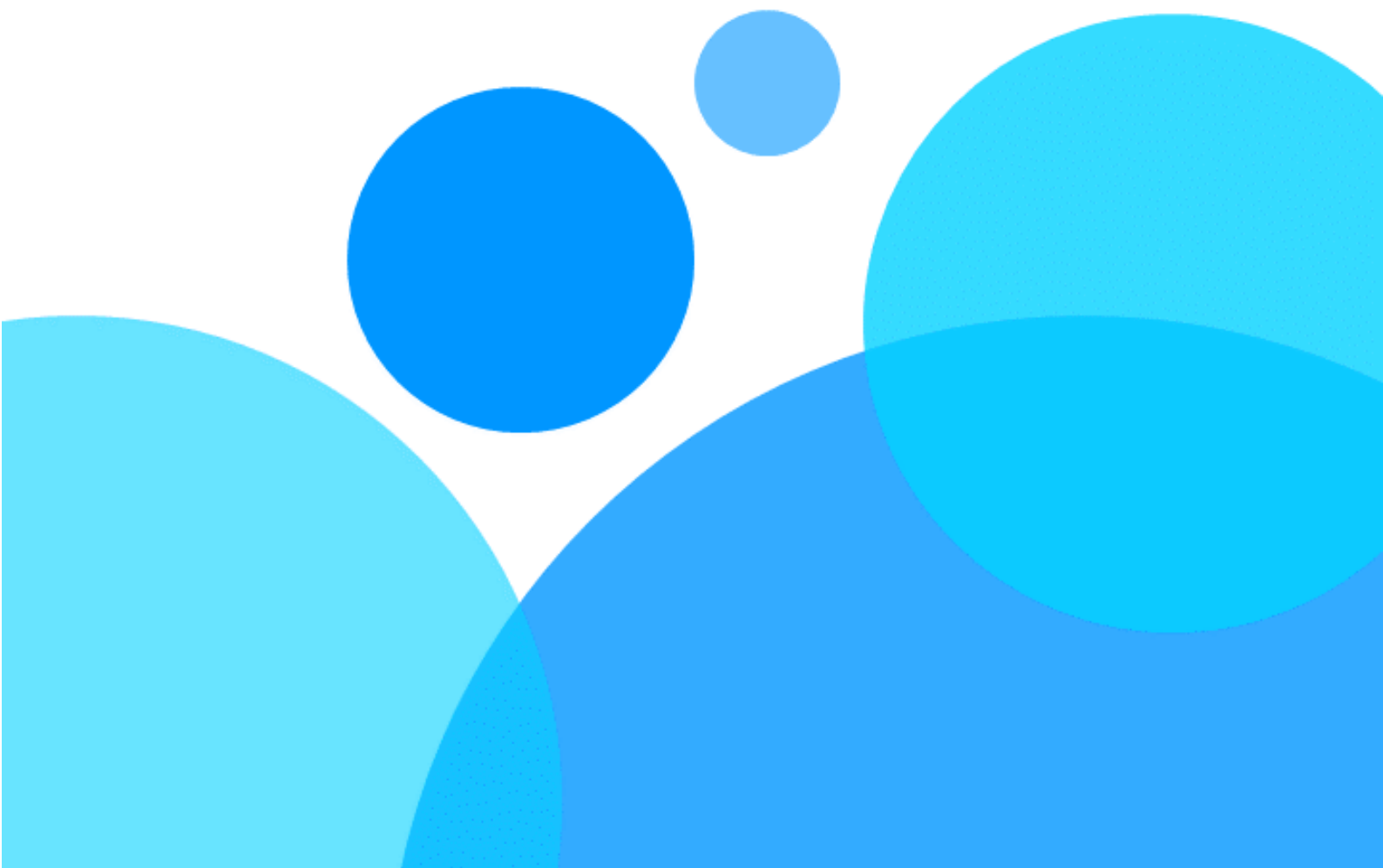




自由释放数据潜能

FineDirect V1.0 性能测试报告

2017年05月27日





目录

1 直连解决方案介绍	3
1.1 直连方案产生背景	3
1.2 直连方案原理	3
1.3 直连方案样例组网	4
2 直连解决方案性能	4
2.1 总体结论	6
2.2 亿级数据分组测试	6
2.3 具体场景对比结果	7
2.3.1 维度汇总	7
2.3.2 排序	9
2.3.3 过滤	10
2.3.4 topN	11
2.3.5 同期环期计算	13
2.3.6 关联功能与 JOIN 对比	13

1 直连解决方案介绍

1.1 直连方案产生背景

BI 分析工具, 基本遵从“设置数据源 → 构建 CUBE → 生成 CUBE → 分析数据”的模式。

面对日益庞大又时刻变化的数据, 构建和生成 CUBE 成为了影响分析时效性的一个重要瓶颈。往往部分数据的些许更新, 数据建模中一小部分的改动, 就需要重新花费大量时间生成 CUBE。这期间需要花费大量的时间, 影响了**数据分析的时效性**。

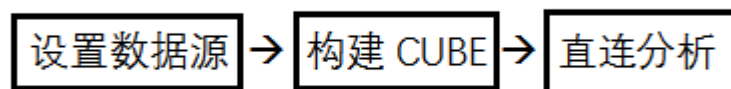
当数据实时变动, 按原有模式, CUBE 生成过程中很难进行数据分析, 导致了很难对实时变动的原始数据, 进行**实时的数据分析**。

不仅如此, 目前各种大数据使用的数据库, 性能已经有很大的提升, 而原有的模式无法完全的利用起各种**高性能数据库的强大性能**。

针对以上痛点, 直连数据库进行 BI 分析的解决方案应运而生。

1.2 直连方案原理

直连数据库进行 BI 分析的解决方案(下称“直连方案”), 由原先的“设置数据源 → 构建 CUBE → 生成 CUBE → 分析数据”, 取消了生成 CUBE 的步骤, 不再将数据库中数据转存为自有的数据格式。分析过程中的数据都直接来自数据库查询的结果。

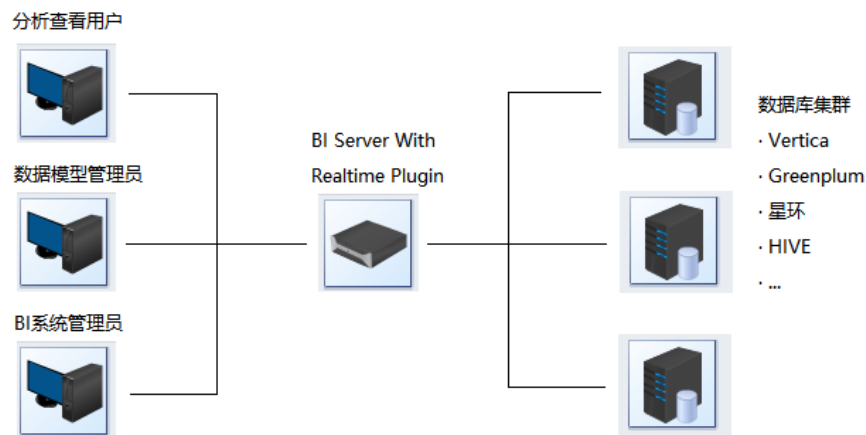


BI 分析用户在已学会 BI 的 CUBE 构建的基础上, 不需要任何额外的学习成本, 直接配置数据连接, 新增直连的数据集到 BI 中, 就可以使用“直连方案”, 添

加的直连数据集字段, 进行数据分析了. 数据查询的 SQL 语句由系统自动生成, 大大简化了用户的操作.

1.3 直连方案样例组网

“直连方案”组网如图所示, 通过附带直连功能的 BI 服务器, 搭建起数据分析用户与数据库之前的桥梁. 通过 BI 服务的数据集配置, 表关联配置, 数据过滤等功能, 自动生成 SQL 向数据库进行查询.



2 直连解决方案性能

“直连方案”在取消生成 CUBE 缩短分析耗时, 提高分析时效性的同时, 性能也毫不落后. 在高性能数据库的支持下, 甚至性能优于当前的本地生成 CUBE 的解决方案. (注: 需要高性能数据库支持以达到分析查询数据的高效).

以下通过对比测试, 验证“直连方案”的性能:

1/ 验证环境:

客户机	BI 服务器	数据库
IP:	IP:	IP:



<p>10.113.0.217</p> <p>浏览器:</p> <p>Chrome 56(64-bit)</p>	<p>10.113.0.75</p> <p>BI 工程:</p> <p>FineBI v4.0 + 直连功能</p> <p>服务器配置:</p> <p>INTEL E5 CPU *1 / 16G 内存</p>	<p>10.113.0.10(主节点)</p> <p>10.113.0.11</p> <p>10.113.0.12</p> <p>数据库:</p> <p>某 V 数据库</p> <p>集群配置:</p> <p>三机集群, 6 个节点</p> <p>单台配置:</p> <p>INTEL E5 CPU *2 /64G 内存 /1TB SSD</p>
--	--	---

2/ 使用数据表:

使用两张表 test_1w, test_1ww, 数据量分别为 1 万条, 1 亿条记录. 两张表通过字段值, 构建关联关系.

3/ 验证项目

对数据进行常规的 BI 分析, 如维度汇总, 排序, 控件过滤, topN 过滤, 同比环比计算等, 比较后台计算时间.

并且使用 loadrunner 进行并发访问场景下的前端加载时间记录

此外特别验证“直连方案”的关联功能与数据库本身 JOIN 语法的性能对比.

2.1 总体结论

- “直连方案”在亿级数据量下可以在秒级展现;
- “直连方案”与原生成 CUBE (现 FineIndex) 的方案相比, 有 10%左右效率优势;
- 由于“直连方案”的数据缓存设计, 并发情况下不会对数据库造成额外的负荷导致响应速度变慢;
- “直连方案”配置关联与数据库自带的 JOIN 相比, 维度汇总查询性能基本近似, 明细表查询性能优于 JOIN;
- “直连方案”与友商数据库直接数据库查询功能相比, 比 Y 友商查询效率高 20%左右, 特别过滤性能耗时只需其一半; 比 T 友商查询效率高 2 倍以上, 数据量越大差距越大.

下面给出具体数据

2.2 亿级数据分组测试

1 亿数据/1 个维度/1 个指标, 计算数据耗时如下:

操作		10 分组	100 分组	1000 分组	10000 分组
维度汇总	首次加载	0.359	0.406	1.086	1.273
	第一次	0.019	0.010	0.022	0.026
	第二次	0.017	0.018	0.030	0.041
	第三次	0.015	0.019	0.019	0.074
	平均时间	0.017	0.012	0.023	0.047
控件过滤 10	第一次	/	0.336	0.385	0.472

条	第二次	/	0.346	0.287	0.236
	第三次	/	0.267	0.301	0.411
	平均时间	/	0.317	0.324	0.373
维度过滤 100 条	首次加载	/	/	0.965	1.488
	第一次	/	/	0.009	0.047
	第二次	/	/	0.007	0.024
	第三次	/	/	0.009	0.035
	平均时间	/	/	0.008	0.035
指标排序	第一次	0.007	0.007	0.021	0.054
	第二次	0.010	0.004	0.027	0.017
	第三次	0.008	0.007	0.007	0.036
	平均时间	0.008	0.006	0.018	0.035

2.3 具体场景对比结果

2.3.1 维度汇总

验证场景:

使用 test_1w 和 test_1ww 表, 小表字段作为维度, 大表数值字段作为指标进行关联表维度汇总. 添加文本和数值控件, 绑定 test_1w 表的字段.

如下示例. 记录并且对比各种方案的后台计算时间.

文本类控件: 无限制 | 数值类控件: 无限制 | 值: 无限制

group100	amount1	1ww记录数
group100_0	16386104268	1000000
group100_1	16378350154	1000000
group100_10	16382881305	1000000
group100_11	16399890918	1000000
group100_12	16362471179	1000000
group100_13	16385338839	1000000
group100_14	16368832736	1000000
group100_15	16391059904	1000000
group100_16	16371295637	1000000
group100_17	16383481393	1000000
group100_18	16367478742	1000000
group100_19	16394018160	1000000
group100_2	16382878553	1000000
group100_20	16377150262	1000000
group100_21	16390776759	1000000
group100_22	16368787509	1000000
group100_23	16377078713	1000000
group100_24	16386800937	1000000
group100_25	16376417143	1000000
group100_26	16387149760	1000000
汇总	1638389021159	10000000

验证结果:

操作		直连方案	生成 CUBE 文件方案	Y 友商	T 友商
单用户访问后台计算时间	首次加载	3.125	4	4.315	30 以上
	第一次	0.085	0.943	0.125	15
	第二次	0.020	0.925	0.097	6
	第三次	0.016	1.000	0.098	6
	平均时间	0.041	0.956	0.106	6.33

并发打开分析 耗时	1 用户	2.08	2.89	2.97	/
	10 用户	2.11	3.01	3.25	/
	50 用户	3.12	3.78	4.17	/
	100 用户	4.14	5.45	5.67	/

(单位: 秒)

2.3.2 排序

验证场景:

使用 test_1ww 数据创建维度汇总, 对指标数值进行排序.

使用 test_1ww 数据创建明细数据表, 对数值字段进行排序.

如图所示. 记录并且对比各种方案的后台计算时间.

统计组件			明细表				
group100	if	amount1	id	amoun..	amoun..	group...	group100000
group100_11			24861366	0	10533	group10_6	group1000000_861
group100_19			65494589	0	14228	group10_9	group1000000_494
group100_30			74168724	0	16385	group10_4	group1000000_166
group100_31			74472807	0	24272	group10_7	group1000000_472
group100_38			70304140	0	19546	group10_0	group1000000_304
group100_4			35812095	0	23777	group10_5	group1000000_812
group100_47			12410481	0	15134	group10_1	group1000000_410
group100_56			39770368	0	17812	group10_8	group1000000_770
group100_6			50784337	0	17247	group10_7	group1000000_784
group100_64			88410803	0	7839	group10_3	group1000000_410
group100_66			20560725	0	10825	group10_5	group1000000_560
group100_67			61956102	0	19772	group10_2	group1000000_956
group100_7			39626187	0	13321	group10_7	group1000000_626
group100_70			44325831	0	22973	group10_1	group1000000_325
group100_73			56007126	0	19023	group10_6	group1000000_712
group100_79			51858323	0	18010	group10_3	group1000000_856
group100_80			92488690	0	16256	group10_0	group1000000_486
group100_83			78680099	0	8166	group10_9	group1000000_680
group100_85			44366519	0	21381	group10_9	group1000000_366
group100_88			54604362	0	19914	group10_2	group1000000_604
汇总			7746642	0	13913	group10_2	group1000000_746
			19252356	0	8005	group10_6	group1000000_252

验证结果:

操作		直连方案	生成 CUBE 文件方案	Y 友商
维度汇总排序	第一次	0.005	0.957	0.008
	第二次	0.003	0.843	0.009
	第三次	0.005	0.889	0.008
	平均时间	0.004	0.896	0.008
1 亿数据明细表 排序	第一次	4.134	2.331	4.905
	第二次	5.123	3.412	7.032
	第三次	3.412	2.712	6.310
	平均时间	4.223	2.818	6.083

(单位: 秒)

2.3.3 过滤

验证场景:

2.3.1 场景, 修改控件值进行文本控件, 数值范围控件过滤, 记录并且对比各种方案的后台计算时间.

验证结果:

操作	直连方案	生成 CUBE 文件方	Y 友商	T 友商

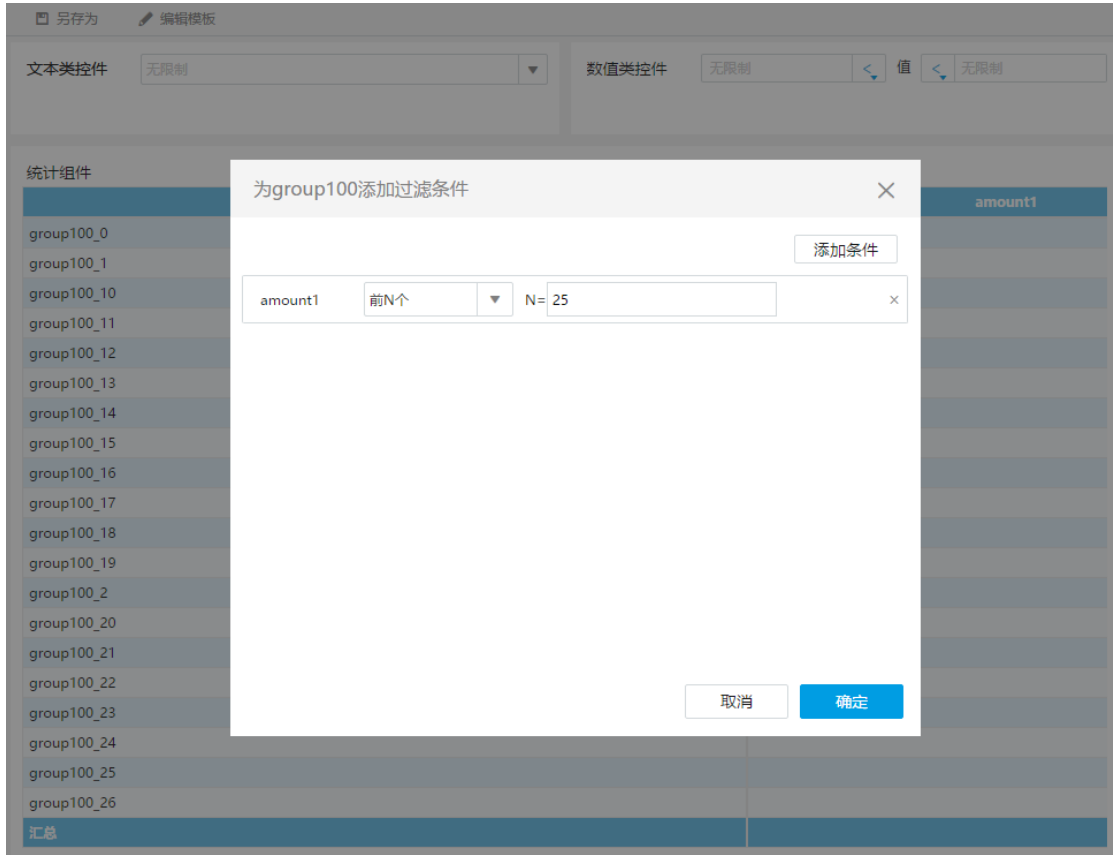
			案		
文本控件 过滤 10 个值	第一次	0.096	0.120	0.187	16.39
	第二次	0.068	0.114	0.218	10.23
	第三次	0.058	0.131	0.220	12.98
	平均时间	0.074	0.121	0.208	13.20
数值控件范围 过滤	第一次	2.081	1.612	2.97	26.32
	第二次	1.786	1.704	3.25	21.34
	第三次	1.983	2.189	4.17	27.13
	平均时间	1.950	1.830	5.67	24.93

(单位: 秒)

2.3.4 topN

验证场景:

2.3.1 场景, 预览时对指标进行自定义过滤, 选择 topN 过滤. 记录并且对比各种方案的后台计算时间.



验证结果:

操作		直连方案	生成 CUBE 文件方案	Y 友商	T 友商
维度 TOPN 过滤	第一次	0.032	0.023	0.034	1.23
	第二次	0.028	0.038	0.037	1.34
	第三次	0.018	0.021	0.029	0.98
	平均时间	0.026	0.027	0.033	1.18

(单位: 秒)

2.3.5 同期环期计算

验证场景:

2.3.1 场景, 添加同期环期计算指标进行分析预览, 记录后台计算时间..

验证结果:

操作		直连方案	生成 CUBE 文件 方案	Y 友商	T 友商
维度 TOPN 过滤	首次打开	2.312	1.890	3.612	15
	第一次	0.045	0.043	0.094	5
	第二次	0.021	0.038	0.137	5
	第三次	0.037	0.031	0.089	5
	平均时间	0.034	0.037	0.106	5

(单位: 秒)

2.3.6 关联功能与 JOIN 对比

验证场景:

使用上文中两张数据表创建分析, A 分析使用“直连方案”提供的方法构建表关联, B 分析使用 SQL 的 JOIN 语句进行查询, 使用“直连方案”进行展示.

对比后台计算时间

验证结果:

操作		使用关联功能	使用 JOIN 关联
维度汇总	首次加载	1.523	1.423
	第一次	0.031	0.021
	第二次	0.020	0.025
	第三次	0.016	0.023
	平均时间	0.022	0.023
明细表	首次加载	0.723	1.414
	第一次	0.017	0.006
	第二次	0.019	0.013
	第三次	0.019	0.012
	平均时间	0.018	0.011
维度文本控件过滤	第一次	0.040	0.006
	第二次	0.037	0.012
	第三次	0.034	0.007
	平均时间	0.037	0.008
明细表排序	第一次	5.570	7.252



	第二次	5.476	7.400
	第三次	5.579	7.345
	平均时间	5.542	7.332

(单位: 秒)

