

报告编号: ZT022507187

数据库基准测试报告

Database Benchmark Testing Report

基准类型: HTAP 数据库基准

产品名称: OceanBase

版本号: V4

委托单位: 杭州沃趣科技股份有限公司

北京奥星贝斯科技有限公司

报告时间: 2025 年 7 月 9 日

项目编号: CSTCZT2507124

CSTC 中国软件评测中心
(工业和信息化部软件与集成电路促进中心)

注意事项

1. 本报告未加盖“中国软件评测中心（工业和信息化部软件与集成电路促进中心）检验检测专用章”无效。
2. 本报告未经测试人员、审核人员、批准人员签字无效。
3. 本报告涂改无效。
4. 本报告测试结果只对被测产品有效。
5. 未经中国软件评测中心（工业和信息化部软件与集成电路促进中心）书面批准，不得部分复印报告。
6. 如对本报告的检验检测结果有异议，应在收到报告后的 10 个工作日内与我中心联系。

- 单位地址：北京市昌平区豆各庄 11 号
- 邮政编码：102206
- 电 话：010-60738888/60738999
- 电子邮箱：service@cstc.org.cn
- 投诉电话：010-60738666
- 官方网址：<https://www.cstc.org.cn>

OceanBase V4

测试结论

中国软件评测中心（工业和信息化部软件与集成电路促进中心）受杭州沃趣科技股份有限公司、北京奥星贝斯科技有限公司的委托，于 2025 年 5 月 12 日至 2025 年 7 月 9 日，依据 GB/T 25000.51-2016《系统与软件工程 系统与软件质量要求和评价（SQuaRE）第 51 部分：就绪可用软件产品（RUSP）的质量要求和测试细则》、CSTCQBRYJB001《系统和软件产品测试规范 V4.0》及 HTAP 数据库基准测试指导的要求，在模拟测试环境下，对“OceanBase V4”进行了数据库基准测试。测试结果如下：

1、系统架构。被测产品为数据库管理系统，数据库服务器和客户端服务器操作系统采用银河麒麟高级服务器操作系统 V10（SP3）。测试网络带宽为 1000Mbps。

2、性能效率。对被测产品执行 HyBench 基准测试，并发用户数设置为 $apclient = 10$, $tpclient = 7000$, $xapclient = 1$, $xtpclient = 4500$ ，数据规模为 1500X，测试结果：

执行 OLAP 负载的 QPS 值为 7.93，

执行 OLTP 负载的 TPS 值为 28398.57，

执行 OLXP 负载的 QPS 值为 0.02，

执行 OLXP 负载的 TPS 值为 21512.40，

数据新鲜度（Freshness）值为 0，

Geometric Mean（H-Score）值为 1692.07。

测试结论：

“OceanBase V4”性能效率方面符合 HTAP 数据库基准测试指导的相关要求，在现场测试过程中，系统运行稳定，通过了中国软件评测中心（工业和信息化部软件与集成电路促进中心）的数据库基准测试。

中国软件评测中心（工业和信息化部软件与集成电路促进中心）

2025 年 7 月 9 日



1. 测试结果汇总

说明: “●”为符合项, 符合需求规格说明书的要求; “*”为不符合项; “#”为修改后符合项。

案例编号	测试指标	测试说明	测试结果
1. 性能效率			
OBYT_XL0001	HyBench 基准测试	并发用户数设置: apclient = 10 tpclient = 7000 xapclient = 1 xtpclient = 4500 数据规模: 1500X	
		• 执行 OLAP 负载的 QPS 值	7.93
		• 执行 OLTP 负载的 TPS 值	28398.57
		• 执行 OLXP 负载的 QPS 值	0.02
		• 执行 OLXP 负载的 TPS 值	21512.40
		• 数据新鲜度 (Freshness) 值	0
		• Geometric Mean (H-Score) 值	1692.07
		• 服务器 CPU 资源利用率	数据库服务器 1 90%

2. 性能效率案例描述

2.1. 案例描述

案例名称	HyBench 基准测试	案例编号	OBYT_XL0001
案例目的	本案例测试数据库在 Hybench 基准下的性能表现		
场景设置	测试工具设置: risk_rate=0.1 fresh_interval=200 apRunMins=36 tpRunMins=36 xpRunMins=48 apround=1		
数据量检查	测试执行前涉及表单数据量检查: CUSTOMER: 450000000 条 COMPANY: 3000000 条 SAVINGACCOUNT: 453000000 条 CHECKINGACCOUNT: 453000000 条 LOANAPPS: 900000000 条 TRANSFER: 9000000000 条 CHECKING: 900000000 条 LOANTRANS: 900000000 条		
	测试执行后涉及表单数据量检查: CUSTOMER: 450000000 条 COMPANY: 3000000 条 SAVINGACCOUNT: 453000000 条 CHECKINGACCOUNT: 453000000 条 LOANAPPS: 904610347 条 TRANSFER: 9486251547 条 CHECKING: 923717033 条		

	LOANTRANS: 903512859 条
OLAP 执行 SQL 语句	[AP-1] <pre>sql = "" select /*+ AP_1 parallel(8) */ sourceid,targetid, case when sourceid = ? then 'outbound' when targetid= ? then 'inbound' end as direction, sum(amount) as total_amount from transfer where sourceid= ? or targetid = ? group by sourceid,targetid order by total_amount desc;""</pre> [AP-2] <pre>sql = ""select /*+ AP_2 parallel(8) */ cu.custid,la.loanapps_ts,la.amount,lt.loantrans_ts,lt.amount,lt.status from customer cu,loanapps la, loantrans lt where cu.custID=la.applicantid and la.id=lt.appid and cu.custID = ?;""</pre> [AP-2.1] <pre>sql = "SELECT applicantid FROM loanapps where applicantid<? ORDER BY RAND() LIMIT 1;"</pre> [AP-3] <pre>sql = ""select /*+ AP_3 parallel(8) */ custId, name,phone,sum(amount) from (select cu.custid,cu.name,cu.phone,t.amount from customer cu join transfer t on cu.custid = t.targetid where custid= ? union all select cu.custid, cu.name,cu.phone,ch.amount from customer cu join checking ch on cu.custid = ch.targetid where cu.custid= ?) income group by custid,name,phone""</pre> [AP-4] <pre>sql = ""select /*+ AP_4 parallel(8) */ custid, name,phone,sum(amount) from (select cu.custid,cu.name,cu.phone,t.amount</pre>

	<pre> from customer cu join transfer t on cu.custid = t.sourceid where custid= ? union all select cu.custid,cu.name,cu.phone,ch.amount from customer cu join checking ch on cu.custid = ch.sourceid where cu.custid= ?) outcome group by custid,name,phone"" [AP-5] sql = ""select /*+ AP_5 parallel(8) */ cu.custid ,sum(tr.amount) as total_amount from company co,transfer tr,customer cu where cu.custid = tr.targetid and cu.companyid = co.companyid and tr.sourceid = co.companyid and co.companyid = ? and tr.type='salary' group by cu.custid order by total_amount desc;"" [AP-6] sql = ""select /*+ AP_6 parallel(16) */ cid, total_expense, sum(ch.amount) as checking_amount from (select co.companyid as cid, sum(tr.amount) as total_expense from company co, transfer tr where co.companyid = tr.sourceid and co.name like 'software%' group by co.companyid order by total_expense DESC) big_company, checking ch where big_company.cid=ch.sourceid group by cid, total_expense order by total_expense DESC, checking_amount desc;"" [AP-7] sql = "" (select /*+ AP_7 parallel(1) */ tr.sourceid as sourceid, tr.targetid as targetid, tr.type as type, sum(tr.amount) as amount </pre>
--	---

<pre> from company co, transfer tr, customer cu where cu.custid=tr.sourceid and tr.targetid=co.companyid and co.companyid=? and type!='salary' group by sourceid, targetid, type order by amount DESC limit 3) UNION ALL (select /*+ AP_7 parallel(1) */ ch.sourceid as sourceid, ch.targetid as targetid, ch.type as type, sum(ch.amount) as amount from company co, checking ch, customer cu where cu.custid=ch.sourceid and ch.targetid = co.companyid and co.companyid=? and type!='salary' group by sourceid, targetid, type order by amount DESC limit 3) UNION ALL (select /*+ AP_7 parallel(1) */ tr.sourceid as sourceid, tr.targetid as targetid, tr.type as type, sum(tr.amount) as amount from company co, transfer tr, customer cu where cu.custid=tr.targetid and tr.sourceid=co.companyid and co.companyid=? and type!='salary' group by sourceid, targetid, type order by amount DESC limit 3) UNION ALL (select /*+ AP_7 parallel(1) */ ch.sourceid as sourceid, ch.targetid as targetid, ch.type as type, sum(ch.amount) as amount from company co, checking ch, customer cu where cu.custid=ch.targetid </pre>

	<pre> and ch.sourceid=co.companyid and co.companyid= ? and type!='salary' group by sourceid, targetid, type order by amount DESC limit 3); """" [AP-8] sql ="""" With blocked_account as (select /*+ AP_8 parallel(1) */ c.name as name, s.userID as id from customer c, savingaccount s where c.custid=s.userID and (c.isblocked=1 or c.isblocked=1) limit 100) select /*+ parallel(1) */ name, t.sourceid, t.targetid, sum(t.amount) as amount from blocked_account b, transfer t where b.id=t.sourceid group by name, t.sourceid, t.targetid UNION ALL select /*+ parallel(1) */ name, t.sourceid, t.targetid, sum(t.amount) as amount from blocked_account b, transfer t where b.id=t.targetid group by name, t.sourceid, t.targetid UNION ALL select /*+ parallel(1) */ name, c.sourceid, c.targetid, sum(c.amount) as amount from blocked_account b, checking c where b.id=c.sourceid group by name, c.sourceid, c.targetid UNION ALL select /*+ parallel(1) */ name, c.sourceid, c.targetid, sum(c.amount) as amount from blocked_account b, checking c where b.id=c.targetid group by name, c.sourceid, c.targetid order by amount DESC;"""" [AP-9] sql = """"select /*+ AP_9 parallel(8) */ c.name, sum(t.amount) as total_revenue from company c, transfer t where c.companyid = t.targetid and exists (select * from loanapps la </pre>
--	--

	<pre> where la.applicantid=c.companyid and loanapps_ts between str_to_date('2023-11-01','%Y-%m-%d') and (str_to_date('2023-11- 01','%Y-%m-%d') + 60)) and not exists (select * from loantrans lt where lt.applicantid=c.companyid and loantrans_ts between str_to_date('2023-11-01','%Y-%m-%d') and (str_to_date('2023-11- 01','%Y-%m-%d') + 60)) group by c.name order by total_revenue DESC;""" [AP-10] sql = """With big_account_inbound as (select /*+ AP_10 parallel(16) */ t.targetid as id, c.name as name, t.amount as amount from customer c, transfer t where t.transfer_ts between str_to_date('2023-11-01','%Y-%m-%d') and (cast('2023-11-01' as date) + 60) and c.custid=t.targetid group by name, id order by amount DESC limit 16), big_account_outbound as (select /*+ parallel(8) */ b.id, b.name as name, sum(t.amount) as amount from big_account_inbound b, transfer t where t.transfer_ts between str_to_date('2023-11-01','%Y-%m-%d') and (cast('2023-11-01' as date) + 60) and b.id=t.sourceid group by b.id, b.name) select /*+ parallel(16) */ bin.id, bin.name, (sum(bin.amount)-sum(bout.amount)) as revenue from big_account_inbound bin, big_account_outbound bout where bin.id=bout.id group by bin.id, bin.name order by revenue DESC;""" [AP-11] </pre>
--	---

	<pre> sql = """select /*+ AP_11 parallel(50) */ co.companyid, cu.custid, sum(t.amount) as invest_amount, count(t.amount) as count, avg(sa.balance) as saving_balance, avg(ca.balance) as checking_balance, avg(la.amount) as avg_loan_amount from company co, transfer t, customer cu, savingaccount sa, loanapps la, checkingaccount ca where t.targetid =co.companyid and t.sourceid =cu.custid and cu.custid =sa.userid and cu.custid =ca.userid and la.applicantid =cu.custid and t.type='invest' and sa.balance> 1000 and ca.balance> 1000 and co.category in ('software_IT','internet_service', 'telecommunication', 'technology_service', 'computer_communication_manufacturing') group by co.companyid, cu.custid order by invest_amount DESC, count DESC, saving_balance DESC, checking_balance DESC, avg_loan_amount DESC;""" [AP-12] sql = """select /*+ AP_12 parallel(50) */ co.companyid as cid, count(case when lt.status='accept' then 1 else null end) * 100.0/count(la.id) as rate from company co, loanapps la, loantrans lt where co.category =? and co.companyid=la.applicantid and la.applicantid=lt.applicantid group by cid order by rate DESC limit 10;""" [AP-13] sql = """select /*+ AP_13 parallel(8) */ sourceid,targetid ,province,amount, rank() over (partition by province order by amount desc) as ranking from transfer tr,customer cu where tr.targetid = cu.custid and tr.sourceid = ?""" </pre>
OLTP 执行 SQL 语句	<pre> [TP-1] sql = "select /*+ TP-1 */ custid from customer where custid=?;" </pre>

	[TP-2] sql = "select /*+ TP-2 */ companyID from company where companyID=?;" [TP-3] sql = "select /*+ TP-3 */ accountid from SAVINGACCOUNT where accountid=?;" [TP-4] sql = "select /*+ TP-4 */ accountid from CHECKINGACCOUNT where accountid=?;" [TP-5] sql = ""select /*+ TP-5 */ id from transfer where sourceid = ? or targetid = ? order by transfer_ts desc limit 1;"" [TP-6] sql = ""select /*+ TP-6 */ id from checking where sourceid = ? or targetid = ? order by checking_ts desc limit 1;"" [TP-7] sql = ""select /*+ TP-7 */ id from loanapps where applicantid=? order by loanapps_ts desc limit 1;"" [TP-8] sql = ""select /*+ TP-8 */ id from loantrans where applicantid=? order by loantrans_ts desc limit 1;"" [TP-9] sql = ["SELECT /*+ TP-9 */ balance FROM savingAccount WHERE accountID = ?", "UPDATE /*+ TP-9 */ savingAccount SET balance = balance - ? where accountID = ?", "UPDATE /*+ TP-9 */ savingAccount SET balance = balance + ? where accountID = ?",
--	---

	<pre> "INSERT /*+ TP-9 */ INTO transfer(id,sourceID,targetID,amount,type,transfer_ts) VALUES (0, ?,?,?,?,?)"] [TP-10] sql = ["SELECT /*+ TP-10 */custID FROM CUSTOMER WHERE COMPANYID = ?", "UPDATE /*+ TP-10 */SAVINGACCOUNT SET BALANCE = BALANCE - ? WHERE ACCOUNTID = ?", "UPDATE /*+ TP-10 */SAVINGACCOUNT SET BALANCE = BALANCE + ? WHERE ACCOUNTID = ?", "INSERT /*+ TP-10 */INTO TRANSFER(id,sourceID,targetID,amount,type,transfer_ts) VALUES (0, ?,?,?,?,?)"] [TP-11] sql = ["SELECT /*+ TP-11 */ balance FROM CHECKINGACCOUNT WHERE accountID = ?", "UPDATE /*+ TP-11 */ CHECKINGACCOUNT SET balance = balance - ? where accountID = ?", "UPDATE /*+ TP-11 */ CHECKINGACCOUNT SET balance = balance + ? where accountID = ?", "INSERT /*+ TP-11 */ INTO CHECKING VALUES (0, ?,?,?,?,?)"] [TP-12] sql = ["SELECT /*+ TP-12 */ loan_balance FROM customer WHERE custID = ?", "UPDATE /*+ TP-12 */ customer SET loan_balance = loan_balance - ? where custID = ?", "SELECT /*+ TP-12 */ loan_balance FROM company WHERE companyID = ?", "UPDATE /*+ TP-12 */ company SET loan_balance = loan_balance - ? where companyID = ?", </pre>
--	---

	<pre> "INSERT /*+ TP-12 */ INTO LOANAPPS VALUES (0, ?,?,?,?,?);"] [TP-13] sql = ["INSERT /*+ TP-13 */ INTO LOANTRANS VALUES(DEFAULT, ?,?,?,?,?);", "UPDATE /*+ TP-13 */ CUSTOMER SET loan_balance = loan_balance + ? where custID = ?", "UPDATE /*+ TP-13 */ COMPANY SET loan_balance = loan_balance + ? where companyID = ?", "UPDATE /*+ TP-13 */ LOANAPPS SET STATUS=? WHERE ID=?;"] [TP-14] sql = ["UPDATE /*+ TP-14 */ SAVINGACCOUNT SET BALANCE = BALANCE + ? WHERE ACCOUNTID = ?;", "UPDATE /*+ TP-14 */ LOANTRANS SET STATUS='lent', LOANTRANS_TS=? WHERE ID=?;"] [TP-15] sql = ["SELECT /*+ TP-15 use_das(LOANTRANS) index(LOANTRANS idx_loantrans_1) */ id, APPLICANTID, duration, LOANTRANS_TS FROM LOANTRANS WHERE status='lent' order by LOANTRANS_TS limit 1;", "UPDATE /*+ TP-15 */ LOANTRANS SET delinquency=1, LOANTRANS_TS=? WHERE id=?;"] [TP-16] sql = ["SELECT /*+ TP-16 */ balance from SAVINGACCOUNT WHERE ACCOUNTID = ?;" </pre>
--	--

	<pre> "UPDATE /*+ TP-16 */ SAVINGACCOUNT SET BALANCE = BALANCE - ? WHERE ACCOUNTID = ?;", "UPDATE /*+ TP-16 */ LOANTRANS SET status='repaid',LOANTRANS_TS=? WHERE id=?;"] [TP-17] sql = ["select /*+ TP-17 */ balance,isblocked from savingAccount where accountid = ?", "select /*+ TP-17 */ balance,isblocked from checkingAccount where accountid =?", "UPDATE /*+ TP-17 */ SAVINGACCOUNT SET BALANCE = BALANCE - ? WHERE ACCOUNTID = ?", "update /*+ TP-17 */ checkingAccount set balance = balance + ? where accountid = ?"] [TP-18] sql = ["select /*+ TP-18 */ balance,isblocked from savingAccount where accountid = ?", "select /*+ TP-18 */ balance,isblocked from checkingAccount where accountid =?", "update /*+ TP-18 */ checkingAccount set balance = balance - ? where accountid = ?", "UPDATE /*+ TP-18 */ SAVINGACCOUNT SET BALANCE = BALANCE + ? WHERE ACCOUNTID = ?"] </pre>
OLXP 执行 SQL 语句	<pre> [AT-1] sql = ["SELECT /*+ AT-1 BEGIN_OUTLINE_DATA FULL(@\"SEL\$1\" \"hy1500x2p\".\"savingaccount\"@\"SEL\$1\") OPTIMIZER_FEATURES_ENABLE('4.3.5.1') END_OUTLINE_DATA */ isblocked,balance FROM savingAccount WHERE accountID = ?", "select /*+ AT-1 BEGIN_OUTLINE_DATA LEADING(@\"SEL\$1\" (\"t\"@\"SEL\$1\" \"s\"@\"SEL\$1\")) USE_NL(@\"SEL\$1\" \"s\"@\"SEL\$1\") </pre>

	<pre> INDEX(@\"SEL\$1\" \"t\"@\"SEL\$1\" \"transfer_idx\") FULL(@\"SEL\$1\" \"s\"@\"SEL\$1\") USE_DAS(@\"SEL\$1\" \"s\"@\"SEL\$1\") OPTIMIZER_FEATURES_ENABLE('4.3.5.1') END_OUTLINE_DATA */ count(*) from transfer t, savingaccount s where t.targetid=? and s.isblocked=1 and t.sourceid = s.accountid;\", \"UPDATE /*+ AT-1 BEGIN_OUTLINE_DATA FULL(@\"UPD\$1\" \"hy1500x2p\".\"savingaccount\"@\"UPD\$1\") OPTIMIZER_FEATURES_ENABLE('4.3.5.1') END_OUTLINE_DATA */ savingAccount SET balance = balance - ? where accountID = ?\", \"UPDATE /*+ AT-1 BEGIN_OUTLINE_DATA FULL(@\"UPD\$1\" \"hy1500x2p\".\"savingaccount\"@\"UPD\$1\") OPTIMIZER_FEATURES_ENABLE('4.3.5.1') END_OUTLINE_DATA */ savingAccount SET balance = balance + ? where accountID = ?\", \"INSERT /*+ AT-1 */ INTO transfer(id,sourceID,targetID,amount,type,transfer_ts) VALUES (0, ?,?,?,?,?)\"] [AT-2] sql = [\"SELECT /*+ AT-2 BEGIN_OUTLINE_DATA FULL(@\"SEL\$1\" \"hy1500x2p\".\"checkingaccount\"@\"SEL\$1\") OPTIMIZER_FEATURES_ENABLE('4.3.5.1') END_OUTLINE_DATA */ isblocked,balance FROM checkingAccount WHERE accountID = ?\", \"select /*+ AT-2 index(CH PRIMARY) BEGIN_OUTLINE_DATA LEADING(@\"SEL\$1\" (\"ch\"@\"SEL\$1\" \"ca\"@\"SEL\$1\")) USE_NL(@\"SEL\$1\" \"ca\"@\"SEL\$1\") FULL(@\"SEL\$1\" \"ch\"@\"SEL\$1\") FULL(@\"SEL\$1\" \"ca\"@\"SEL\$1\") USE_DAS(@\"SEL\$1\" \"ca\"@\"SEL\$1\") OPTIMIZER_FEATURES_ENABLE('4.3.5.1') END_OUTLINE_DATA */ count(*) from checking ch, checkingAccount ca where ch.sourceid=? and ca.isblocked=1 and ch.targetid=ca.accountid;\", \"UPDATE /*+ AT-2 BEGIN_OUTLINE_DATA FULL(@\"UPD\$1\" \"hy1500x2p\".\"checkingaccount\"@\"UPD\$1\") OPTIMIZER_FEATURES_ENABLE('4.3.5.1') END_OUTLINE_DATA */ checkingAccount SET balance = balance - ? where accountID = ?\", </pre>
--	---

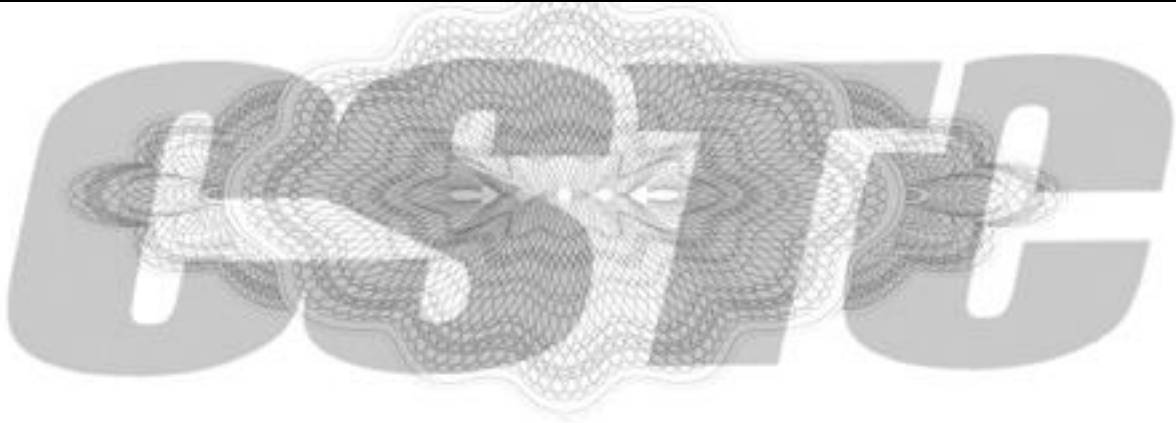
	<pre> "UPDATE /*+ AT-2 BEGIN_OUTLINE_DATA FULL(@\"UPD\$1\" \"hy1500x2p\".\"checkingaccount\"@\"UPD\$1\") OPTIMIZER_FEATURES_ENABLE('4.3.5.1') END_OUTLINE_DATA */ checkingAccount SET balance = balance + ? where accountID = ?", "INSERT /*+ AT-2 */ INTO checking VALUES (0, ?,?,?,?,?)"] [AT-3] sql = ["SELECT /*+ AT-3 */ isblocked FROM SavingAccount WHERE accountID = ?", """"Select /*+ AT-3 */ (inbound.t_amount-outbound.t_amount) as t_amount from (select sa.accountid as sid, sum(t.amount) as t_amount from savingaccount sa, transfer t where sa.accountid=? and sa.accountid=t.sourceid group by sid) as outbound, (select /*+ AT-3 */ sa.accountid as sid, sum(t.amount) as t_amount from savingaccount sa, transfer t where sa.accountid=? and sa.accountid=t.targetid group by sid) as inbound""", "INSERT /*+ AT-3 */ INTO LOANTRANS VALUES(0, ?,?,?,?,?);", "UPDATE /*+ AT-3 */ LoanApps SET status = 'accept',loanapps_ts=? where id = ?", " UPDATE /*+ AT-3 */ LoanApps SET status = 'reject',loanapps_ts=? where id = ?"] [AT-3.1] sql = "select /*+ AT-3.1 */ t1.applicantid, t1.id, t1.loanapps_ts, t1.duration from loanapps t1 where t1.id >= cast(round(rand() * 902500404,0) as SIGNED) and t1.status='under_review' order BY t1.id limit 1;" [AT-4] sql = ["select /*+ AT-4 */ count(distinct sa2.accountid) from transfer t, savingAccount sa1, savingAccount sa2, loantrans lt where sa1.accountid = ? and sa1.accountid = </pre>
--	--

	<pre> t.targetid and t.sourceid = sa2.accountid and lt.applicantid=sa2.accountid and lt.status='accept';", "UPDATE /*+ AT-4 */ savingAccount SET balance = balance + ? where accountID = ?;", "UPDATE /*+ AT-4 */ LoanTrans SET status = 'lent', contract_timestamp=? where id = ?;"] [AT-4.1] sql = ""SELECT /*+ AT-4.1 */ applicantid, amount, id, loantrans_ts FROM loantrans where status='accept' LIMIT 1"" [AT-5] sql = ["SELECT /*+ AT-5 */ id FROM LoanTrans WHERE applicantID = ? and date(contract_timestamp) + duration < str_to_date('2024-01-01','%Y-%m-%d') and status!='lent';", "UPDATE /*+ AT-5 */ LoanTrans SET delinquency=1 where id = ?"] [AT-5.1] sql ="SELECT /*+ AT-5.1 */ applicantid FROM loantrans where status='lent' LIMIT 1;" [AT-6.1] sql = "SELECT /*+ AT-6.1 parallel(8) */ applicantid FROM loantrans where status='lent' and delinquency=1 LIMIT 1" [AT-6] sql = ["SELECT /*+ AT-6 */ count(applicantid) FROM LoanTrans WHERE date(contract_timestamp) + duration + 365 < str_to_date('2024-01- 01','%Y-%m-%d') and applicantid=?;", "UPDATE /*+ AT-6 */ savingaccount SET isBlocked = 1 where accountID = ?;", </pre>
--	--

	<pre> "UPDATE /*+ AT-6 */ checkingaccount SET isBlocked = 1 where accountID = ?;"] [IQ-1] sql="""select /*+ IQ-1 parallel(1) */ t.transfer_ts, t.amount, c.custid, c.name from transfer t, customer c where t.sourceid =? and t.targetid = c.custid UNION select t.transfer_ts, t.amount, c.custid, c.name from transfer t, customer c where t.targetid=? and t.sourceid=c.custid order by transfer_ts Limit 10 """ [IQ-2] sql="""select /*+ IQ-2 parallel(1) */ ch.checking_ts, ch.amount, co.companyid, co.name from checking ch, company co where ch.sourceid = ? and ch.targetid= co.companyid UNION select /*+ parallel(1) */ ch.checking_ts, ch.amount, co.companyid, co.name from checking ch, company co where ch.targetid=? and ch.sourceid= co.companyid order by checking_ts Limit 10;""" [IQ-3] sql="""select /*+ IQ-3 parallel(1) */ co.name, cu.custid, cu.name, sa.isblocked, ca.isblocked from customer cu, company co, savingaccount sa, checkingaccount ca where co.companyid = ? and cu.companyid= co.companyid and sa.accountid=cu.custid and ca.accountid=cu.custid;""" [IQ-4] </pre>
--	--

	<pre> sql="""(select /*+ IQ-4 parallel(2) */ co.name, t.amount, t.transfer_ts from company co, transfer t where t.sourceid = ? and t.type='invest' and t.targetid= co.companyid) UNION (select /*+ parallel(2) */ cu.name, t.amount, t.transfer_ts from customer cu, transfer t where t.sourceid =? and t.type='invest' and t.targetid= cu.custid);""" [IQ-5] sql="""select /*+ IQ-5 parallel(2) */ t.amount, t.targetid, co.name, cu.name from loantrans lt Left join transfer t on lt.applicantid = t.sourceid Left join customer cu on t.targetid=cu.custid Left join company co on t.targetid=co.companyid where t.sourceid =? and lt.delinquency = 1;""" [IQ-5.1] sql="SELECT /*+ IQ-5.1 */ applicantid FROM loantrans LIMIT 1" [IQ-6] sql="""with abnormal_accounts as (select /*+ parallel(16) */ distinct(applicantid) as id, amount from loantrans lt1 where contract_timestamp is not null and date(contract_timestamp) + duration < str_to_date('2024-01-01','%Y-%m-%d') order by amount desc limit 100) select /*+ use_hash(la,lt2) parallel(16) */ aa.id, lt2.amount, lt2.loantrans_ts, lt2.status, la.loanapps_ts from loantrans lt2, loanapps la, abnormal_accounts aa where aa.id=lt2.applicantid and la.id= lt2.appid;""" [fresh] sql = """select /*+ parallel(16) */ t.fresh_ts, t.transfer_ts, t.amount, c.custid, c.name </pre>
--	--

	from transfer t, customer c where t.targetid=? and t.targetid=c.custid and t.fresh_ts is not null order by t.fresh_ts DESC limit 1 "" [AppQueue] sql = "select id,applicantid,amount,duration,status,loanapps_ts from loanapps where status=? limit 10000" [TransQueue] sql = "select id, applicantid, appid, amount, status, loantrans_ts, duration, contract_timestamp, delinquency from loantrans where status = ? limit 10000"
--	---



2.2. 详细测试结果

2.2.1. OLAP 响应时间

SQL 语句序号	Max (ms)	Min (ms)	Avg (ms)	95% (ms)	99% (ms)
AP Query1	214.00	7.00	16.10	26.95	36.79
AP Query2	69.00	6.00	14.62	27.00	37.79
AP Query3	393.00	15.00	35.06	54.00	71.00
AP Query4	458.00	11.00	28.70	47.00	63.00
AP Query5	269.00	19.00	40.49	64.95	79.79
AP Query6	4221.00	1337.00	3299.73	3927.00	4109.37
AP Query7	515.00	26.00	54.55	74.00	94.79
AP Query8	351.00	68.00	157.29	211.00	235.58
AP Query9	473.00	10.00	40.25	71.00	94.79
AP Query10	319.00	20.00	53.55	92.00	120.79
AP Query11	10416.00	2682.00	5442.60	6362.85	6961.43.63
AP Query12	9758.00	5377.00	7159.84	8117.95	8484.48
AP Query13	118.00	11.00	26.68	46.00	66.00

2.2.2. OLTP 响应时间

SQL 语句序号	Max (ms)	Min (ms)	Avg (ms)	95% (ms)	99% (ms)
TP Transaction1	4428	0	42.16	189.00	249.00
TP Transaction2	4430	0	42.03	188.00	248.00
TP Transaction3	4428	0	42.19	189.00	249.00
TP Transaction4	4428	0	42.03	189.00	249.00
TP Transaction5	4431	0	45.30	194.00	254.00
TP Transaction6	4429	0	43.92	191.00	251.00
TP Transaction7	4445	1	48.75	190.00	250.00
TP Transaction8	4452	1	49.11	191.00	251.00
TP Transaction9	5808	0	269.56	893.00	1195.00
TP Transaction10	14706	20	384.58	1126.00	1462.00
TP Transaction11	5823	1	269.02	891.00	1195.00
TP Transaction12	5300	2	238.66	712.00	786.00
TP Transaction13	4568	1	189.22	486.00	686.00

SQL 语句序号	Max (ms)	Min (ms)	Avg (ms)	95% (ms)	99% (ms)
TP Transaction14	25683	1	152.32	412.00	531.00
TP Transaction15	123808	2	1547.92	6246.00	13427.00
TP Transaction16	20848	0	107.20	312.00	605.00
TP Transaction17	4662	0	276.32	556.00	601.00
TP Transaction18	4587	0	276.31	556.00	600.00

2.2.3. OLXP 响应时间

SQL 语句序号	Max (ms)	Min (ms)	Avg (ms)	95% (ms)	99% (ms)
Interavtive Query1	175945.00	140830	149140.00	175945.00	175945.00
Interavtive Query2	1058	5	130.85	1058.00	1058.00
Interavtive Query3	112	28	52.00	112.00	112.00
Interavtive Query4	497	17	183.18	497.00	497.00
Interavtive Query5	247	6	43.33	247.00	247.00
Interavtive Query6	302417	213996	265520.25	302417.00	302417.00
Analytical Transaction1	8391	0	7.86	10.00	25.00
Analytical Transaction2	2537	0	6.39	7.00	17.00
Analytical Transaction3	2613	1	9.24	10.90	15.00
Analytical Transaction4	2822	3	7.79	9.00	13.00
Analytical Transaction5	2368	1	12.54	32.00	53.00
Analytical Transaction6	55876	3	13759.24	44427.00	48247.00

3. 测试环境

3.1. 软、硬件配置

数据库服务器（厂商设备 1 台）：

硬件 型号：R6250H0

CPU：Hygon C86 7490 64 核*2 2999MHz（超线程 256 核）

内存：1510GB

硬盘：13TB（NVMe SSD）

软件 操作系统：银河麒麟高级服务器操作系统 V10（SP3）

数据库：OceanBase V4

客户端服务器（厂商设备 1 台）：

硬件 型号：WOQU Q750XF

CPU：Intel Xeon Gold 6448H 3200MHz

内存：502GB

硬盘：7TB（NVMe SSD）

软件 操作系统：银河麒麟高级服务器操作系统 V10（SP3）

测试工具：HyBench-EE V1.0

测试工具：HyBench-EE V1.0（CSTC10245050）

网络带宽：1000Mbps

附件：表定义及索引定义语句

创建表单：

```
CREATE TABLE IF NOT EXISTS customer (  
    custID int ,  
    companyID int,  
    gender varchar(6),  
    name varchar(15),  
    age int,  
    phone varchar(11),  
    province varchar(15),  
    city varchar(15),  
    loan_balance decimal(15,2),  
    saving_credit int,  
    checking_credit int,  
    loan_credit int,  
    Isblocked int,  
    created_date Date,  
    last_update_timestamp timestamp,  
    primary key(custid)  
) partition by hash(custid) partitions 2
```

;

```
CREATE TABLE IF NOT EXISTS company (  
    companyID int PRIMARY KEY,  
    name varchar(100),  
    category varchar(100),  
    staff_size int,  
    loan_balance decimal(15,2),  
    phone varchar(11),  
    province varchar(15),  
    city varchar(15),  
    saving_credit int,  
    checking_credit int,  
    loan_credit int,  
    Isblocked int,
```

```
created_date Date,  
last_update_timestamp timestamp default current_timestamp  
) partition by hash(companyid) partitions 2  
;  
CREATE TABLE IF NOT EXISTS savingAccount (  
    accountID int PRIMARY KEY,  
    userID int,  
    balance decimal(15,2),  
    Isblocked int,  
    savingAccount_ts timestamp default current_timestamp  
) partition by hash(accountid) partitions 2  
;  
CREATE TABLE IF NOT EXISTS checkingAccount (  
    accountID int PRIMARY KEY,  
    userID int,  
    balance decimal(15,2),  
    Isblocked int,  
    checkingAccount_ts timestamp  
) partition by hash(accountid) partitions 2  
;  
create table if not exists transfer (  
    id bigint ,  
    sourceid int,  
    targetid int,  
    amount decimal(15,2),  
    type varchar(100),  
    transfer_ts timestamp default current_timestamp,  
    fresh_ts timestamp default current_timestamp,  
    primary key(sourceid,transfer_ts,id)  
) partition by hash (sourceid) partitions 2  
;  
create table if not exists checking (  
    id int ,  
    sourceid int,  
    targetid int,
```



```
amount decimal(15,2),
type varchar(100),
checking_ts timestamp default current_timestamp,
primary key(sourceid,checking_ts,id)
) partition by hash(sourceid) partitions 2
;
CREATE TABLE IF NOT EXISTS loanapps (
id int ,
applicantID int,
amount decimal(15,2),
duration int,
status varchar(12),
loanapps_ts timestamp default current_timestamp,
PRIMARY KEY(id)
) partition by hash(id) partitions 2
;
CREATE TABLE IF NOT EXISTS loantrans (
id int ,
applicantID int,
appID int,
amount decimal(15,2),
status varchar(12),
loantrans_ts timestamp,
duration int,
contract_timestamp timestamp default current_timestamp,
delinquency int,
PRIMARY KEY(id)
) partition by hash(id) partitions 2;
```

添加自增 id :

```
alter table checking modify id int auto_increment;
alter table loanapps modify id int auto_increment;
alter table loantrans modify id int auto_increment;
alter table transfer modify id bigint auto_increment;
```

创建索引

```
CREATE /*+ parallel(200) */ INDEX idx_company_category_companyid ON
company(category, companyid);

create /*+ parallel(200) */ index idx_loanapps_1 on loanapps(status,id) global partition by
key(status) partitions 2;

CREATE /*+ parallel(200) */ INDEX idx_loantrans_applicantid_status ON
loantrans(applicantid, status) local;

CREATE /*+ parallel(200) */ INDEX idx_company_companyid_name ON company ( name);

CREATE /*+ parallel(200) */ INDEX idx_transfer_sourceid_amount ON transfer (sourceid,
amount);

CREATE /*+ parallel(200) */ INDEX idx_checking_sourceid_amount ON checking (sourceid,
amount);

CREATE /*+ parallel(200) */ INDEX idx_savingaccount_userid_balance ON
savingaccount(userid, balance);

CREATE /*+ parallel(200) */ INDEX idx_checkingaccount_userid_balance ON
checkingaccount(userid, balance);

CREATE /*+ parallel(200) */ INDEX idx_loanapps_applicantid_ts ON loanapps(applicantid,
loanapps_ts);

CREATE /*+ parallel(200) */ INDEX idx_loantrans_appid ON loantrans(appid);

create /*+ parallel(200) */ index idx_checking_2 on checking(targetid) global partition by
hash(targetid) partitions 2;

create /*+ parallel(200) */ index idx_customer_1 on customer(companyid) global global
partition by hash(companyid) partitions 2;

create /*+ parallel(200) */ index idx_loanapps on loanapps(applicantID) global partition by
hash(applicantID) partitions 2;

create /*+ parallel(200) */ index idx_loantrans_1 on
LOANTRANS(status,LOANTRANS_TS,delinquency)
storing
(id,applicantid,duration,amount,loantrans_ts) local;

create /*+ parallel(200) */ index transfer_idx on transfer(targetid) global partition by
hash(targetid) partitions 2;
```