

SAP系统大数据量资产负债表的性能优化案例

数据计算中间件-集算器应用场景



真实案例，加作者了解更多

案例背景



我们先来看一张资产负债表的样子!

查询条件

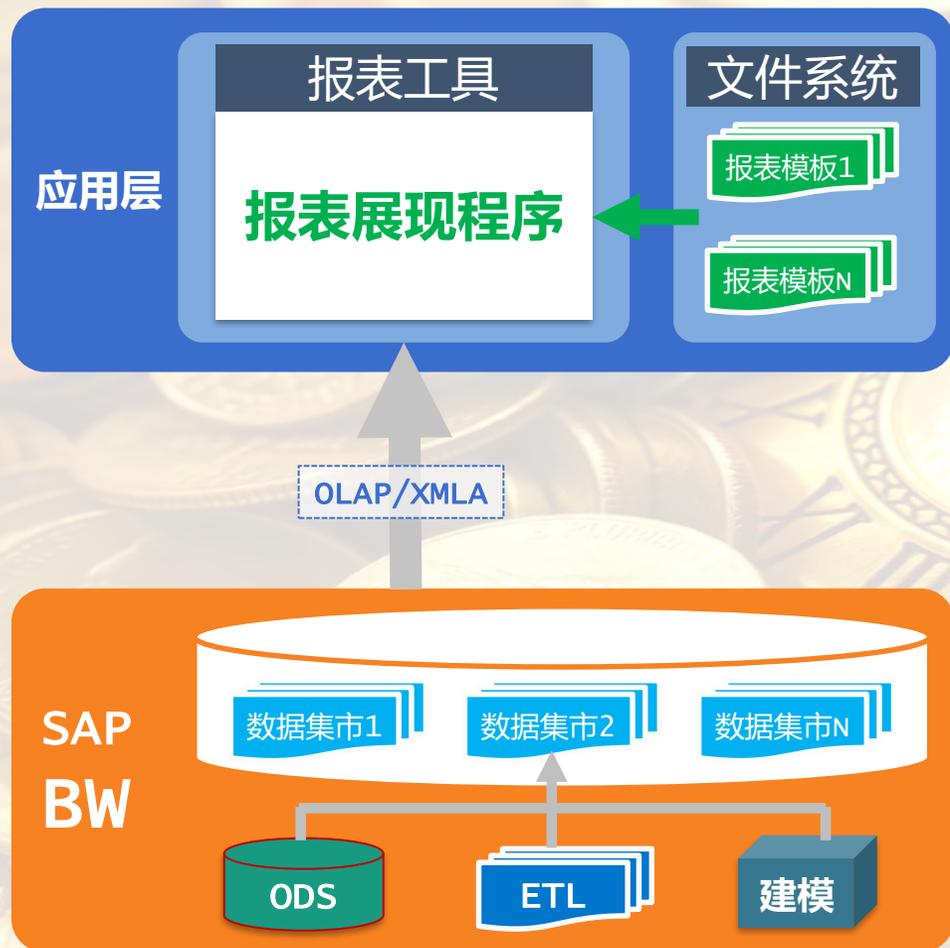
会计年度 (必输)
 公司代码 (非必输)
 利润中心 (非必输)
 过账期间 (必输)

项目	行次	期末余额	年初余额	项目	行次	期末余额	年初余额
流动资产:	1	-----	-----	流动负债:	73	-----	-----
货币资金	2	=Sum([查询 1].「期末	=Sum([查询 1].「年初余额」) where (短期借款	74	=-Sum([查询 H101].「期末余额」)	=-Sum([查询 1].「年初余额」)
△结算备付金	3	-----	-----	△向中央银行借款	75	-----	-----
△拆出资金	4	-----	-----	△吸收存款及同业存放	76	-----	-----
以公允价值计量且其变动计	5	=Sum([查询 1].「期末	=Sum([查询 1].「年初余额」) where (△拆入资金	77	-----	-----
衍生金融资产	6	-----	-----	以公允价值计量且其变动计入当期	78	-----	-----
应收票据	7	=Sum([查询 1].「期末	=Sum([查询 1].「年初余额」) where (衍生金融资产	79	=-Sum([查询 1].「期末余额」) where	=-Sum([查询 1].「年初余额」)
应收账款	8	=Sum([查询 1].「期末	=Sum([查询 1].「年初余额」) where (应付票据	80	=-Sum([查询 1].「期末余额」) where	=-Sum([查询 1].「年初余额」)
合同资产	9	=Sum([查询 1].「期末	=Sum([查询 1].「年初余额」) where (应付账款	81	=-Sum([查询 1].「期末余额」) where	=-Sum([查询 1].「年初余额」)
预付款项	10	=Sum([查询 1].「期末	=Sum([查询 1].「年初余额」) where (预收账款	82	=-Sum([查询 1].「期末余额」) where	=-Sum([查询 1].「年初余额」)
△应收保费	11	-----	-----	△卖出回购金融资产款	83	-----	-----
△应收分保账款	12	-----	-----	△应付手续费及佣金	84	-----	-----
△应收分保合同准备金	13	-----	-----	应付职工薪酬	85	=-Sum([查询 1].「期末余额」) where	=-Sum([查询 1].「年初余额」)
应收利息	14	=Sum([查询 1].「期末	=Sum([查询 1].「年初余额」) where (其中: 应付工资	86	=-Sum([查询 1].「期末余额」) where	=-Sum([查询 1].「年初余额」)
应收股利	15	=Sum([查询 1].「期末	=Sum([查询 1].「年初余额」) where (应付福利费	87	=-Sum([查询 1].「期末余额」) where	=-Sum([查询 1].「年初余额」)
其他应收款	16	=Sum([查询 1].「期末	=Sum([查询 1].「年初余额」) where (#其中: 职工奖励及	88	-----	-----
△买入返售金融资产	17	-----	-----	应交税费	89	=-Sum([查询 1].「期末余额」) where	=-Sum([查询 1].「年初余额」)
存货	18	=Sum([查询 1].「期末	=Sum([查询 1].「年初余额」) where (其中: 应交税金	90	-----	-----
其中: 原材料	19	=Sum([查询 1].「期末	=Sum([查询 1].「年初余额」) where (应付利息	91	=-Sum([查询 1].「期末余额」) where	=-Sum([查询 1].「年初余额」)
库存商品(产成品)	20	=Sum([查询 1].「期末	=Sum([查询 1].「年初余额」) where (应付股利	92	=-Sum([查询 1].「期末余额」) where	=-Sum([查询 1].「年初余额」)
划分为持有待售的资产	21	-----	-----	其他应付款	93	=-Sum([查询 1].「期末余额」) where	=-Sum([查询 1].「年初余额」)
一年内到期的非流动资产	22	=Sum([查询 1].「期末	=Sum([查询 1].「年初余额」) where (△应付分保账款	94	-----	-----
其他流动资产	23	=Sum([查询 1].「期末	=Sum([查询 1].「年初余额」) where (△保险合同准备金	95	-----	-----
流动资产合计	24	=「流动资产合计-期末」	=「流动资产合计-年初」	△代理买卖证券款	96	-----	-----
非流动资产:	25	-----	-----	△代理承销证券款	97	-----	-----
△发放贷款及垫款	26	-----	-----	划分为持有待售的负债	98	-----	-----
可供出售金融资产	27	=Sum([查询 1].「期末	=Sum([查询 1].「年初余额」) where (一年内到期的非流动负债	99	=-Sum([查询 1].「期末余额」) where	=-Sum([查询 1].「年初余额」)
持有至到期投资	28	=Sum([查询 1].「期末	=Sum([查询 1].「年初余额」) where (其他流动负债	100	=-Sum([查询 1].「期末余额」) where	=-Sum([查询 1].「年初余额」)
长期应收款	29	=Sum([查询 1].「期末	=Sum([查询 1].「年初余额」) where (流动负债合计	101	=「流动负债合计-期末」	=「流动负债合计-期初」
长期股权投资	30	=Sum([查询 1].「期末	=Sum([查询 1].「年初余额」) where (非流动负债:	102	-----	-----
投资性房地产	31	=Sum([查询 1].「期末	=Sum([查询 1].「年初余额」) where (长期借款	103	=-Sum([查询 1].「期末余额」) where	=-Sum([查询 1].「年初余额」)
固定资产原值	32	=Sum([查询 1].「期末	=Sum([查询 1].「年初余额」) where (应付债券	104	=-Sum([查询 1].「期末余额」) where	=-Sum([查询 1].「年初余额」)
减: 累计折旧	33	=-Sum([查询 1].「期	=-Sum([查询 1].「年初余额」) where (长期应付款	105	=-Sum([查询 1].「期末余额」) where	=-Sum([查询 1].「年初余额」)
固定资产净值	34	=Sum([查询 1].「期末	=Sum([查询 1].「年初余额」) where (长期应付职工薪酬	106	-----	-----
减: 固定资产减值准备	35	-----	-----	专项应付款	107	=-Sum([查询 1].「期末余额」) where	=-Sum([查询 1].「年初余额」)
固定资产净额	36	=Sum([查询 1].「期末	=Sum([查询 1].「年初余额」) where (预计负债	108	=-Sum([查询 1].「期末余额」) where	=-Sum([查询 1].「年初余额」)
在建工程	37	=Sum([查询 1].「期末	=Sum([查询 1].「年初余额」) where (递延收益	109	=-Sum([查询 1].「期末余额」) where	=-Sum([查询 1].「年初余额」)
工程物资	38	=Sum([查询 1].「期末	=Sum([查询 1].「年初余额」) where (递延所得税负债	110	=-Sum([查询 1].「期末余额」) where	=-Sum([查询 1].「年初余额」)

现状分析



温馨提示：客户采用的底层数据仓库是**SAP BW**！



恶果！

响应慢

单次报表查询，40+秒才能出结果！

原因1

大数据

原数据表两年累积6000+万明细记录！

原因2

计算多

报表每个单元格都是独立计算的指标，往往这样的计算指标多达上百个！

用户期望

查询响应时间在**5-8秒!**

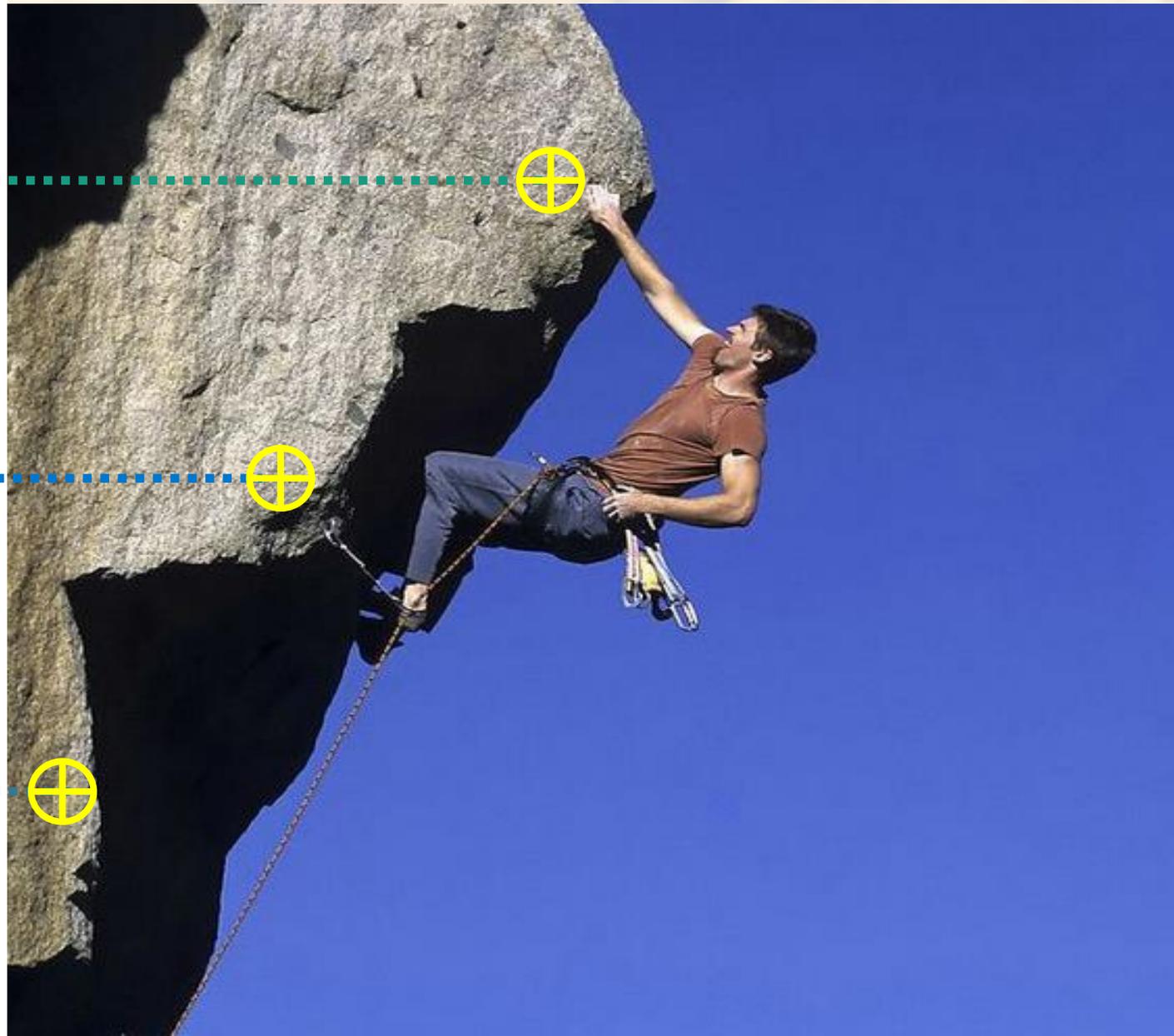
按组合条件过滤，能做到吗?

数据量超过**6千万条**

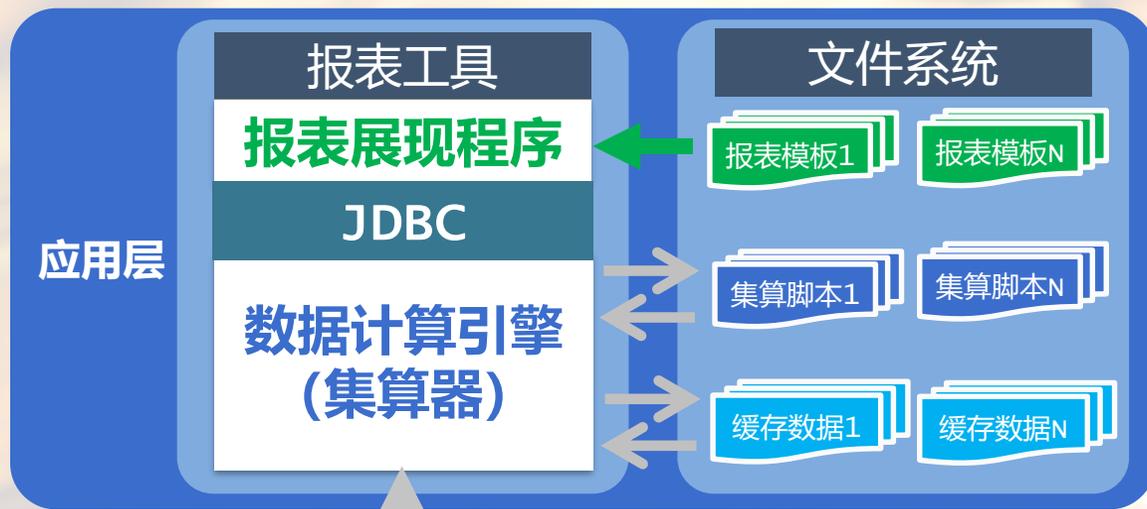
“期末余额”指标须全量遍历

与报表工具全兼容

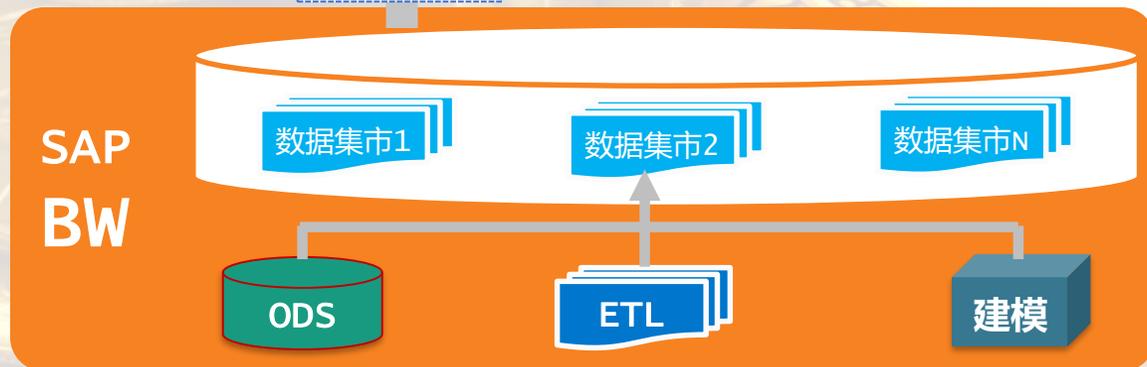
表样不变仅切换数据源



引入数据计算层 - 集算器



OLAP/XMLA



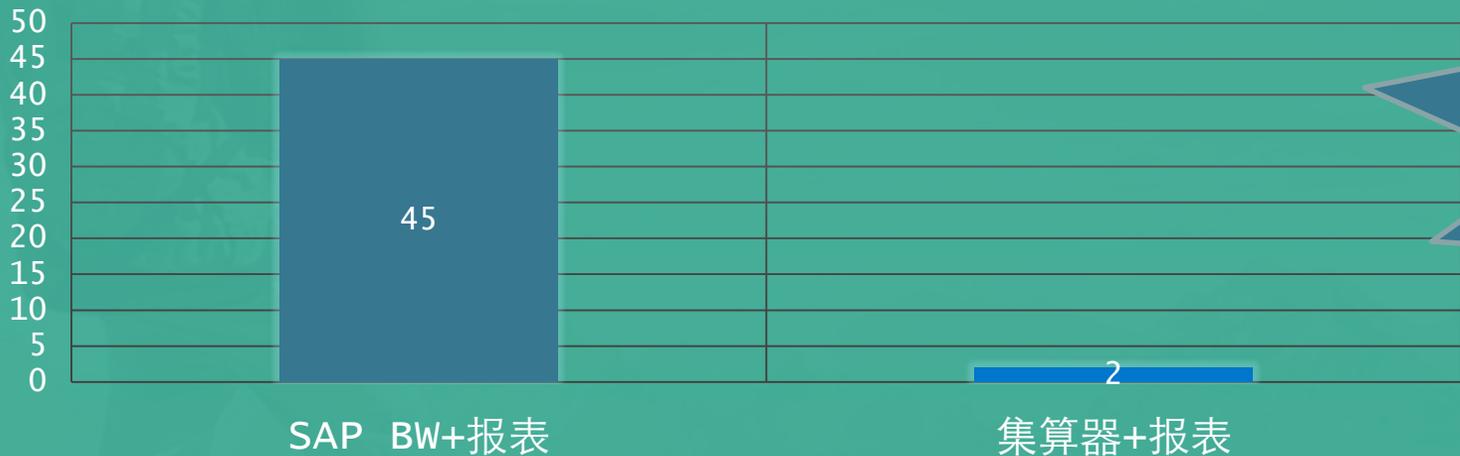
在不改变原有架构体系基础上，引入数据计算层！



各模块之间低耦合，各司其职，便于管理和维护！

实测：集算器表现优秀

查询响应时间（秒）



提升查询效率
22.5倍!

报表从取数到展现整个环节需要**2秒**，实际指标计算部分用时不到**1秒!**

52格

计算代码量减至30%

	A	B	C	D
1	/参数变量	=now()		=(yyyy-inityear)*12+mm
2	[1001,1002,1012]	[2001]	[1101]	[1121,12310106,12310206,12310306,12
3	[1122,12310101,12310401,12319001]	[2201]	[1126]	[2202]
4	[2203]	[2211,2702010200,2702010100]	[1132,12310104,	[221101]
5	[221102]	[1221,12310102,12310202,12310302,123	[2221]	[1321,1401,1402,1403,1404,1405,1406,
6	[2231]	[1405,147103,1471050400]	[2232]	[2241,2235]
7	[1303,1304]	[1503,1504]	[250101,250103,	[15010102,15010202,15010302,150199
8	[153102,12310105,12310205,123103	[1511,1512]	[1521,1607,1608]	[250102,250104,250106]
9	[25020202,25020302,25029902]	[1602]	[27010102,27019	[1601,1602]
10	[2801]	[1604,1609]	[2401030200,240	[1605,1610]
11	[1606]	[2802,2702990100]		
12	=[A2:E11]	=A12.(~.(~=#C1*long(10000000000)+~))		=A12.union().sort()
13	=file("C:\Users\ThinkPad\Desktop\W...")			/总账凭证-later.btx")
14	=A13.iselect@b(C12,key)	=A14.fetch()		/根据有序文件创建游标返回
15	=A12.(~.(B14.(#1).pos@b(~)))			/按照参数条件找到结果集中的坐标位置
16	=A15.(~.sum(B14(~).汇总金额))			/获取指标结果在汇总
17	=A16.new(#.name,~:amount)			/构造帆软需要的key, value
18	return A17			/返回期末余额的指标

SPL

集算器为什么快



数据外置

可获得比数据仓库更高的IO性能和计算能力



压缩存储

数据有序存放,多字段合并唯一主键构造键值对



有序利用

通过键即可高速定位对应的值,避免低效遍历

优势总结



集算器

简称：DCM

(Data Computing
Middleware)



高性能

有序/遍历机制,千万记录秒级响应



易管理

物理文件有序连续存放、易于管理



低成本

压缩数据可简单存储在廉价硬盘中



低耦合

表样不变,数据源切换集算器即可

创新技术 推动应用进步！

