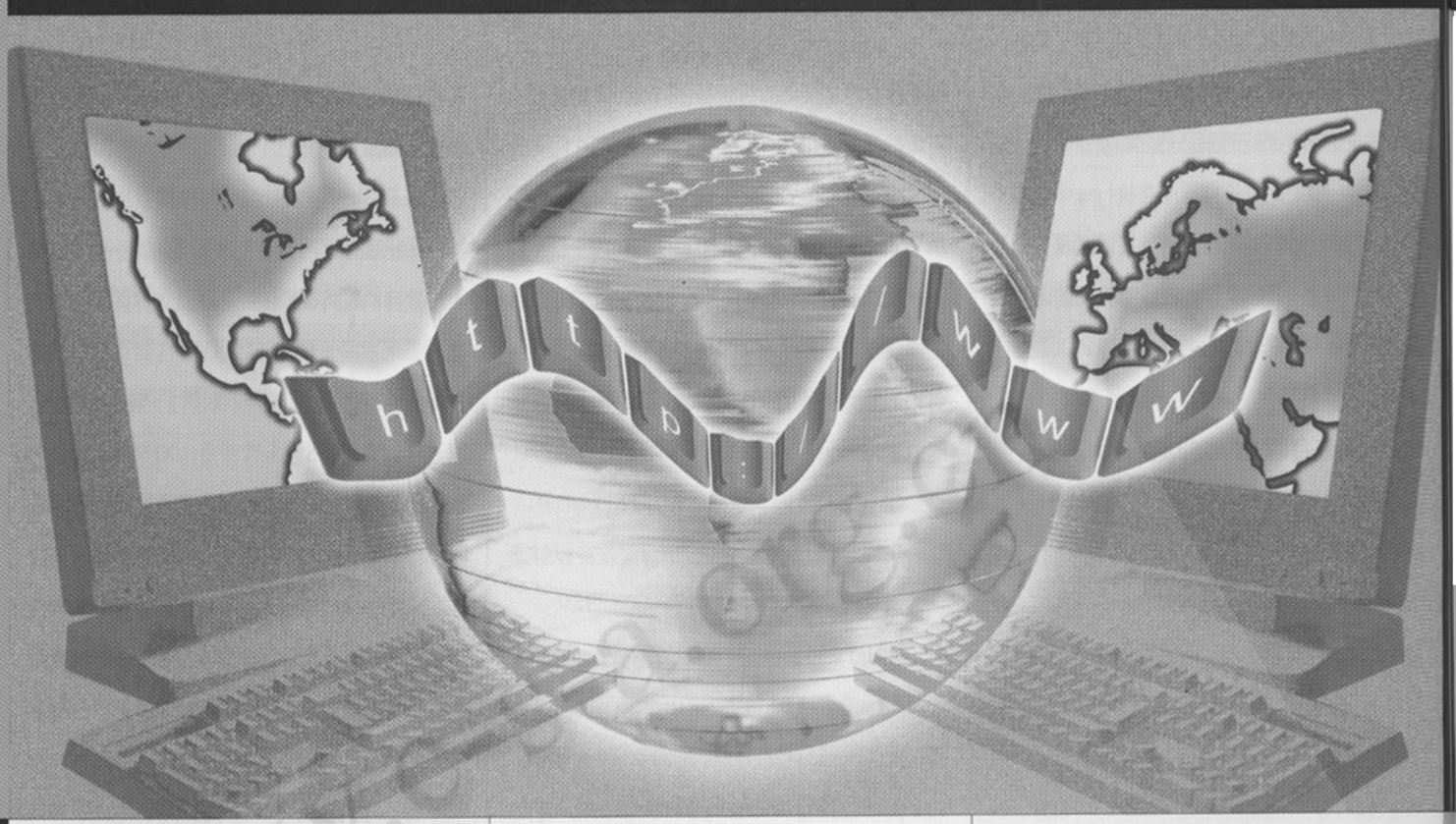
联机分析处理的实现及其性能调优



杨雪峰　李亢

(北京航空航天大学720教研室　100083)

张可彤

(大唐电信企业应用部　100083)

摘要：本文首先对联机分析处理(OLAP)技术进行了简要的介绍，接下来提出了联机分析处理的三种实现方式：关系型联机分析处理(ROLAP)、多维联机分析处理(MOLAP)、混合联机分析处理(HOLAP)，其中对多维联机分析处理的存储方式进行了重点描述，并阐述了基于多维数据组织的MOLAP的性能调优方法，用以提高联机分析处理工具的开发效率，推动计算机应用技术的进一步发展。

关键词：数据仓库　OLAP　MOLAP　ROLAP　HOLAP

1　联机分析处理的实现方法

仿照用户的多角度思考模式，联机分析处理有三种不同的实现方法：

· 关系型联机分析处理(ROLAP，Relational OLAP)

· 多维联机分析处理(MOLAP，Multi-Dimensional OLAP)

· 混合联机分析处理(HOLAP，Hybrid OLAP)

ROLAP表示基于关系型数据库的OLAP实现（Relational OLAP）。以关系数据库为核心，以关系型结构进行多维数据的表示和存储。ROLAP将多维数据库的多维结构划分为两类表：一类是事实表，用来存储数据和维关键字；另一类是维表，即对每个维至少使用一个表来存放维的层次、成员类别等维的描述信息。维表和事实表通过主关键字和外关键字联系在一起，形成了"星型模式"。对于层次复杂的维，为避免冗余数据占用过大的存储空间，可以使用多个表来描述，这种星型模式的扩展称为"雪花模式"。

MOLAP表示基于多维数据组织的OLAP实现（Multidimensional OLAP）。以多维数据组织方式为核心，也就是说，MOLAP使用多维数组存储数据。多维数据在存储中将形成"立方块(Cube)"的结构，在MOLAP中对"立方块"的"旋转"、"切块"、"切片"是产生多维数据报表的主要技术。

HOLAP表示基于混合数据组织的OLAP实现（Hybrid OLAP）。如低层是关系型的，高层是多维矩阵型的。这种方式具有更好的灵活性。

目前ROLAP和MOLAP是常用的OLAP实现方式，在这里主要介绍MOLAP。多维联机分析处理实际上是用多维数组的方式对关系型数据表进行处理。MOLAP的数据存储方式如图1所示：

MOLAP首先对事实表中的所有外键进行排序，并将排序后的具体指标数值一一写进虚拟的

多维立方体中。当然，虚拟的多维立方体只是为了便于理解而构想的，MOLAP实际的数据存储放在数据文件(Data File)中，其数据放置的顺序与虚拟的多维立方体按x，y，z坐标展开的顺序是一致的(如上图(1))。同时，为了数据查找的方便，MOLAP需要预先建立维度的索引，这个索引被放置在MOLAP的概要文件(Outline)中。

概要文件是MOLAP的核心，相当于ROLAP的数据模型设计。概要文件包括所有维的定义(包括复杂的维度结构)以及各个层次的数据汇总关系(例如在时间维，日汇总至月，月汇总至季，季汇总至年)，这些定义往往从关系型维表中直接引入即可。概要文件也包括分析指标的定义，因此可以在概要文件中包含丰富的衍生指标，这些衍生指标由基础指标计算推导出来。

2 MOLAP存储性能调优

如上所述，只要预先定义好概要文件，所有的数据分布就自动确定了，而如何确定概要文件中的稠密维（Dense）、稀疏维（Sparse）及其排列的顺序以求取得良好的性能，是我们下面介绍的性能调优的主要目的。

MOLAP存储性能调试绝大部分是调整Dense/Sparse维，一般调试的目标是使avg block density/block number达到最大。

以保险业决策支持系统中的多维存储为例，进行性能调优的介绍如下：

2.1 首先判断哪些维作为Dense维

如表1所示。

(1) 首先获得所有维的level 0成员个数（除了account维，time维以外，accoun维一般都会是Dense的，这样计算和查询的性能才会好；time维一般会是Sparse的，这样当每次新数据加载时，不会影响到原来的block，才可能作增量计算)。只选择level 0成员的理由是绝大部分数据都会加载到level 0，除了budget等特殊情况，很少会在上层加载数据，可以在数据仓库中通过dimension table很容易计算出level 0成员个数。

(2) 根据数据量的情况，选择一个月（或季度等）的数据，最好比较有代表性。计算每个维的成员在数据中的实际出现次数，如：

```
select count(distinct education) from fact_table;
```

计算出它出现的概率，即select的结果除以第一步的结果。按出现概率的高低，由高到低排列，概率相等则按level 0成员个数排序。

(3) 计算组合概率。如上表所示，即使4个维单独出现的概率都相等，它们不同组合的概率也是不一样的：

```
select count(*) from (select 1 from fact_table
group by education, sex) as temp;
```

(4) 重复上一步骤(3)，寻找几种可能的组合，其概率高，并且产生的block size在几K到1M之间。

(5) 测试不同组合：调整维的排序，Dense在前，Sparse在后。Dense按成员个数由多到少排列，Sparse相反。将Dense维除了level 0一律标记为dynamic calc或者lable only。

(6) 加载数据并计算，记录：

Time to load data;

Data/Index file size after data loaded;

Output of ESSCMD getdbstats;

Calculation message(such as bitmap usage, etc);

Time to calculate;

Data/Index file size after data calculated;

Output of ESSCMD getdbstats;

计算avg block density / block number。

(7) 重复前两个步骤，avg block density / block number值最大者一般存储性能最好。

2.2 数据加载规则

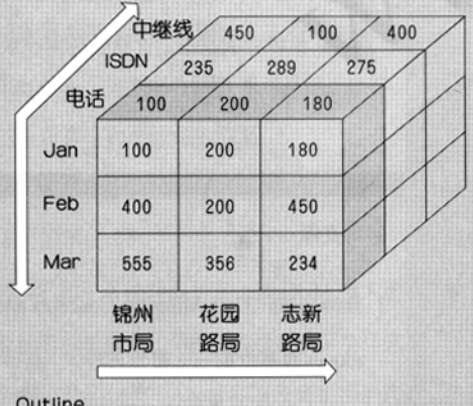
在数据加载（LOAD）的select语句中，应该使Sparse维在前，Dense维在中，Account维最后。应考虑对所有的Sparse维进行order by。

2.3 计算

可以考虑仅用SET MSG ONLY计算，编写计算脚本如下：

```
SET MSG SUMMARY;
SET MSG ONLY;
SET UPDATECALC OFF;
```

图1 MOLAP的数据存储方式



Data File

100	200	180
400	200	450
555	356	234
235	289	275
450	100	400
46	676	456
456	324	245
325	350	520
211	142	543

Outline

地域维	锦州市局	花园路局	志新路局
时间维	Jan	Feb	Mar
申请服务类型	电话	ISDN	中继线

SET CALCCACHE HIGH;

CALC ALL;

可以把日志中输出的sparse calculation作为最后生成的block个数，以刚加载数据时的avgblock density作为最后的avg block density来计算（经过计算后avg block density应该有所增加）。

Essbase.cfg该文件中的设置必须经过essbase重起后才能生效。

CALCCACHE HIGH 100000000

CALCCACHE DEFAULT 30000000

CALCCACHE LOW 20000000

计算需要的CALCCACHE：

Essbase利用Bitmap来帮助计算。Bitmap的大小为S1 * S2 * S3 ...*Sn/8，即所有Sparse维（除了outline中排列在最后的一个）成员个数相乘除以8。

如果没有share member，则最小的calccache至少为前面计算的结果。如果有share member，请参考手册。一般尽量使用多个bitmap来计算。

表1

Dimension	Level 0	Col. Name	Count	Probability
学历	13	education	13	100.0%
年龄	9	Age	9	100.0%
缴费年期	7	Prem_term	7	100.0%
性别	3	Sex	3	100.0%
司龄	21	Hire_age	20	95.2%
婚姻	9	Mari_sts	8	88.9%
缴别	6	Prem_type	5	83.3%
险种	89	Plan_code	45	50.6%
保单来源	8	src_code	4	50.0%
部门	8392	deptno	2273	27.1%
度量维	*	Account	*	*
时间维	*	Time	*	*

Dim组合	Combination	Count	Probability
1,2	117	86	73.5%
1,3	91	88	96.7%
1,4	39	27	69.2%
2,3	63	63	100.0%
2,4	27	18	66.7%
3,4	21	21	100.0%
2,3,4	189	122	64.6%
2,3,1	819	511	62.4%
1,3,4	273	167	61.2%
1,2,3,4	2457	907	36.9%
1,2,3,4,5	51597	8010	15.5%
1,2,3,4,5,6	1083537	14519	1.3%

CALCCACHE的最大设置为200,000,000 byte。

如果Sparse维比较多，导致S1 * S2 * S3 ... *Sn/8就大于200M，那么essbase将去掉Sn，即anchor在Sn上，而不是（理想情况下）最后一个Sparse维，并且重复上述步骤，直到S1 * S2 * S3 ...满足calc cache设置。这就是为什么Sparse维要由小到大排列。

所以，如果将某一个Sparse维变成Dense维能够使essbase anchor在最后一个Sparse维（也是成员最多的）上，并且牺牲的avg block density不多，实际上可以得到计算性能的提高。

2.4 估计计算时间和存储空间

第一次计算所有数据（比如说3年）的时间，可以用前面（如一个月）的时间乘以36。需要的存储空间可以同样的方法来估计。

以上方法是Sparse Split，即利用Sparse（通常是时间）上的一部分数据来测试/估计，这种方法的avg block density，compression ration应该比较准确。

3 结论

上面所叙述的有关MOLAP的实现方式及其性能调优，旨在发展OLAP技术，使其在数据仓库的发展过程中起到一定的推进作用，其经验可以共享，最终目的是推动计算机应用技术的进一步发展。

参考文献

1 JiaweiHan .Data Minning_Concepts and Techniques数据挖掘——概念与技术，高等教育出版社，2001年5月第1版。

2 范明、孟小峰等译，数据挖掘概念与技术，机械工业出版社，2001年8月1日。