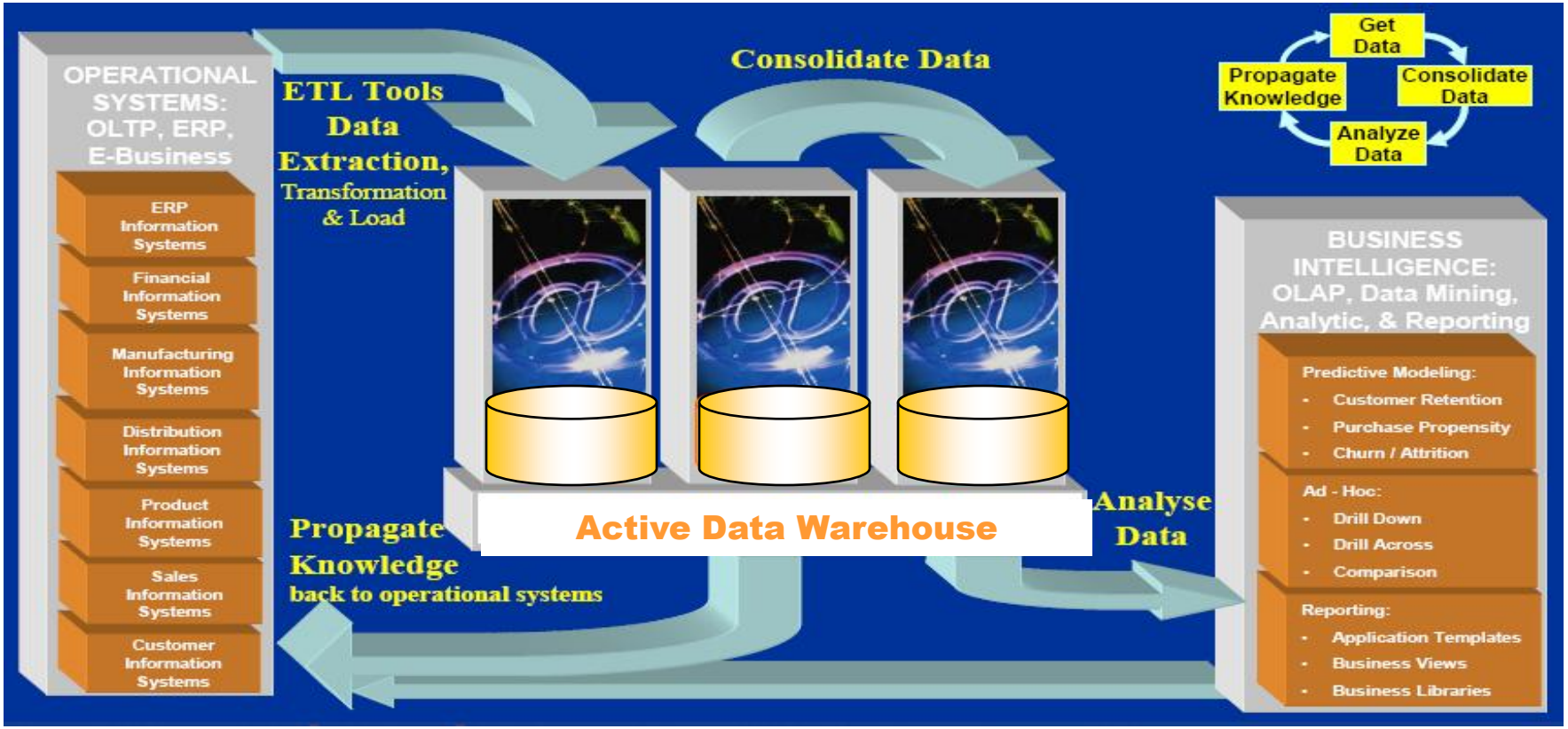
- **主动数据仓库(Active Data Warehouse)[8]**

  主动数据仓库是一个关系型数据仓库环境，支持：

  - 数据的实时更新
  - 快速的响应时间
  - 基于钻取的聚集数据查询能力
  - 动态的交互能力

主动数据仓库概念

概念：是一个集成的信息存储仓库；既具备批量和周期性的数据加载能力，也具备数据变化探测、新数据的连续加载和更新能力；并能结合历史数据和新颖数据实现查询分析和主动行为执行；从而提供对战略决策和战术决策的双重支持 。[自定义]

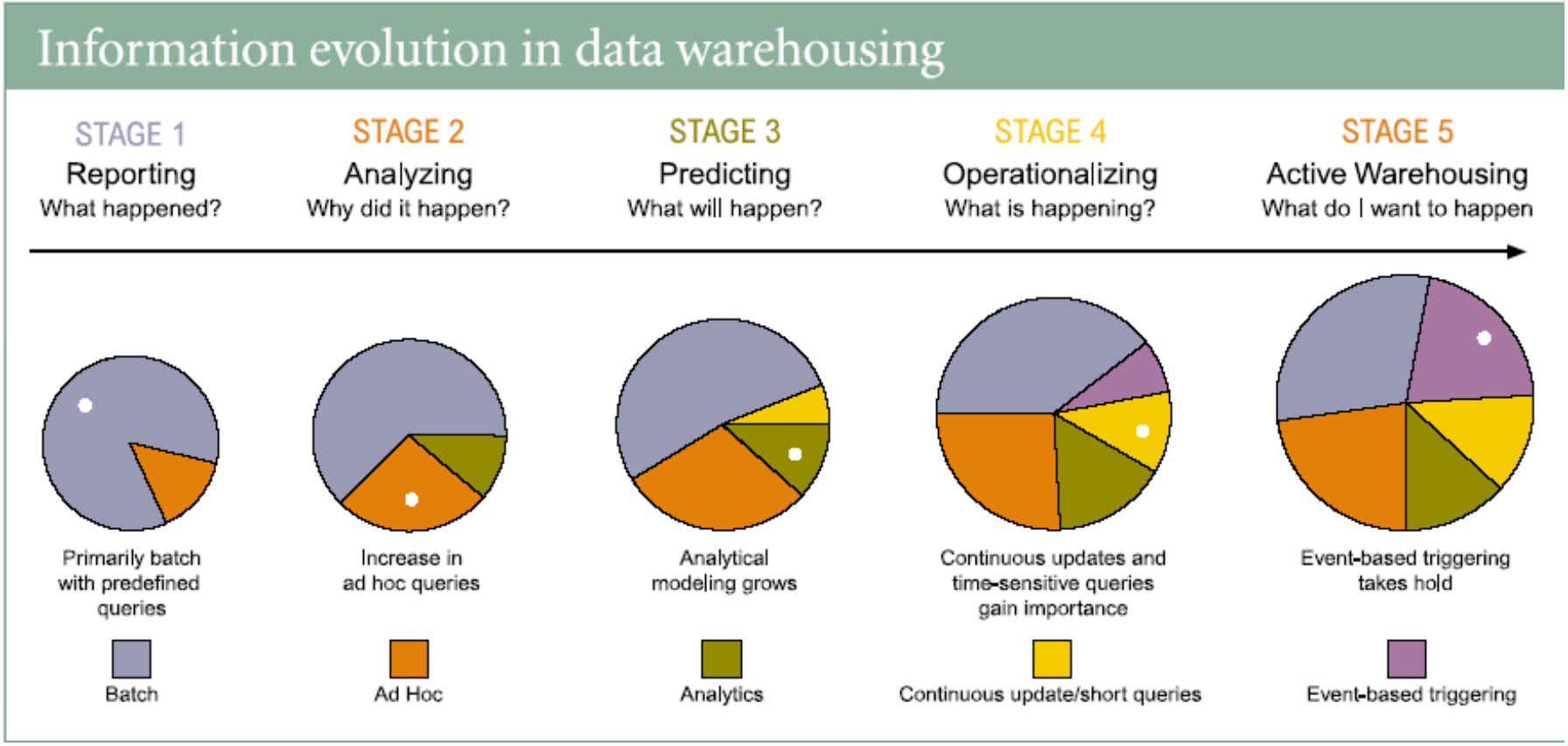


数据仓库相关概念区分

- 主动数据仓库(Active Data Warehouse)
- 及时数据仓库(Right-time Data Warehouse)
- 实时数据仓库(Real-time Data Warehouse)

	主动数据仓库	及时数据仓库	实时数据仓库
更新方式	实时	及时	实时
自动动作执行	有	无	无

数据仓库相关概念区分


Information evolution in data warehousing
STAGE 1
Reporting
What happened?
STAGE 2
Analyzing
Why did it happen?
STAGE 3
Predicting
What will happen?
STAGE 4
Operationalizing
What is happening?
STAGE 5
Active Warehousing
What do I want to happen
Primarily batch with predefined queries
Increase in ad hoc queries
Analytical modeling grows
Continuous updates and time-sensitive queries gain importance
Event-based triggering takes hold
Batch
Ad Hoc
Analytics
Continuous update/short queries
Event-based triggering

主动数据仓库需求

- 数据新颖
- 连续数据集成
- 存储数据的时间一致性
- 主动行为/通知
- 高可用性
- 性能和可扩展性

数据集成方式

- 数据整合（Data Consolidation）[10]
- 数据联邦（Data Federation）
- 数据传播（Data Propagation）
- 混合方式（A Hybrid Approach）

数据仓库中的数据集成

- 传统的数据仓库采用**一次加载并周期更新**的方法，在进行数据更新时，**不允许进行分析操作**。
- 主动数据仓库则必须支持**实时**的查询处理，也就必须要求具备实时数据集成的能力，有好几种方法都是朝着这个方向努力：
  - 为了最小化更新窗口，Labio[2]等人尝试使用批量（bulk）加载工具来实现高性能的数据集成。
  - 04年左右，学术界发表的一些论文中的方法则以最小化整体更新工作量为出发点，来确定更新数据仓库中的单个物化视图的最优策略。
  - 其他方法则重点放在数据/表交换[3]，在这些方法中，ETL工具获得很大的权限，可以删除表、重加载表和操纵其他主要的数据库系统，同时不影响终端用户的查询。

不足：上述方法的一个不尽人意的方面是，数据仓库并不是真正的实时的。在需要真正实时的应用中，最好的方法是象水灌溉一样，数据从源系统源源不断地直接输入到数据仓库中。*怎么实现呢*?

数据集成技术

有许多种技术可以为数据仓库提供数据获取服务，但是，只有部分技术能提供实时(连续)的数据集成。选择技术的标准应该着重参考以下几个方面的因素：数据质量、频率、可接受的延迟、数据集成、转换需求和处理开销。

属性	脚本	ETL	EAI	CDC
数据量*	中等	很高	低	高
频率	间歇性	间歇性	连续性	连续性
延迟	中等到高	中等到高	低	低
数据一致性	无	无	保证	保证
转换	中度	高级	基本	基本
处理开销	间歇性/高	间歇性/高	连续性/中等	连续性/低
*数据量和技术之外的很多其他因素相关，比如数据源、网络带宽和数据库变化捕捉机制				

数据集成技术

脚 本

- 使用灵活且比较经济
- 很容易着手开发和进行修改
- 几乎任何操作系统和绝大部分DBMSs都可以使用脚本
- 耗费开发者的时间和精力
- 不好管理和操作以及不能满足服务水平协议（SLA：Service-Level Agreements）

数据集成技术

ETL

- 实现大规模数据初步加载的理想解决方案
- 提供了高级的转换能力
- 通常都是在“维护时间窗口”进行
- 在ETL任务执行期间，数据源默认不会发生变化

数据集成技术

EAI

- 和ETL解决方案并存，并增强了ETL的功能
- 在源系统和目标系统之间进行连续的数据分发
- 提供高级的工作流支持和基本的数据转换
- 受到数据量的限制

数据集成技术

CDC

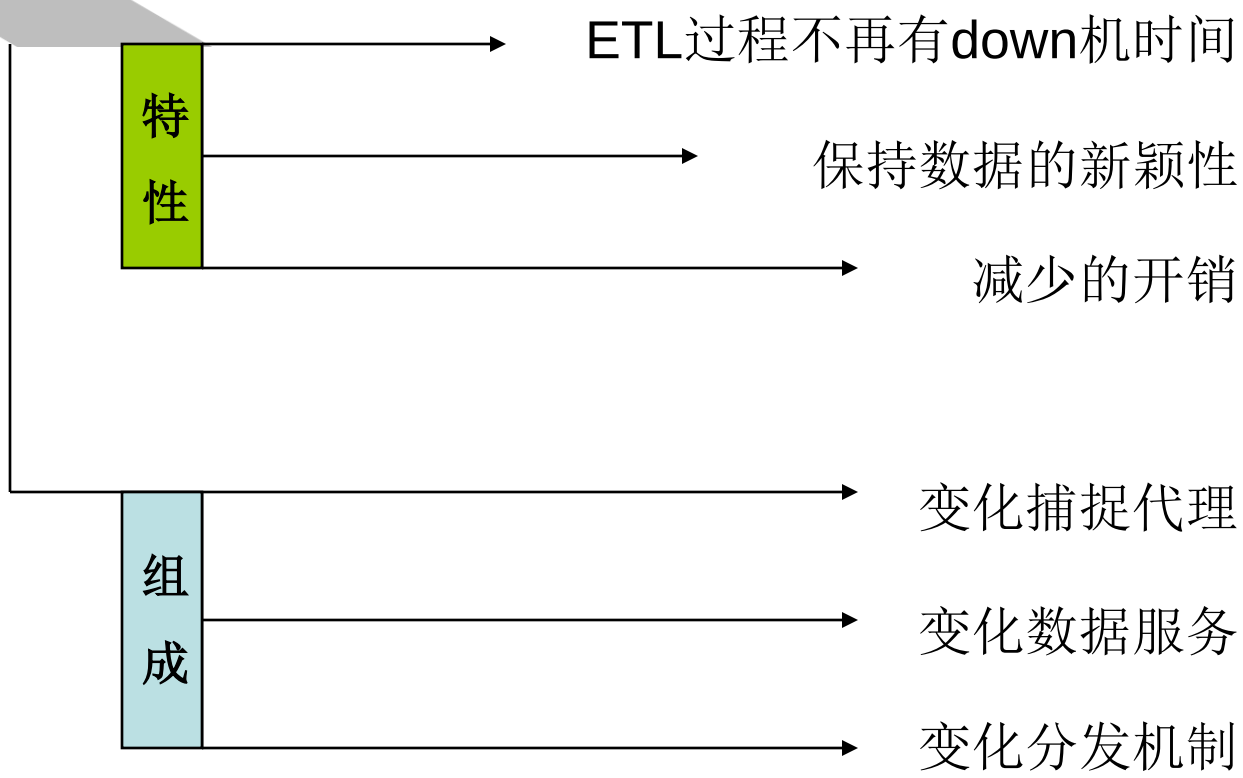
- 提供了连续变化数据的捕捉和分发能力
- 从OLTP系统中捕获变化的数据，进行基本的转换后把数据发送到数据仓库中
- 在体系结构上，CDC属于异步的，但它表现出类似同步的行为
- 维护数据事务的一致性

总　结

- EAI和CDC都只移动变化的数据和更新，而不是整个数据集，从而极大地减少了数据移动量
- **EAI**和**CDC**都不需要假设数据源的状态不发生改变，因为他们自己可以维护数据操作的一致性
- ETL适合作为数据仓库数据初步加载时的解决方案，而EAI和CDC则更适合作为此后的连续数据加载解决方案

变化数据捕捉技术

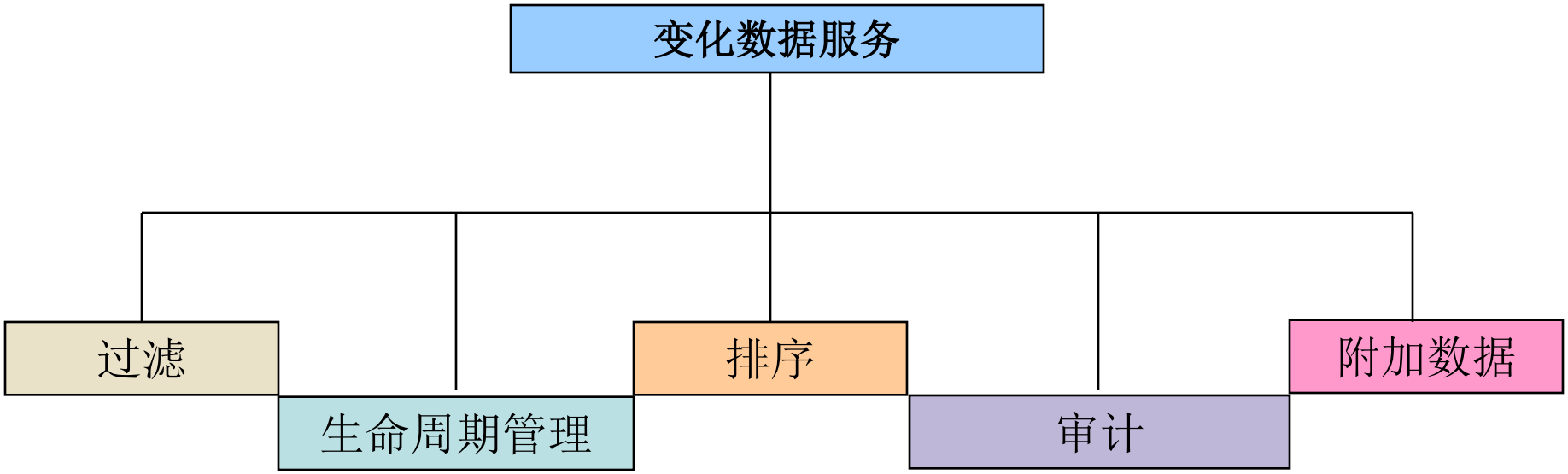
CDC 是一种变化数据捕捉技术，可以实现实时高效的数据集成



变化捕捉代理

- 变化捕捉代理是一个软件组件
- 负责确定和捕捉发生在运营系统中的数据变化
- 可以对变化捕捉代理进行专门优化，使它适用于特定的源系统

变化数据服务



变化分发机制

- 负责把变化分发到消费者（通常是ETL程序）
- 可以支持一个或多个消费者，并且提供了灵活的数据分发方式
- Pull方式需要消费者周期性地发送请求，通常采用标准接口实现
- Push方式需要一直监听和等待变化的发生，一旦捕捉到变化，就立刻转移变化的数据，通常采用消息中间件来实现
- 提供动态返回的能力，从而满足重复处理和恢复处理等任务

数据分发方式

数据分发方式

推vs.拉	周期vs.非周期	一对一vs.一对多	数据分发选择
拉	非周期	一对一	request/response
		一对多	request/response with snooping
	周期	一对一	polling
		一对多	polling with snooping
推	非周期	一对一	Message queue
		一对多	Publish/subscribe
	周期	一对一	email sending
		一对多	email list digest

在传统的数据仓库实施方案中，大都采用pull机制。不管是周期性还是非周期性的pull，都会对运营系统造成额外负担，当我们需要近实时性地数据集成时，这种负担更加严重。对于运营系统来说，push机制是比较理想的，因为系统自己可以控制什么时候把数据推向什么地方。

应用场景1：面向批处理的pull CDC

场景描述

- ETL工具周期性地请求变化，每次都接收批量数据
- 变化分发请求可以采取不同的频度
- 提供变化数据的一种比较好的方式是以数据表的记录的形式表示
- CDC需要维护上次变化分发的位置和分发新的变化

总结：这种应用场景和传统的ETL很相似，不同的是，pull CDC只需要转移变化的数据，并不需要转移所有的数据，这就极大地减少了资源消耗，也消除了传统ETL过程的down机时间。

应用场景2：实时CDC（push CDC）

场景描述

- 满足零延迟的要求
- 变化分发机制一旦探测到变化，就把变化push给ETL程序
- 通常是通过可靠的传输机制来实现

注释：虽然面向消息和面向事件的集成方法在EAI产品中更为常见，但现在，已经有很多ETL工具厂商在他们的解决方案中提供这种功能，以满足高端、实时的商务应用需求。当BI应用需要零延迟和最新的数据时，这种实时的数据集成方法就是必须的。

关于CDC技术需要思考的问题

1 对运营系统的入侵（intrusion）程度

- 所有的CDC解决方案都会对系统造成一定程度的影响
- 最高级别的入侵是源代码入侵
- 程度稍低的入侵是“进程内”或“地址空间”入侵
- 入侵程度最低的解决方案不会影响应用的运营数据源

关于CDC技术需要思考的问题

2 捕捉延迟

- CDC解决方案的一个最主要的考虑因素
- 延迟会受到诸多因素的影响，比如：变化捕捉方法、对变化的处理和变化分发机制的选择
- 变化可以周期性地、高频率地甚至实时地进行分发
- 越是实时的解决方案，对运营系统的入侵程度就越高
- 不同的BI应用对数据延迟的要求也不同， CDC解决方案应能进行灵活配置

关于CDC技术需要思考的问题

3 过滤和排序服务

- CDC解决方案应提供不同的服务实现对分发数据的过滤和排序
- 过滤可以保证只有需要的变化才被分发
- 排序则定义了变化被分发的顺序

关于CDC技术需要思考的问题

4 支持多个消费者

- 捕捉到的变化可能需要被分发到一个以上的消费者那里，比如多个ETL进程、数据同步应用和商务活动监测等等

- CDC解决方案需要支持多个消费者，每个消费者可能具有不同的延迟要求

关于CDC技术需要思考的问题

5 失败和恢复

- CDC解决方案必须保证变化能够被正确地分发，即使系统和网络发生异常

- 在进行恢复的时候，必须保证变化分发数据流从最近一次位置开始，而且必须保证在整个分发周期内满足变化的事务一致性

关于CDC技术需要思考的问题

6 主机和遗产数据源

- 专家估计[4]，主机系统仍然存储了大约70%的公司商业信息，主机仍然处理世界上大量的商业事务
- 主机数据源通常存储大量的数据，这就更需要有高效的方法来转移数据
- 此外，比较流行的主机数据源，比如VSAM，是非关系型的，这就给把数据集成加大了难度
- ETL和DW工具一般都要求关系型数据源，这就需要被非关系型数据源映射成关系型

关于CDC技术需要思考的问题

7 和ETL工具的无缝集成

- CDC解决方案与其他ETL工具之间互操作的难易程度
- 采用标准接口和插件的形式可以降低风险，并加快数据集成进度

参考文献

[1] Tho, M. Nguyen; Tjoa, A. Min. “Zero-Latency Data Warehousing for Heterogeneous Data Sources and Continuous Data Streams”;Proceedings of iiWAS 2003, Fifth International Conference on Information and Web-based Applications Services, Jakarta, Indonesia; Austrian Computer Society (OCG) (2003), 3-902134-72-0; 55 – 64.

[2]LABIO,WJ.;YERNENI,R.;GARCIA-MOLINA,H.;Shrinking the Warehouse Update Window;in:ACM SIGMOD Record,Vol.28(2),PP:383-394,June 1999.

[3]KIMBALL, R. Real-time Partitions, in: Intelligent Enterprise Magazine. Vol.5(10),June 2002.

[4] Itamar Ankorion ,Change Data Capture – Efficient ETL for Real-Time BI Article published in DM Review Magazine,January 2005 Issue.

[5] Brobst, S. “Active Data Warehousing and Enterprise Application Integration,” Proceedings of Data Warehousing 2002:From Data Warehousing to the Corporate Knowledge Center, Physica-Verlag Heidelberg, November 12-13, 2002. pp. 15-23.

[6] Brobst, S. and J. Rarey. *The Five Stages of an Active Data Warehouse Evolution*. Teradata Magazine. Winter, 2001. 4.

[7] Stephen Brobst，Carrie Ballinger，*Active Data Warehousing: Why Teradata Warehouse is the Only Proven Platform*，October 2003

参考文献

[8] OLAP and the Active Data Warehouse. Wingspan Technology, Inc. WhitePaper.2003.
[9] Dan E. Linstedt.Active and Right-Time Data Warehousing Defined. http://www.b-eye-network.com/blogs/mt/mt-tb.cgi/346. January, 2006.
[10] Colin White. Data Integration: Using ETL, EAI, and EII Tools to Create an Integrated Enterprise. TDWI Report,November,2005.