阿里云Postgres数据库发展与规划

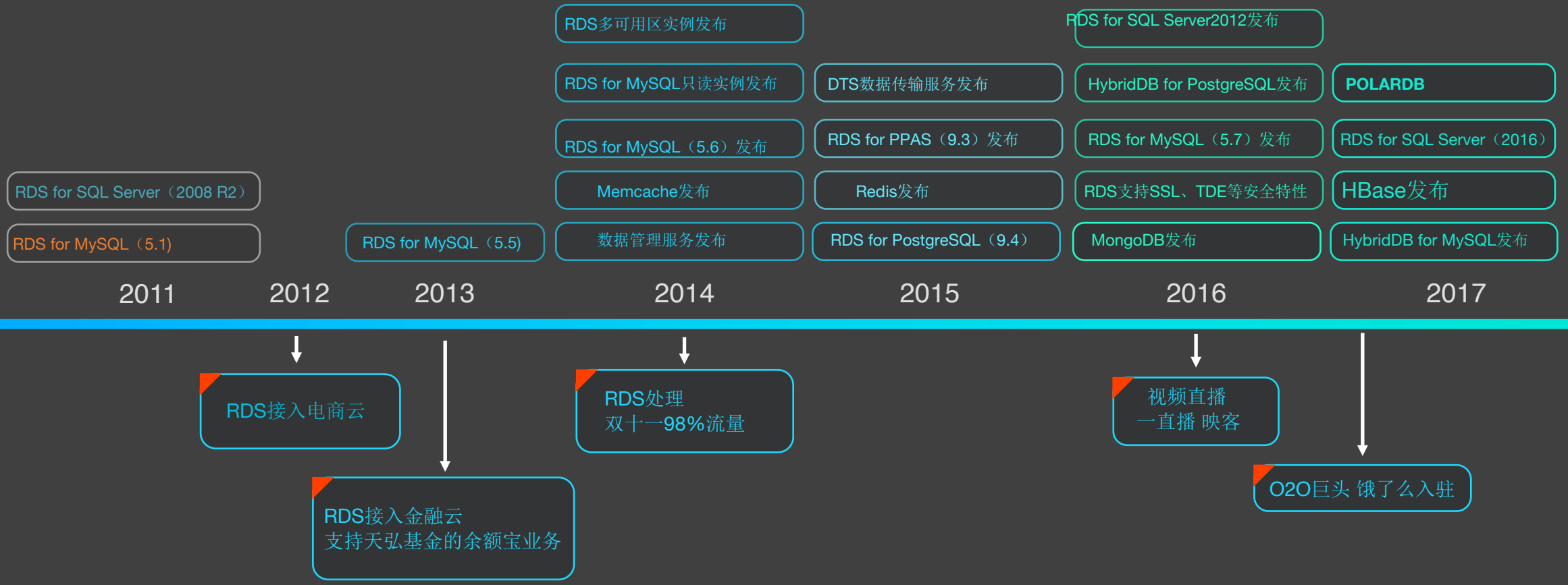
-Postgres不是只是数据库，Postgres是新生态

2019年03月08日 萧少聪(铁庵)

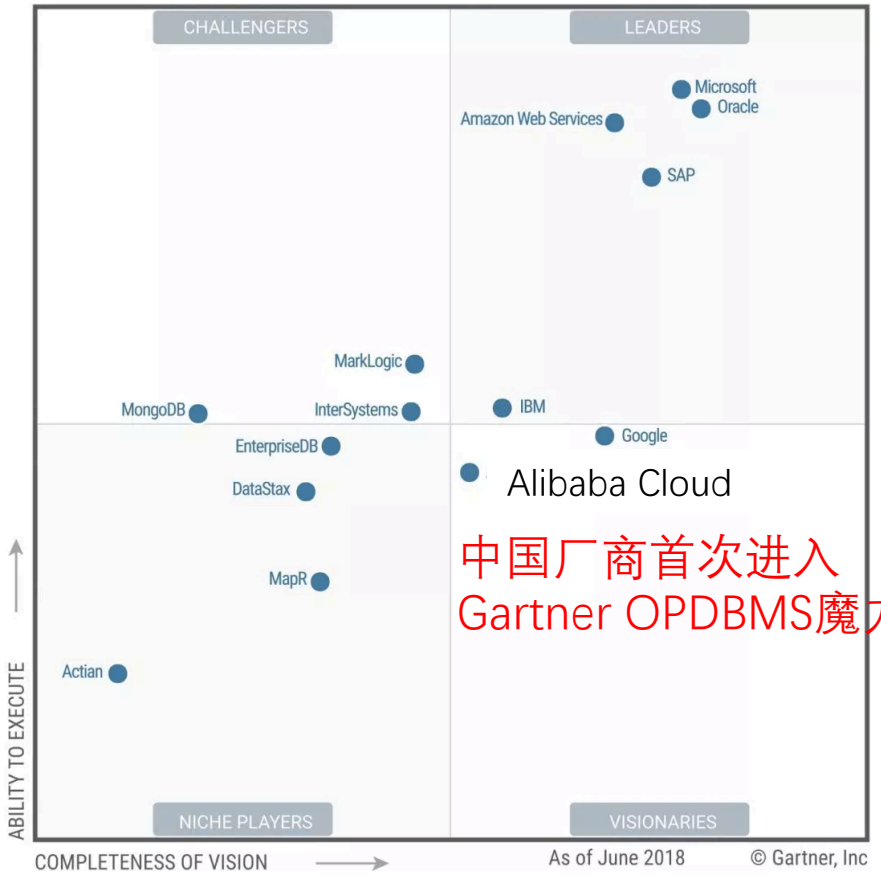
不断进击的数据库

The Histroy of ApsaraDB

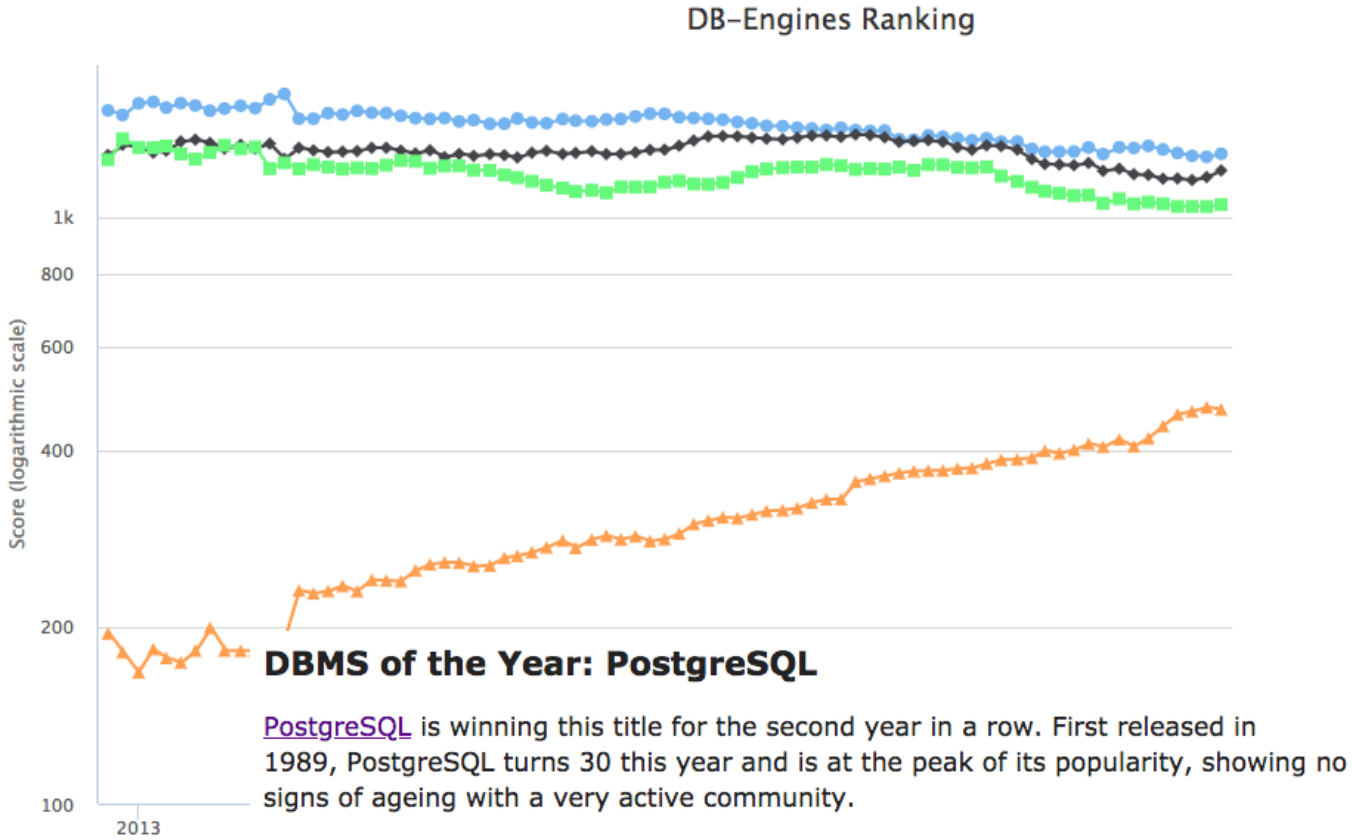
连续五年，每年1000项以上功能优化，为极致而生



数据库不一样的2018

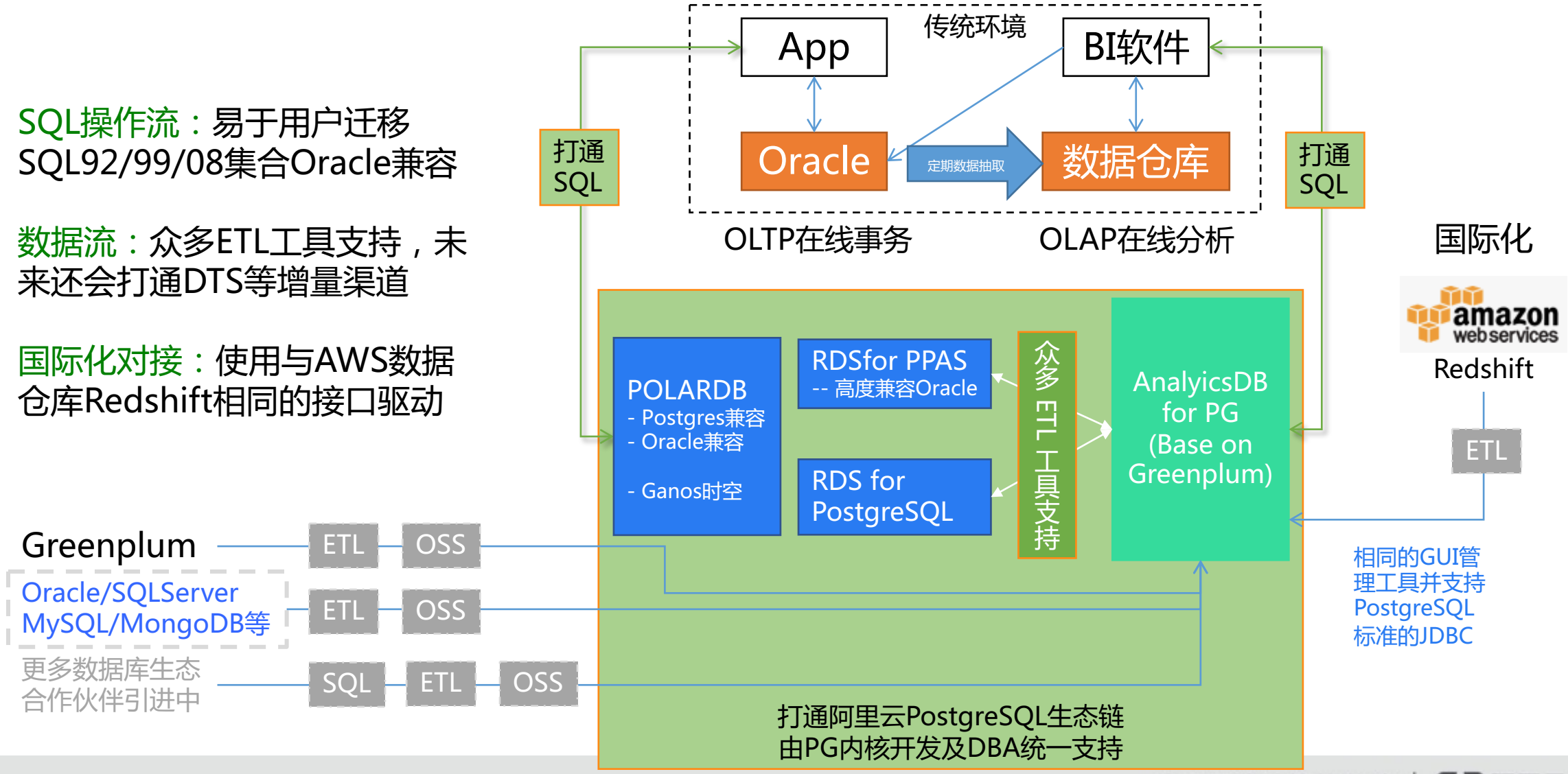


中国厂商首次进入
Gartner OPDBMS魔力象限

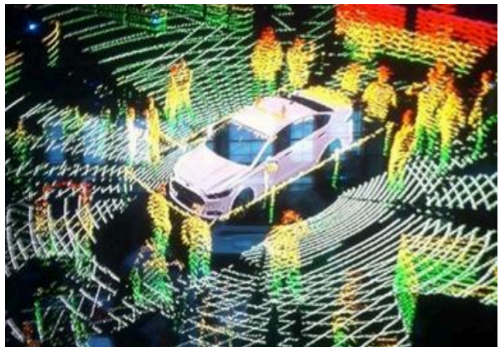


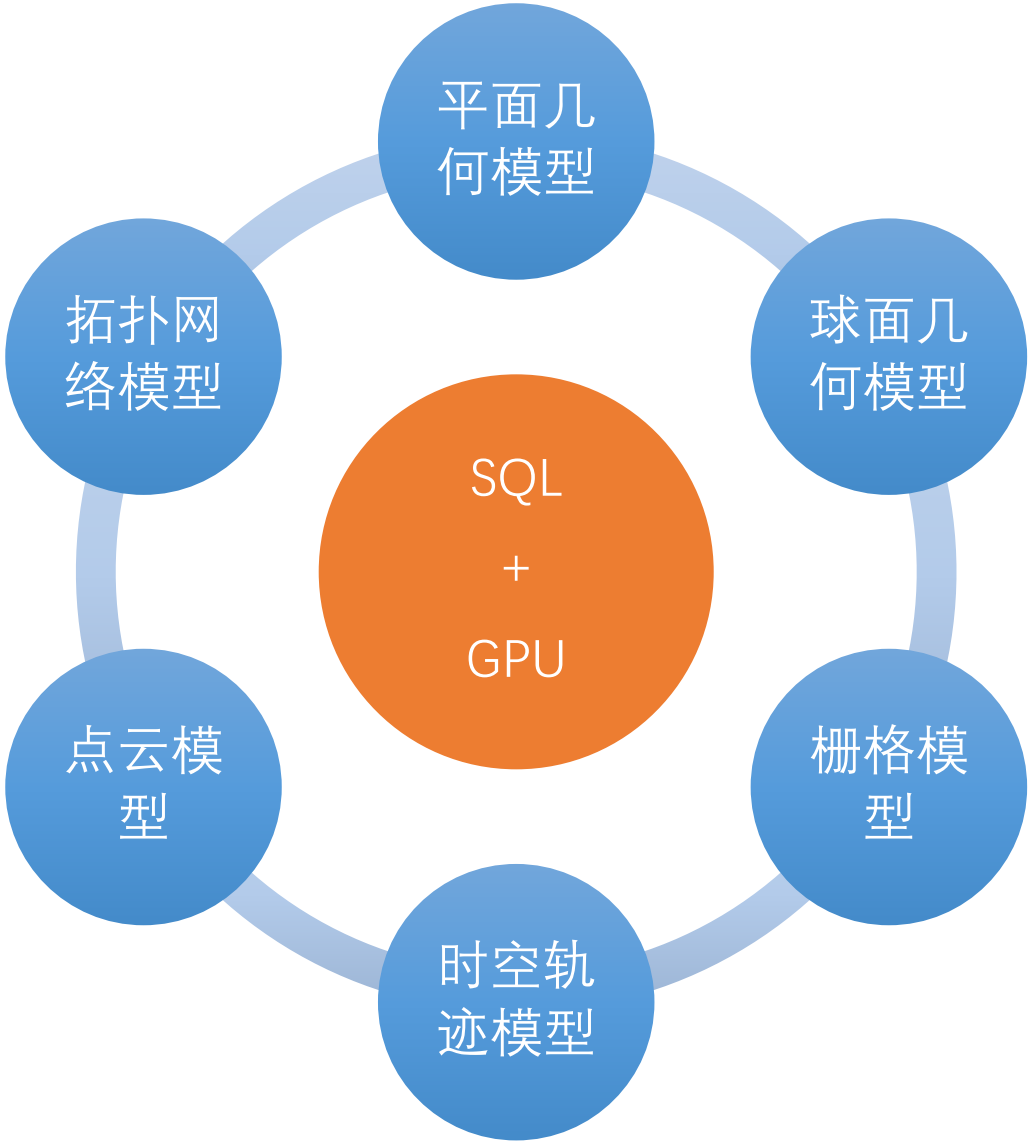
Postgres不是一个数据库，Postgres是一整套生态

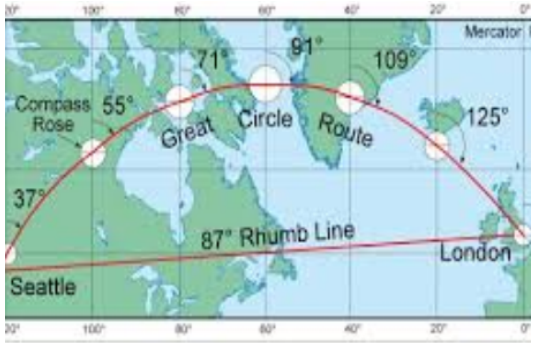
- SQL操作流：易于用户迁移SQL92/99/08集合Oracle兼容
- 数据流：众多ETL工具支持，未来还会打通DTS等增量渠道
- 国际化对接：使用与AWS数据仓库Redshift相同的接口驱动



PG Ganos — 时空多模型





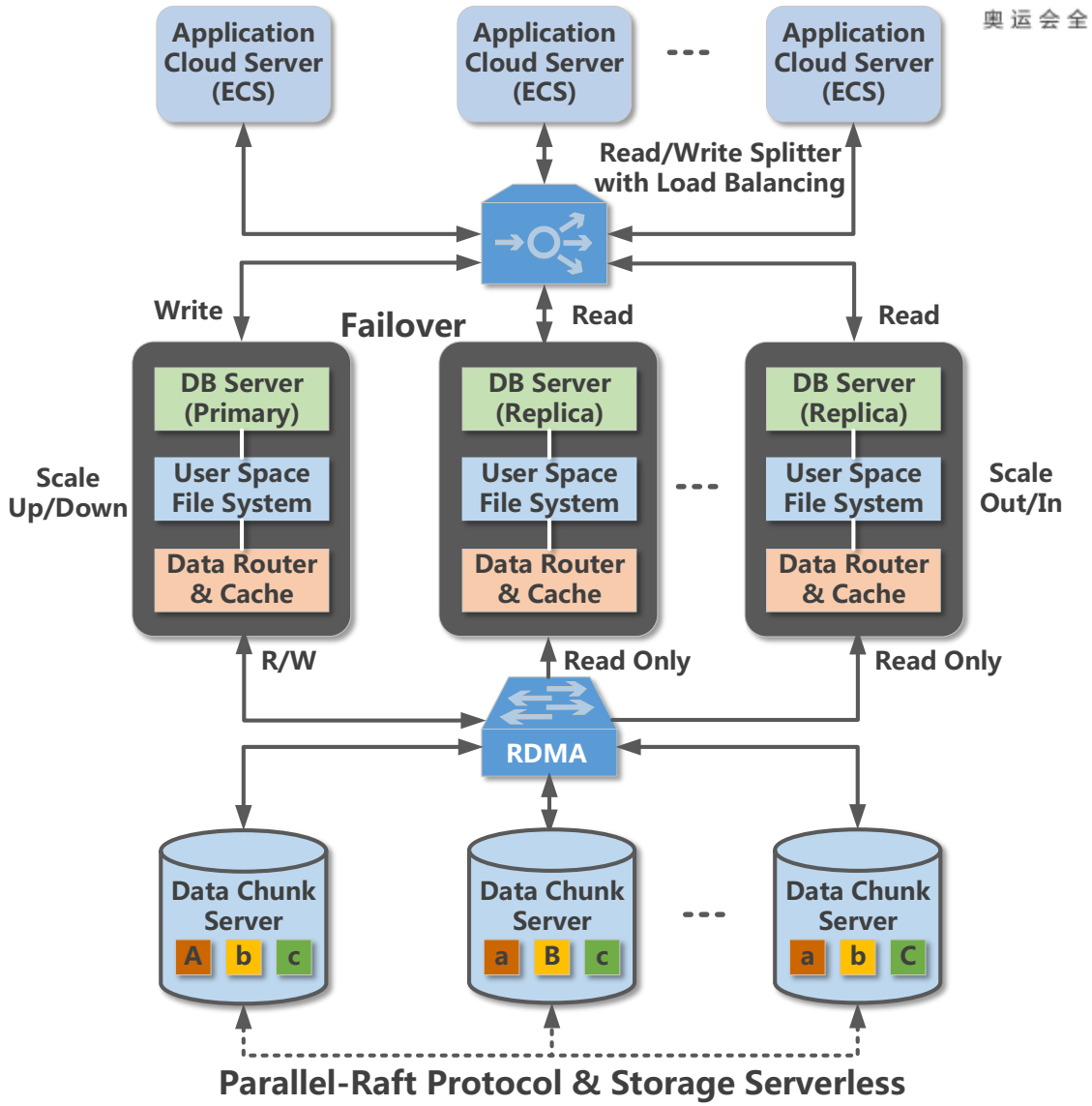


POLARDB产品架构

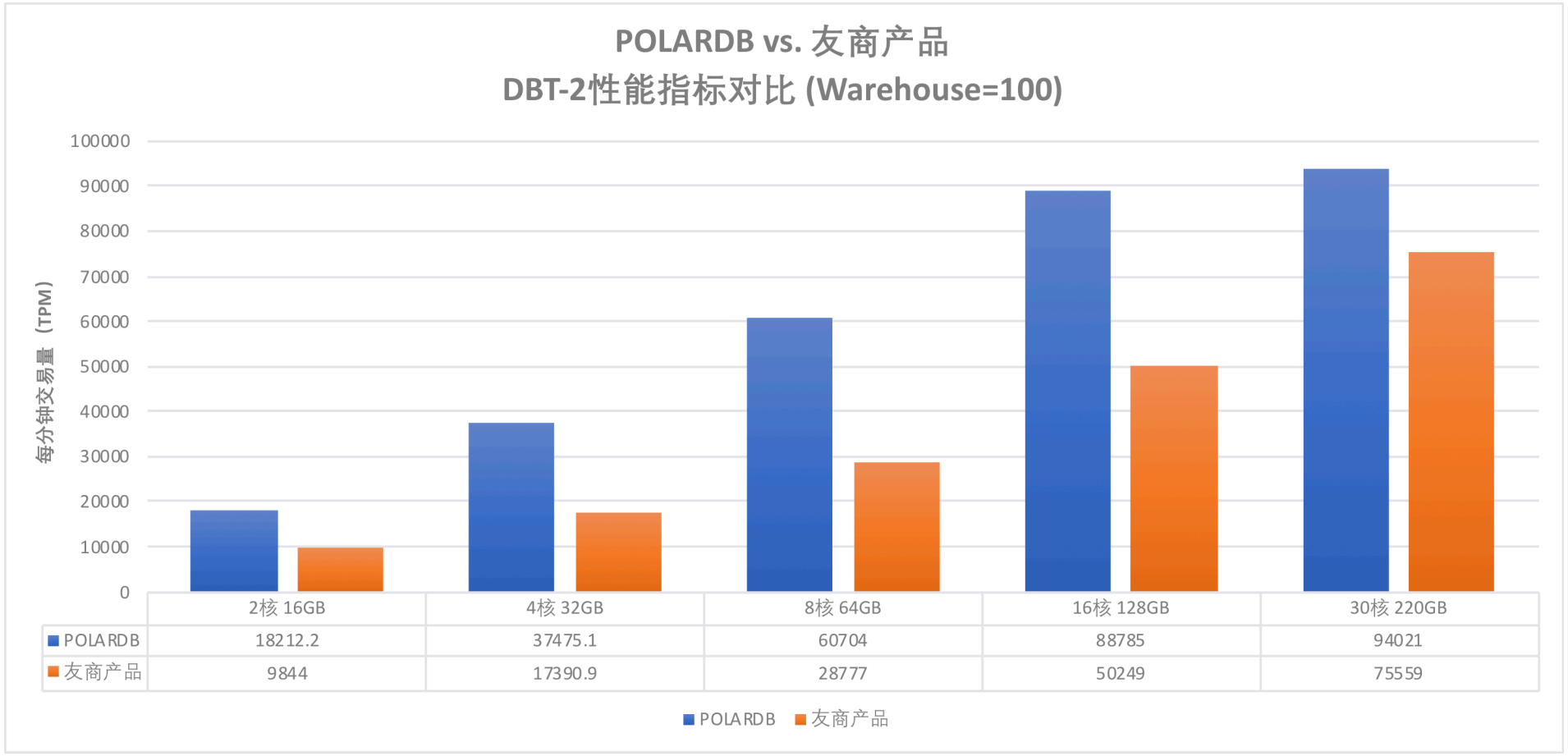
- 云原生数据库 = Severless + 多维弹性
  存储按数据实际使用量收费
  存储容量在线自适应扩容
  支持分钟级别升降配、只读节点增减

- 高可用、可扩展的读写分离架构
  Active-Active自带只读节点
  支持自适应负载均衡的读写分离访问
  只读节点毫秒级数据延迟
  最大支持16个只读节点

- 高可靠的共享分布式存储
  数据三副本
  海量数据秒级备份/按时间点恢复



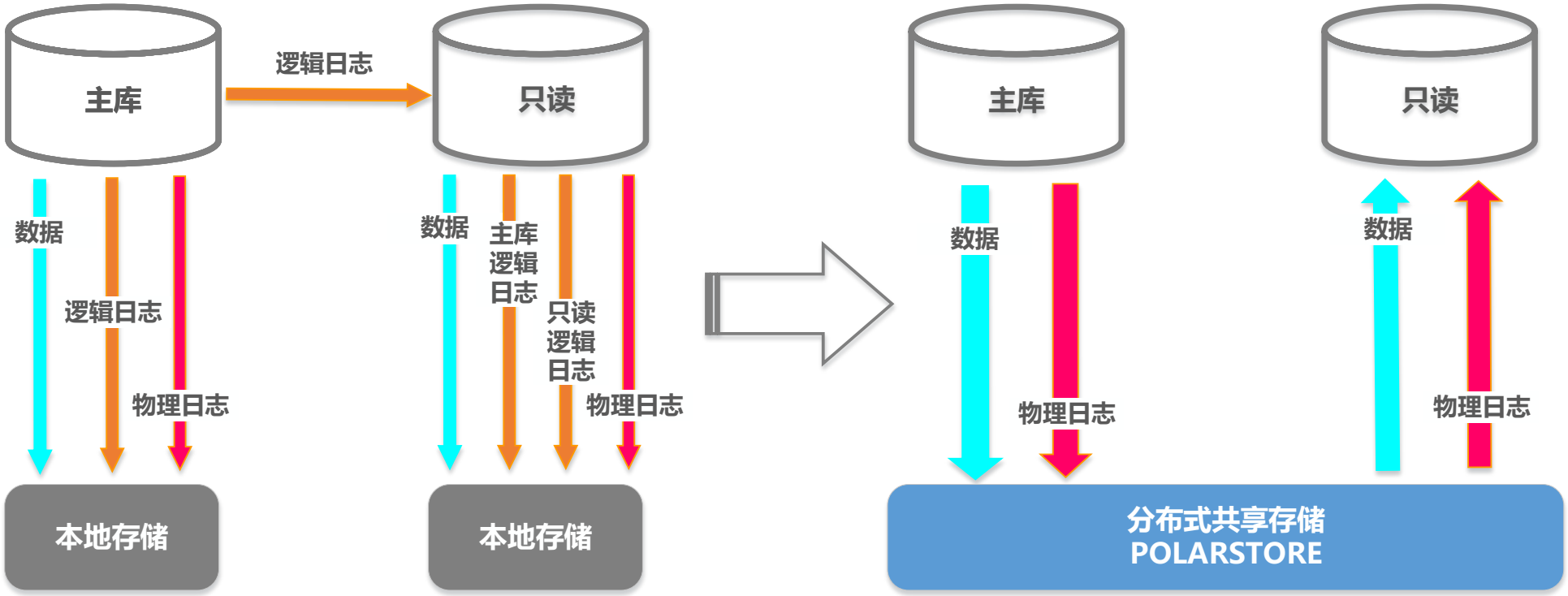
POLARDB 基准测试TPC-C



- DBT-2基准测试是一个衡量OLTP 系统性能和可伸缩性的基准测试项目。它通过模拟仓库和订单管理系统，测试广泛的数据库功能，包括查询、更新和小批量队列式事务。每分钟处理的交易量（TPM）是核心衡量指标。DBT-2是TPC-C基准测试的开源实现。

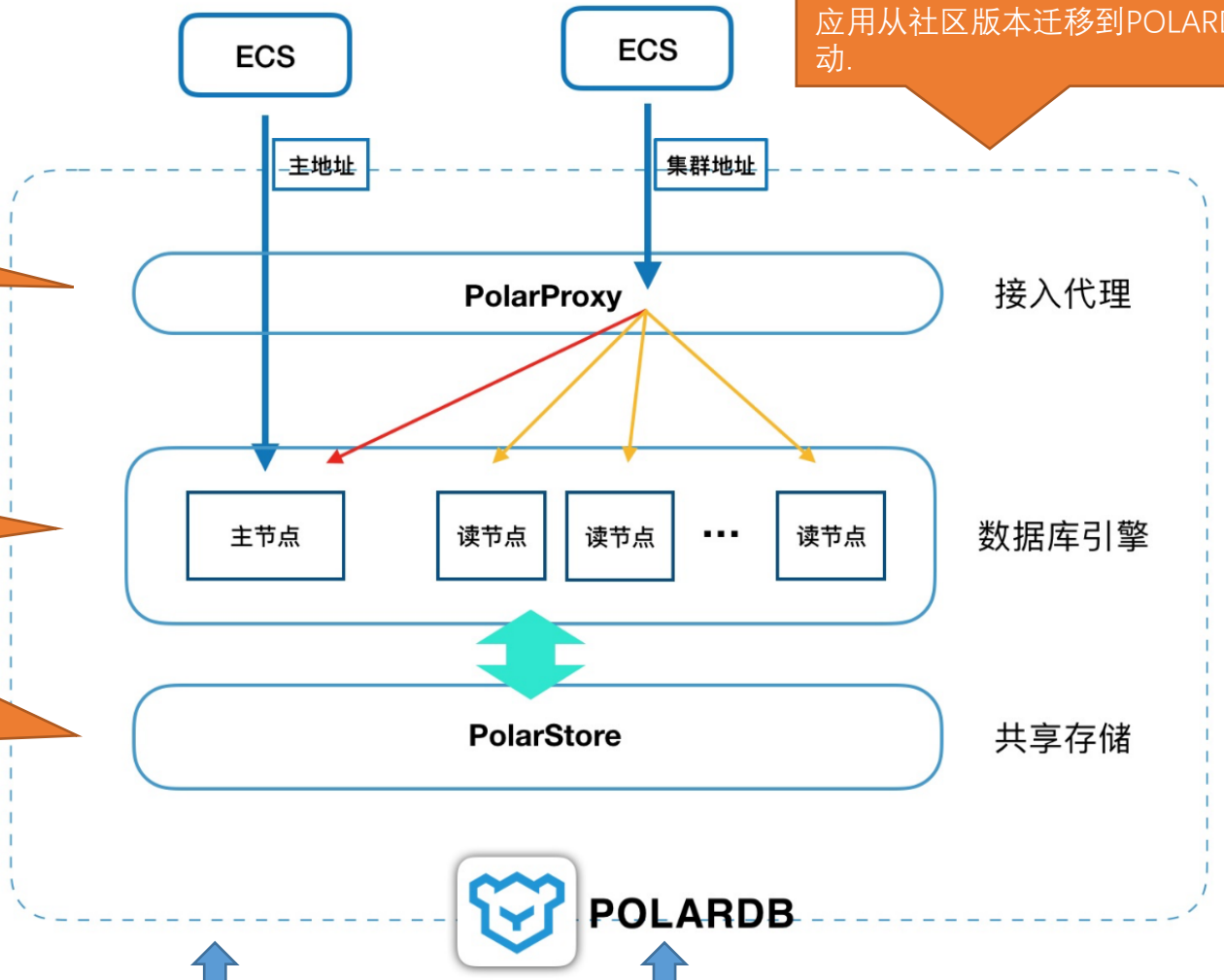
物理复制

- 基于Redo Log的物理复制
- 数据复制延迟 < 10ms



POLARDB 架构与优势解读

proxy集群,计算集群,存储集群浑然一体
应用从社区版本迁移到POLARDB无需任何改动.



读写分离
自动故障转移

分钟级弹性伸缩节点，可扩展至16个计算节点
只读节点毫秒级延迟

分布式共享存储，RPO=0
采用RDMA、Parallel Raft、3D Xpoint等最新的软硬件技术，性能最高提升6倍
Serverless，单实例可弹性扩展至100TB
价格优势：16个节点共用一份存储，相比传统读写分离架构，存储成本大幅降低。

OSS_FDW外部表：冷软分离，分级存储。
历史数据想存多久久存多久。

阿里云OSS海量对象存储

云生态
无缝对接

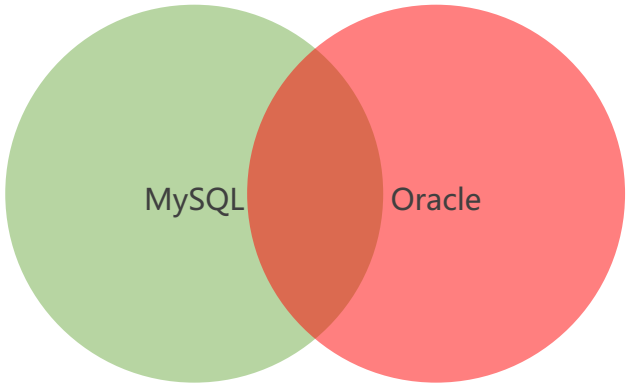
ADB

函数计算

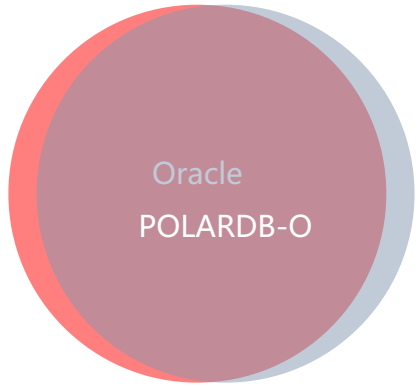
MAX
Compute

POLARDB-O 高度兼容Oracle

全自主可控技术，高度兼容Oracle，基于POLARDB软件件融合架构，提供极致性能。



MySQL与Oracle兼容度主要在标准数据类型及标准SQL，所有PL/SQL都要重写



POLARDB-O兼容Oracle近40种数据类型、100+函数及数据库包、PL/SQL语法、表分区语法、Hint优化标记、类AWR性能分析报告、DUAL表

- 相比迁移到MySQL，工作量减少90%以上
- 后续业务开发依然可以用Oracle语法
- 一次开发Oracle、POLARDB-O通用

云数据库POLARDB-O - Oracle兼容的数据类型

兼容Oracle数据类型包括：

- BLOB
- CLOB
- DATE
- INTEGER
- NUMBER
- REAL
- VARCHAR
- VARCHAR2
- NVARCHAR2
- 等

同时还支持自定义数据类型
CRETE TYPE

Name	Alias	Description
BLOB	LONG RAW, RAW(n), BYTEA	Binary data
BOOLEAN		Logical Boolean (true/false)
CHAR [*(n)*]	CHARACTER [(n)]	Fixed-length character string of n characters
CLOB	LONG, LONG VARCHAR	Long character string
DATE	TIMESTAMP(0)	Date and time to the second
DOUBLE PRECISION	FLOAT, FLOAT(25) - FLOAT(53)	Double precision floating-point number
INTEGER	INT, BINARY INTEGER, PLS INTEGER	Signed four-byte integer
NUMBER	DEC, DECIMAL, NUMERIC	Exact numeric with optional decimal places
NUMBER(p [, s])	DEC(p [, s]), DECIMAL(p [, s]), NUMERIC(p [, s])	Exact numeric of maximum precision, p, and optional scale, s
REAL	FLOAT(1) - FLOAT(24)	Single precision floating-point number
TIMESTAMP [(p)]		Date and time with optional, fractional second precision, *p*
TIMESTAMP [(p)] WITH TIME ZONE		Date and time with optional, fractional second precision, p, and with time zone
VARCHAR2(n)	CHAR VARYING(n), CHARACTER VARYING(n), VARCHAR(n)	Variable-length character string with a maximum length of *n* characters
XMLTYPE		XML data

云数据库POLARDB-O - Oracle兼容的DDL

```
CREATE [ GLOBAL TEMPORARY ] TABLE table_name (
  { column_name data_type [ DEFAULT default_expr ]
  [ column_constraint [ ... ] ] | table_constraint } [, ...]
  )
  [ ON COMMIT { PRESERVE ROWS | DELETE ROWS } ]
  [ TABLESPACE tablespace ]
```

where *column_constraint* is:

```
  [ CONSTRAINT constraint_name ]
  { NOT NULL |
    NULL |
    UNIQUE [ USING INDEX TABLESPACE tablespace ] |
    PRIMARY KEY [ USING INDEX TABLESPACE tablespace ] |
    CHECK (expression) |
    REFERENCES reftable [ ( refcolumn ) ]
      [ ON DELETE action ] }
  [ DEFERRABLE | NOT DEFERRABLE ] [ INITIALLY DEFERRED |
    INITIALLY IMMEDIATE ]
```

and *table_constraint* is:

```
  [ CONSTRAINT constraint_name ]
  { UNIQUE ( column_name [, ...] )
      [ USING INDEX TABLESPACE tablespace ] |
    PRIMARY KEY ( column_name [, ...] )
      [ USING INDEX TABLESPACE tablespace ] |
    CHECK ( expression ) |
    FOREIGN KEY ( column_name [, ...] )
        REFERENCES reftable [ ( refcolumn [, ...] ) ]
      [ ON DELETE action ] }
  [ DEFERRABLE | NOT DEFERRABLE ]
  [ INITIALLY DEFERRED | INITIALLY IMMEDIATE ]
```

List Partitioning Syntax

Use the first form to create a list-partitioned table:

```
CREATE TABLE [ schema. ]table_name
   table_definition
   PARTITION BY LIST(column)
   [SUBPARTITION BY {RANGE|LIST} (column[, column ]...)]
   (list_partition_definition[, list_partition_definition]...);
```

Where *list_partition_definition* is:

```
      PARTITION [partition_name]
        VALUES (value[, value]...)
        [TABLESPACE tablespace_name]
        [(subpartition, ...)]
```

Range Partitioning Syntax

Use the second form to create a range-partitioned table:

```
CREATE TABLE [ schema. ]table_name
   table_definition
   PARTITION BY RANGE(column[, column ]...)
   [SUBPARTITION BY {RANGE|LIST} (column[, column ]...)]
   (range_partition_definition[, range_partition_definition]...);
```

Where *range_partition_definition* is:

```
      PARTITION [partition_name]
        VALUES LESS THAN (value[, value]...)
        [TABLESPACE tablespace_name]
        [(subpartition, ...)]
```

云数据库POLARDB-O - Oracle兼容的DML

```
SELECT [ optimizer_hint ] [ ALL | DISTINCT ]
  * | expression [ AS output_name ] [, ...]
  FROM from_item [, ...]
  [ WHERE condition ]
  [ [ START WITH start_expression ]
      CONNECT BY { PRIOR parent_expr = child_expr |
        child_expr = PRIOR parent_expr }
    [ ORDER SIBLINGS BY expression [ ASC | DESC ] [, ...] ] ]
  [ GROUP BY expression [, ...] [ LEVEL ] ]
  [ HAVING condition [, ...] ]
  [ { UNION [ ALL ] | INTERSECT | MINUS } select ]
  [ ORDER BY expression [ ASC | DESC ] [, ...] ]
  [ FOR UPDATE ]
```

where *from_item* can be one of:

```
  table_name[@dblink ] [ alias ]
  ( select ) alias
  from_item [ NATURAL ] join_type from_item
    [ ON join_condition | USING ( join_column [, ...] ) ]
```

```
SELECT /*+ ORDERED */ e.ename, d.dname, h.startdate
  FROM emp e, dept d, jobhist h
  WHERE d.deptno = e.deptno
  AND h.empno = e.empno;
```

Create an ascending sequence called `serial`, starting at 101:

```
CREATE SEQUENCE serial START WITH 101;
```

Select the next number from this sequence:

```
SELECT serial.NEXTVAL FROM DUAL;

 nextval
---------
     101
(1 row)
```

Create a sequence called `supplier_seq` with the `NOCACHE` option:

```
CREATE SEQUENCE supplier_seq
    MINVALUE 1
    START WITH 1
    INCREMENT BY 1
    NOCACHE;
```

```
SELECT empno, ename, job FROM emp WHERE ROWNUM < 5;

 empno | ename |   job
-------+-------+----------
  7369 | SMITH | CLERK
  7499 | ALLEN | SALESMAN
  7521 | WARD  | SALESMAN
  7566 | JONES | MANAGER
(4 rows)
```

云数据库POLARDB-O - Oracle 存储过程、函数、触发器

```
CREATE OR REPLACE PACKAGE emp admin
IS
    FUNCTION get dept name (
        p_deptno        NUMBER
    ) RETURN VARCHAR2;
    FUNCTION update emp sal (
        p empno         NUMBER,
        p_raise         NUMBER
    ) RETURN NUMBER;
    PROCEDURE hire emp (
        p empno         NUMBER,
        p_ename         VARCHAR2,
        p_job           VARCHAR2,
        p sal           NUMBER,
        p hiredate      DATE,
        p_comm          NUMBER,
        p_mgr           NUMBER,
        p deptno        NUMBER
    );
    PROCEDURE fire emp (
        p_empno         NUMBER
    );
END emp admin;
/
--
--  Package body for the 'emp_admin' package.
--
CREATE OR REPLACE PACKAGE BODY emp admin
IS
    --
    --  Function that queries the 'dept' table based on the department
    --  number and returns the corresponding department name.
    --
    FUNCTION get_dept_name (
        p_deptno        IN NUMBER
    ) RETURN VARCHAR2
    IS
        v_dname         VARCHAR2(14);
    BEGIN
        SELECT dname INTO v dname FROM dept WHERE deptno = p deptno;
        RETURN v dname;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            DBMS_OUTPUT.PUT LINE('Invalid department number ' || p deptno);
            RETURN '';
    END;
    --
```

```
CREATE OR REPLACE PROCEDURE list emp
IS
    v empno         NUMBER(4);
    v ename         VARCHAR2(10);
    CURSOR emp cur IS
        SELECT empno, ename FROM emp ORDER BY empno;
BEGIN
    OPEN emp_cur;
    DBMS_OUTPUT.PUT_LINE('EMPNO    ENAME');
    DBMS_OUTPUT.PUT_LINE('-----    -------');
    LOOP
        FETCH emp_cur INTO v_empno, v_ename;
        EXIT WHEN emp_cur%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(v_empno || '     ' || v_ename);
    END LOOP;
    CLOSE emp_cur;
END;
```

PL/SQL

```
CREATE OR REPLACE TRIGGER emp_sal_trig
    BEFORE DELETE OR INSERT OR UPDATE ON emp
    FOR EACH ROW
DECLARE
    sal_diff       NUMBER;
BEGIN
    IF INSERTING THEN
        DBMS OUTPUT.PUT LINE('Inserting employee ' || :NEW.empno);
        DBMS OUTPUT.PUT LINE('..New salary: ' || :NEW.sal);
    END IF;
    IF UPDATING THEN
        sal diff := :NEW.sal - :OLD.sal;
        DBMS OUTPUT.PUT LINE('Updating employee ' || :OLD.empno);
```

```
CREATE OR REPLACE FUNCTION new empno RETURN NUMBER
IS
    v cnt           INTEGER := 1;
    v new empno     NUMBER;
BEGIN
    WHILE v_cnt > 0 LOOP
        SELECT next_empno.nextval INTO v_new_empno FROM dual;
        SELECT COUNT(*) INTO v_cnt FROM emp WHERE empno = v_new_empno;
    END LOOP;
    RETURN v_new_empno;
END;
```

云数据库POLARDB-O - DBMS_*、ALL_*、DBA_*、USER_*

POLARDB 规格（带 Oracle 兼容性）

节点规格	CPU和内存数	最大存储容量	最大连接数	内网带宽	最大IOPS	I/O带宽
polar.o.x4.medium	2核 8GB	5,000 GB	400	1 Gbps	16,000	1 Gbps
polar.o.x4.large	4核 16GB	10,000 GB	800	10 Gbps	64,000	4 Gbps
polar.o.x4.xlarge	8核 32GB	10,000 GB	1600	10 Gbps	128,000	8 Gbps
polar.o.x8.xlarge	8核 64GB	20,000 GB	3200	10 Gbps	160,000	10 Gbps
polar.o.x8.2xlarge	16核 128GB	40,000 GB	6400	10 Gbps	256,000	16 Gbps
polar.o.x8.4xlarge	32核 256GB	60,000 GB	12800	10 Gbps	384,000	24 Gbps
polar.o.x8.12xlarge	88核 710GB	100,000 GB	36000	25 Gbps	512,000	32 Gbps

阿里中间件 Alibaba Middleware

阿里云

Advanced Database & Application Migration

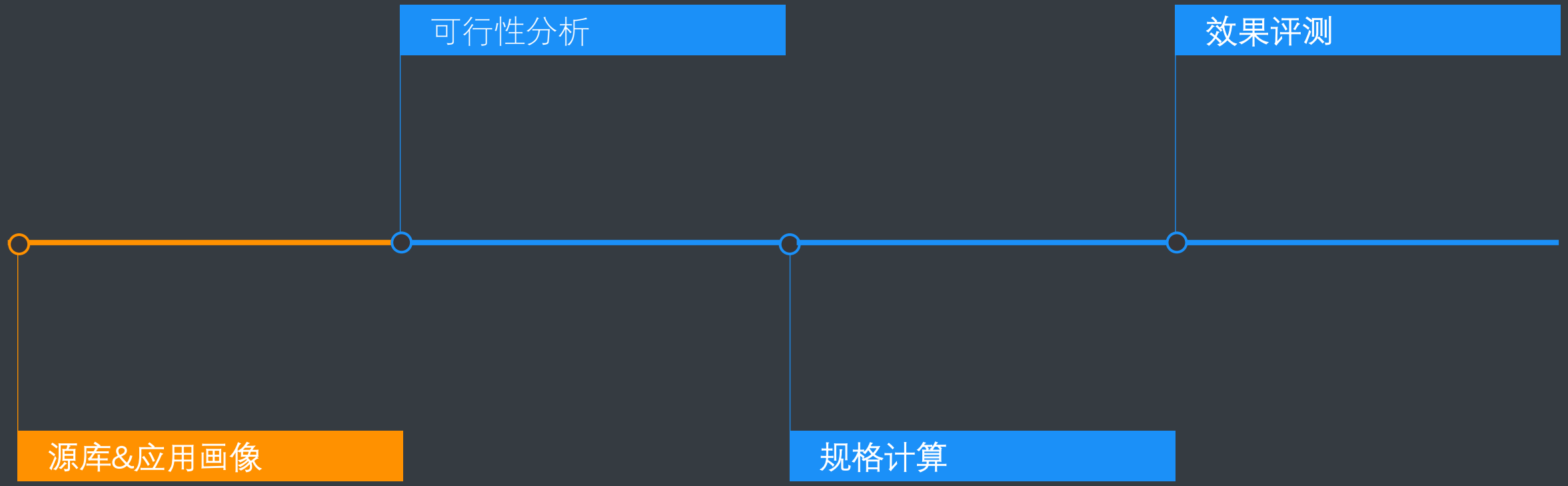
（简称 ADAM，亚当）

数据库和应用迁移（Advanced Database & Application Migration, 以下简称ADAM），是阿里云结合阿里巴巴多年内部业务系统数据库和应用异构迁移的经验（去IOE），自主研发的、迁移ORACLE数据库和应用至阿里云相关云产品的专业产品。

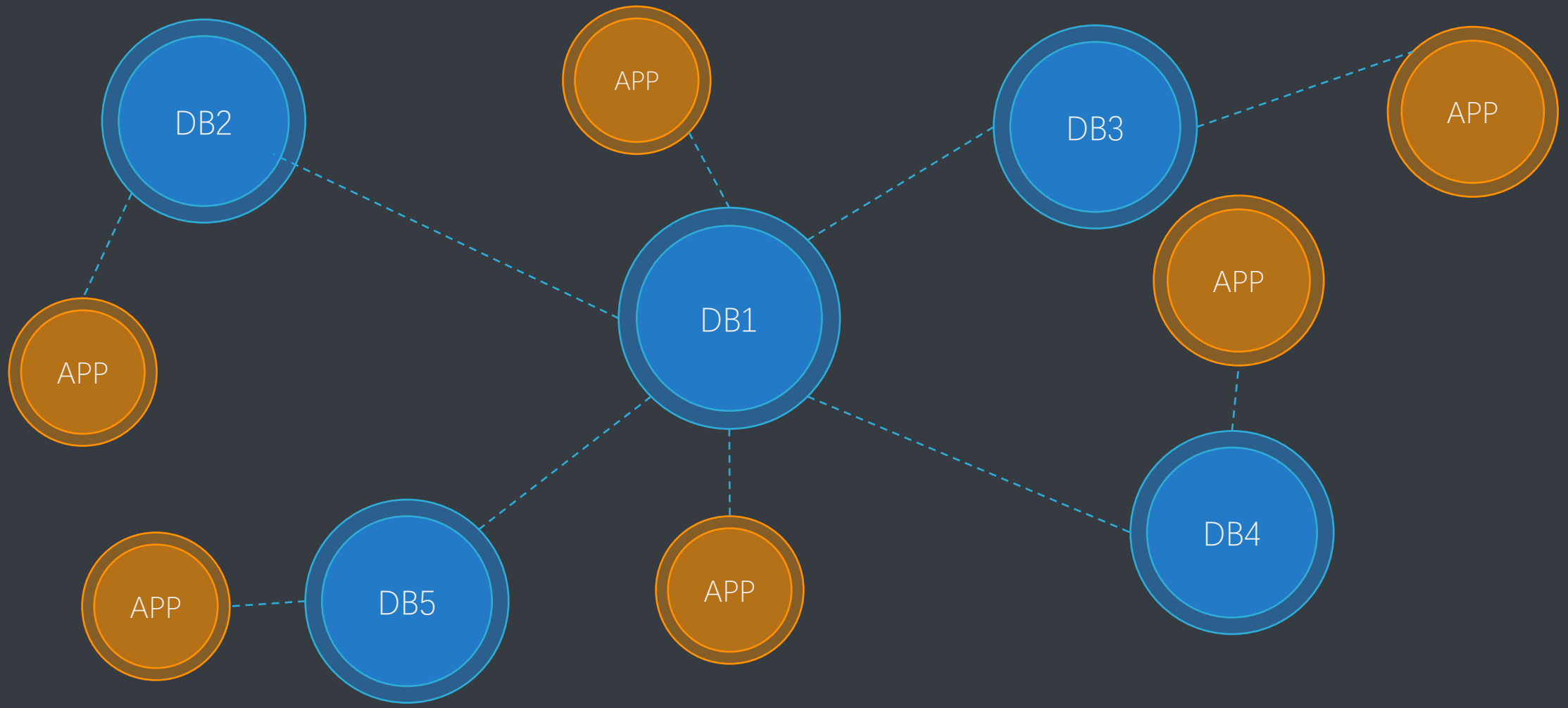
ADAM产品迁云流程


数据采集
智能分析
业务改造
迁移实施
系统优化
ADAM采集工具
ADAM SaaS
ADAM studio
ADAM stuido
ADAM SaaS+studio
ADAM数据库采集工具
ADAM应用采集工具
ADAM solution
report
业务代码改造
数据库测试数据准备
数据库和应用割接
ADAM SaaS studio
DBA

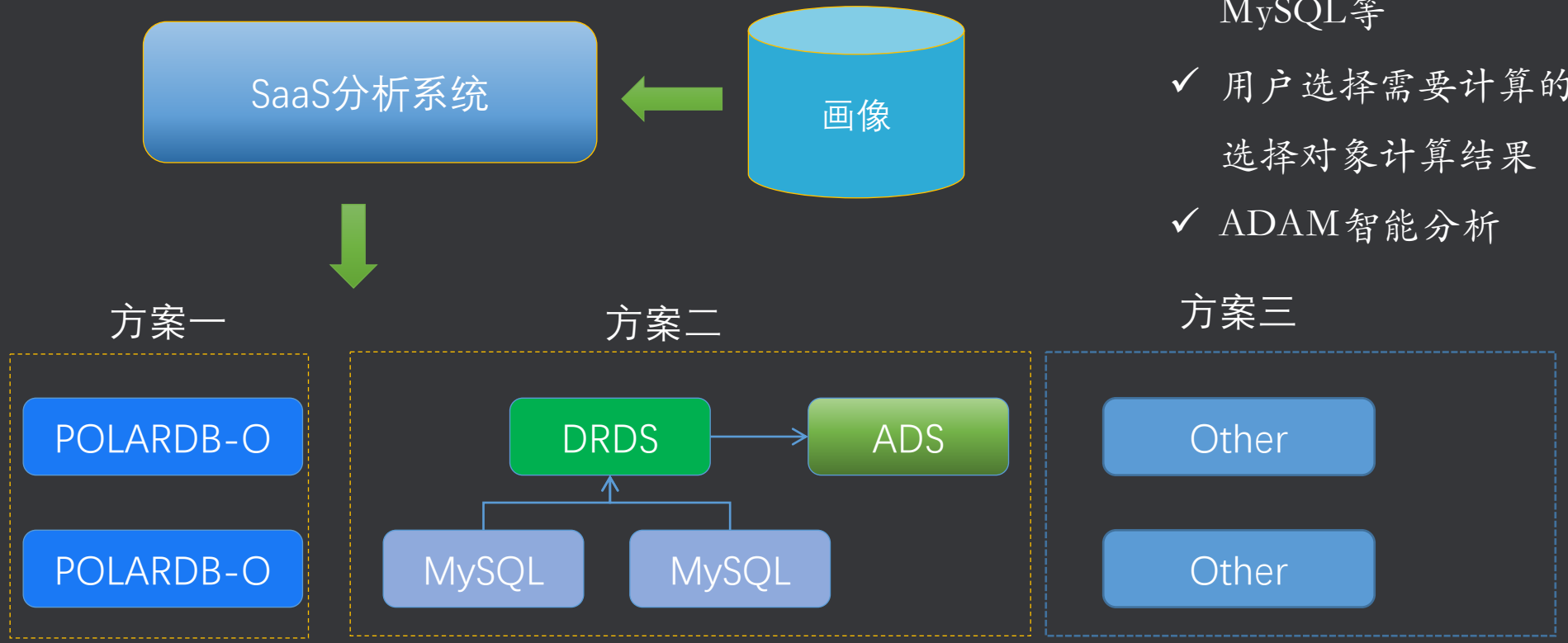
SaaS智能分析



画像关系



智能分析-数据库



可行性&规格计算

- ✓ 用户选择合适的目标数据库，例如PPAS, ADS + MySQL等
- ✓ 用户选择需要计算的Schema和表，方案仅根据选择对象计算结果
- ✓ ADAM智能分析

SQL、Table等兼容性详细分析

SQL详细评估报告

（企业版）

评估时间：2017-05-1016:00:31　负责人：cl

总计评估sql数量为1546，匹配到特性的sql有1175，其中，不兼容的sql数量有6，修改后兼容的sql数量为34，兼容的sql数量为1135。

需要修改SQL详细信息

编号	对象名称	兼容性	SQL	兼容性分析建议
1	JWZH.6hvo413 2yyass	ISNOT NULL 兼容; EXISTS兼容 ; SYSDATE 需要修改; SUBQUERY 兼容; SYS_GUID 需要修改;	insert into c_xsq (xid,userid, mbid, bssj, lxing, ydzt, cssj) (select sys_guid(),:1 ,i.*,sysdate,'1','0',sysdate+3 from ((select a.appid from t_role_app a where exists(select 1 from t_user_role r where a.appid is not null and r.type='u' and r.roleid=a.roleid and r.userid=:2) and not exists(select 1 from c_yyqx q where q.userid=:3 and a.appid=q.mbid and q.lx='1'))) i)	SYSDATE修改指引：写法不同：ORACLE:select SYSDATE value from dual MYSQL:select now() value select sysdate() value SYS_GUID修改指引：
2	JWZH.b4qdnc 4cbtsrd	DISTINCT 兼容; GROUP BY 兼容; HAVING 兼容; ROWNUM 需要修改;	SELECT * FROM (SELECT ROWNUM RN, res.* FROM (select * from (select (select usernumber from t_userinfo where id=userid) usernumber from (select distinct ur.userid,ra.appid from t_user_role ur, t_role_app ra where ur.roleid = ra.roleid and ur.type='u' and exists(select 1 from appresource a where ra.appid = a.appid and (a.tourl like :1 or a.tourl like :2))) group by userid having count(1)>1)) res WHERE ROWNUM <=:3) WHERE RN >=:4	ROWNUM修改指引：用于从查询返回的行的编号，返回的第一行分配的是1，第二行是2，依此类推，这个伪字段可以用于限制查询返回的总行数，而且rownum不能以任何表的名称作为前缀。Rownum常用于oracle的分页语句，MySQL使用limit

1，具体到每一个对象
- Table
- View
- SQL
- PLSQL
- …

2，给出修改建议

3，给出正确的SQL

阿里中间件 Alibaba Middleware　阿里云

SQL溯源分析与建议

AMSTool

即时报告　汇总报告

根据条件生成报告　更新即时报告　生成汇总报告

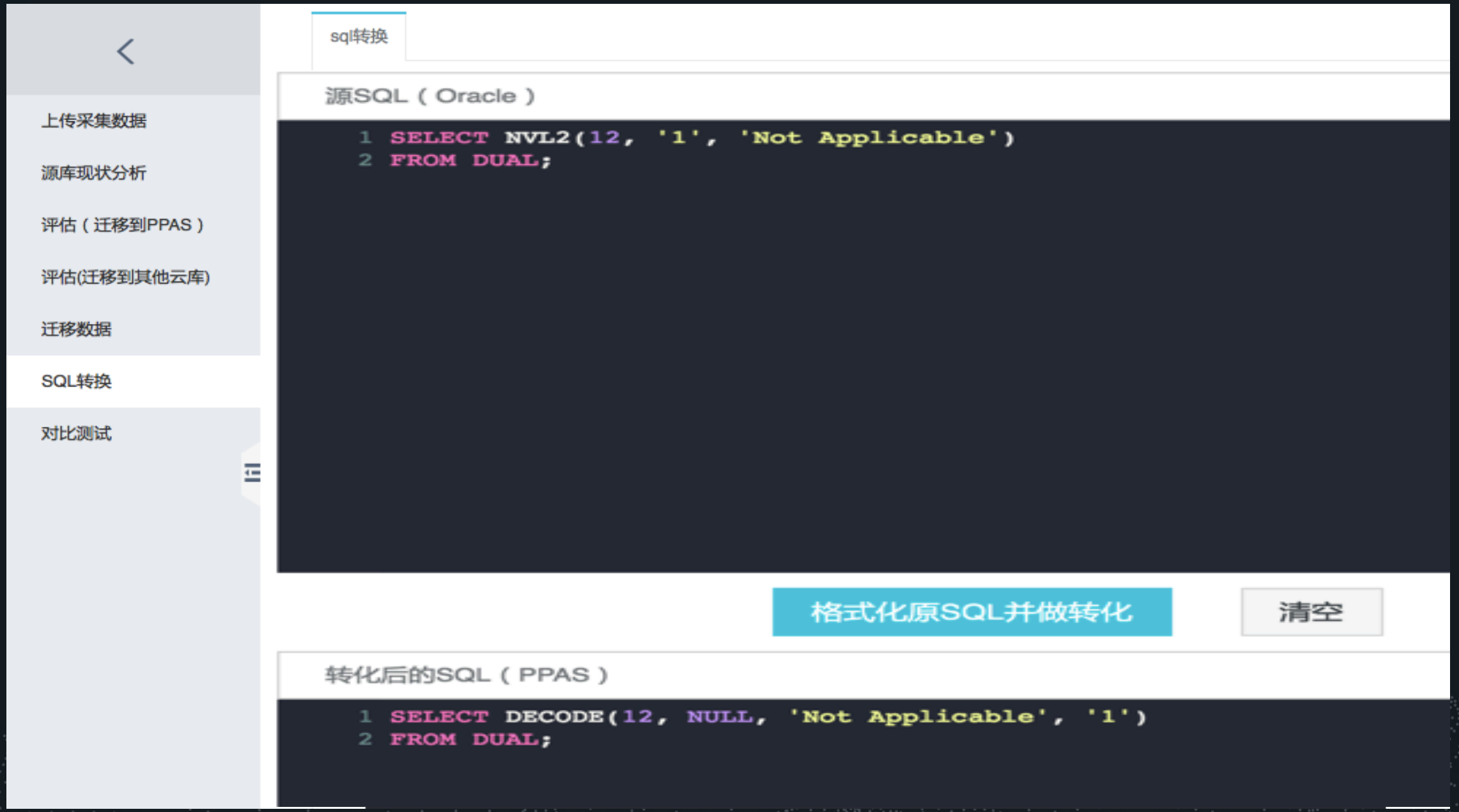
搜索过滤条件　开始时间: 未选择时间　结束时间: 未选择时间　SQL内容　请输入需要搜索的内容　历史记录

排序　执行时间　正序　倒序

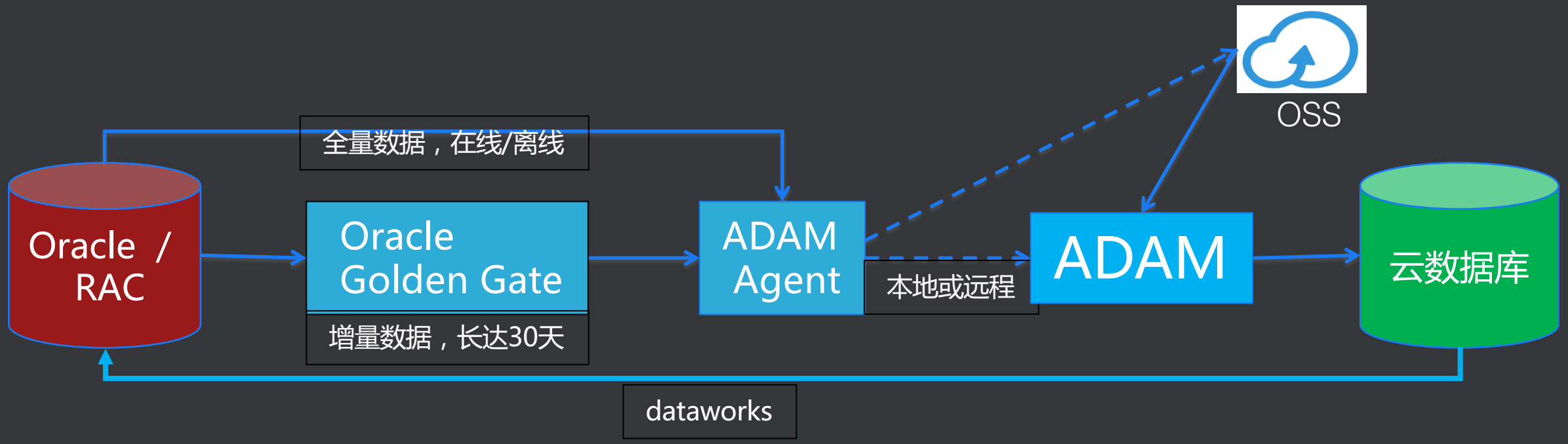
编号	时间标识	SQL内容	执行时间(毫秒)	执行结果	兼容性	场景分析	sql转换结果	调用栈
1	2017-12-15 13:15:24.06	select count(*) from adam_studio.project;	3.666854	执行成功			转换后的目标SQL：SELECT COUNT(*) FROM adam_studio.project;	org.mybatis.spring.SqlSessionTemplate$SqlSessionInterceptor.invoke(SqlSessionTemplate.java:433) com.sun.proxy.$Proxy53.selectList(Unknown Source) org.mybatis.spring.SqlSessionTemplate.selectList(SqlSessionTemplate.java:230) com.sun.proxy.$Proxy57.testAmsSql(Unknown Source)
2	2017-12-15 13:16:02.728	SELECT NVL2(12, '1', 'Not Applicable') FROM DUAL;	3.134	执行成功	NVL2 不兼容		转换后的目标SQL：SELECT DECODE(12, NULL, 'Not Applicable', '1') FROM DUAL;	org.mybatis.spring.SqlSessionTemplate$SqlSessionInterceptor.invoke(SqlSessionTemplate.java:433) com.sun.proxy.$Proxy53.selectList(Unknown Source) org.mybatis.spring.SqlSessionTemplate.selectList(SqlSessionTemplate.java:230) com.sun.proxy.$Proxy57.testAmsSql(Unknown Source)
3	2017-12-15 13:17:35.938	SELECT ams.getAllCount() from dual;	3.874919	执行成功			转换后的目标SQL：SELECT ams.getAllCount() FROM dual;	org.mybatis.spring.SqlSessionTemplate$SqlSessionInterceptor.invoke(SqlSessionTemplate.java:433) com.sun.proxy.$Proxy53.selectList(Unknown Source) org.mybatis.spring.SqlSessionTemplate.selectList(SqlSessionTemplate.java:230) com.sun.proxy.$Proxy57.testAmsSql(Unknown Source)
4	2017-12-15 13:22:37.902	SELECT BITAND(22.2, 111) FROM DUAL;	5.984475	执行成功	BITAND 不兼容		转换后的目标SQL：SELECT CAST(22.2 AS SMALLINT) & CAST(111 AS SMALLINT)	org.mybatis.spring.SqlSessionTemplate$SqlSessionInterceptor.invoke(SqlSessionTemplate.java:433) com.sun.proxy.$Proxy53.selectList(Unknown Source) org.mybatis.spring.SqlSessionTemplate.selectList(SqlSessionTemplate.java:230) com.sun.proxy.$Proxy57.testAmsSql(Unknown Source)

阿里中间件 Alibaba Middleware

阿里云



ADAM studio数据库割接

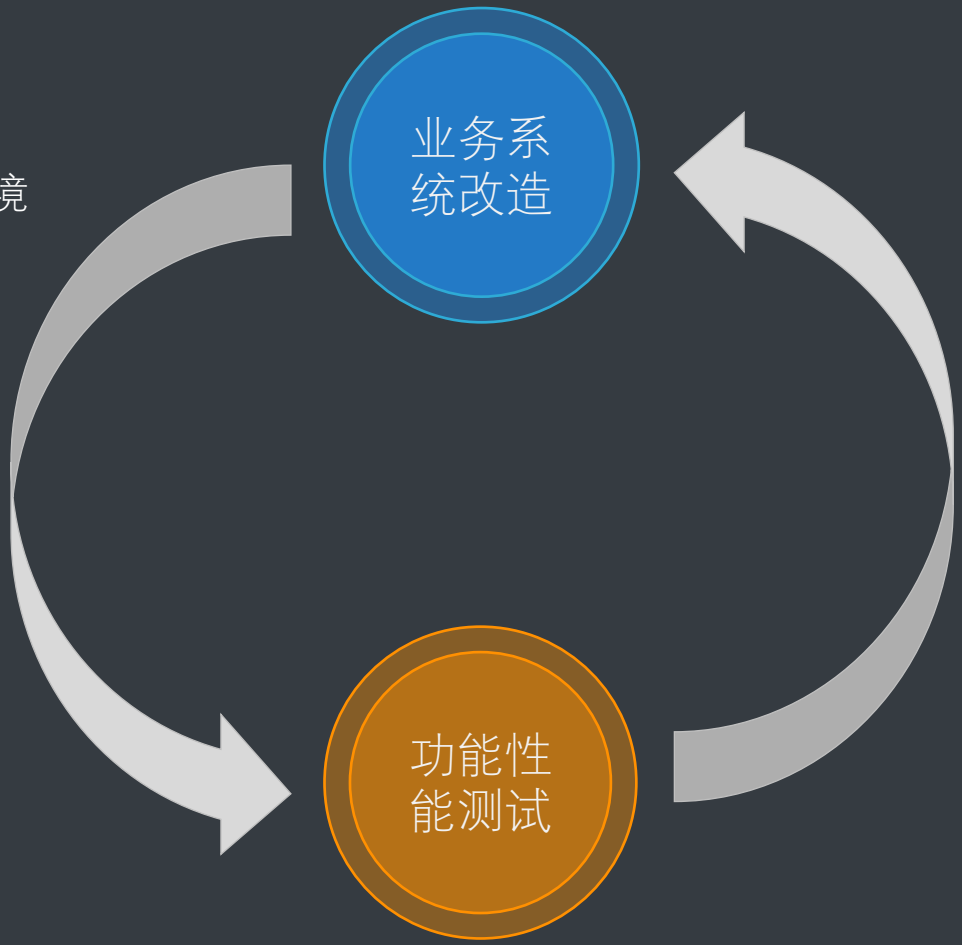


业务改造-测试

业务系统改造

使用ADAM studio构建数据库测试环境

基于测试环境改造业务系统



功能测试

基于新数据库不断测试业务系统功能、性能

上传采集数据
源库现状分析
评估（迁移到PPAS）
评估(迁移到其他云库)
迁移数据
SQL转换
对比测试

项目名：寒玄测试

指引文件下载 | 对测结果展示

原SQL内容 | 请输入需要搜索的内容 | 开始时间 请选择日期 | 结束时间 请选择日期 | 搜索 | 上传数据

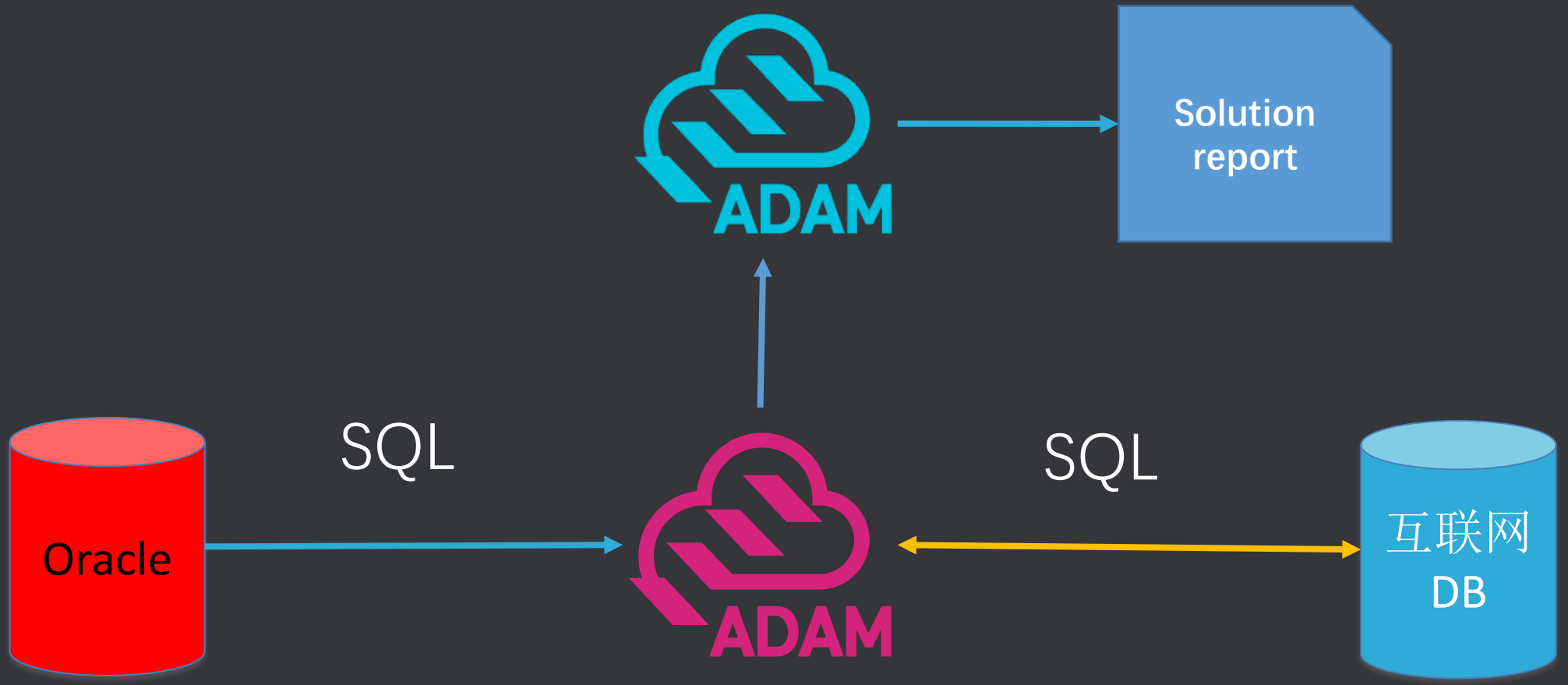
SQL产生时间	原SQL内容	原SQL执行结果	原SQL执行时间（毫秒）	转化后SQL内容	转化后SQL执行结果	转化后SQL执行时间(毫秒)	结果对比详情
	AND NOT EXISTS (SELECT UK2.VIR...查看详情			AND NOT EXISTS (SELECT UK2.VIR...查看详情			27
2017-11-20 20:04:18.0	SELECT SUM(COUNT(*)) FROM "OGG"."GGS_TEMP_UK" UK1 WHERE SEQNO = ? AND NOT EXISTS (SELECT UK2.NUL...查看详情	执行成功	8.044	SELECT SUM(COUNT(*)) FROM "OGG"."GGS_TEMP_UK" UK1 WHERE SEQNO = ? AND NOT EXISTS (SELECT UK2.NUL...查看详情	执行报错	-1	ERROR: relation "ogg.ggs_temp_uk" does not exist 位置：27
2017-11-20 20:04:18.0	SELECT SNO, OBJ_NAME, OWNER_NAME, BASE_OBJ_NAME, BASE_OWNER_NAME, BASE_OBJ_PROPERTY, OBJ_TYPE, COMMA...查看详情	执行成功	13.7183	SELECT SNO, OBJ_NAME, OWNER_NAME, BASE_OBJ_NAME, BASE_OWNER_NAME, BASE_OBJ_PROPERTY, OBJ_TYPE, COMMA...查看详情	执行成功	38.4647	结果相同，校验无误
2017-11-27 10:45:02.0	SELECT VALUE FROM "OGG"."GGS_STICK" WHERE PROPERTY = ?	执行成功	14.4846	SELECT VALUE FROM "OGG"."GGS_STICK" WHERE PROPERTY = ?	执行报错	-1	ERROR: relation "ogg.ggs_stick" does not exist 位置：19
2017-11-27 10:45:02.0	SELECT VALUE FROM "OGG"."GGS_SETUP" WHERE PROPERTY = ?	执行成功	9.29961	SELECT VALUE FROM "OGG"."GGS_SETUP" WHERE PROPERTY = ?	执行成功	45.3598	行数不匹配
2017-11-28 17:53:39.0	SELECT MIN(KEYNAME) FROM "OGG"."GGS_TEMP_UK" UK1 WHERE SEQNO = ? AND NOT EXISTS (SELECT UK2.VIRT...查看详情	未执行	3932	SELECT MIN(UK1.KEYNAME) FROM OGG.GGS_TEMP_UK UK1 WHERE UK1.SEQNO = ? AND NOT EXISTS (SELECT UK2....查看详情	执行报错	-1	执行语句时报错

上一页 1 ... 21 22 23 24 下一页 22/24 到第 页 确定 导出 迁移风险确认报告

ADAM技术支持

咨询·建议

系统优化—风险SQL识别

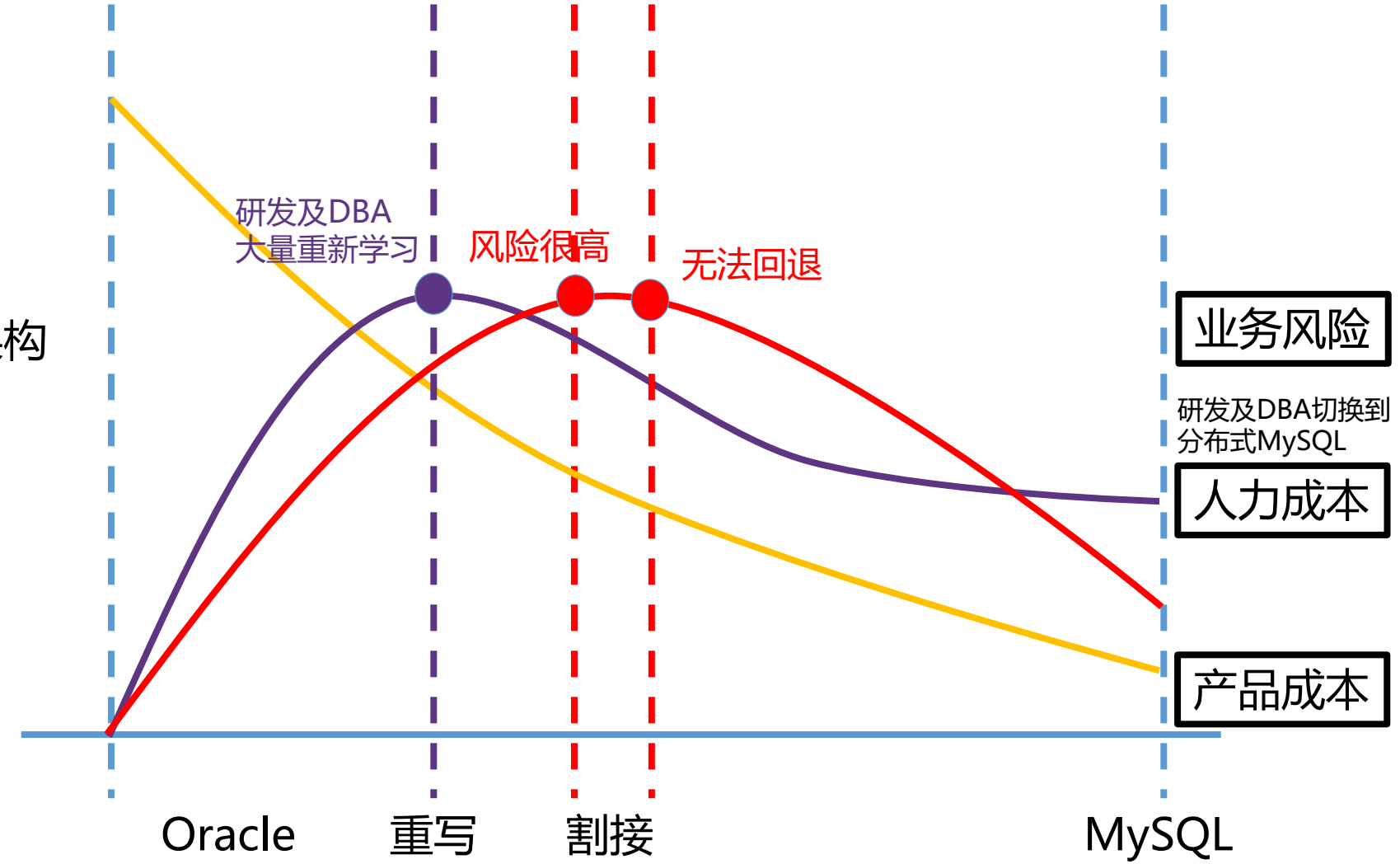


为何Oracle -> MySQL系列 迁移难以推动的原因

ISV及企业迁移风险高

- Oracle -> MySQL
  大改，代码 + 存储过程 + 架构

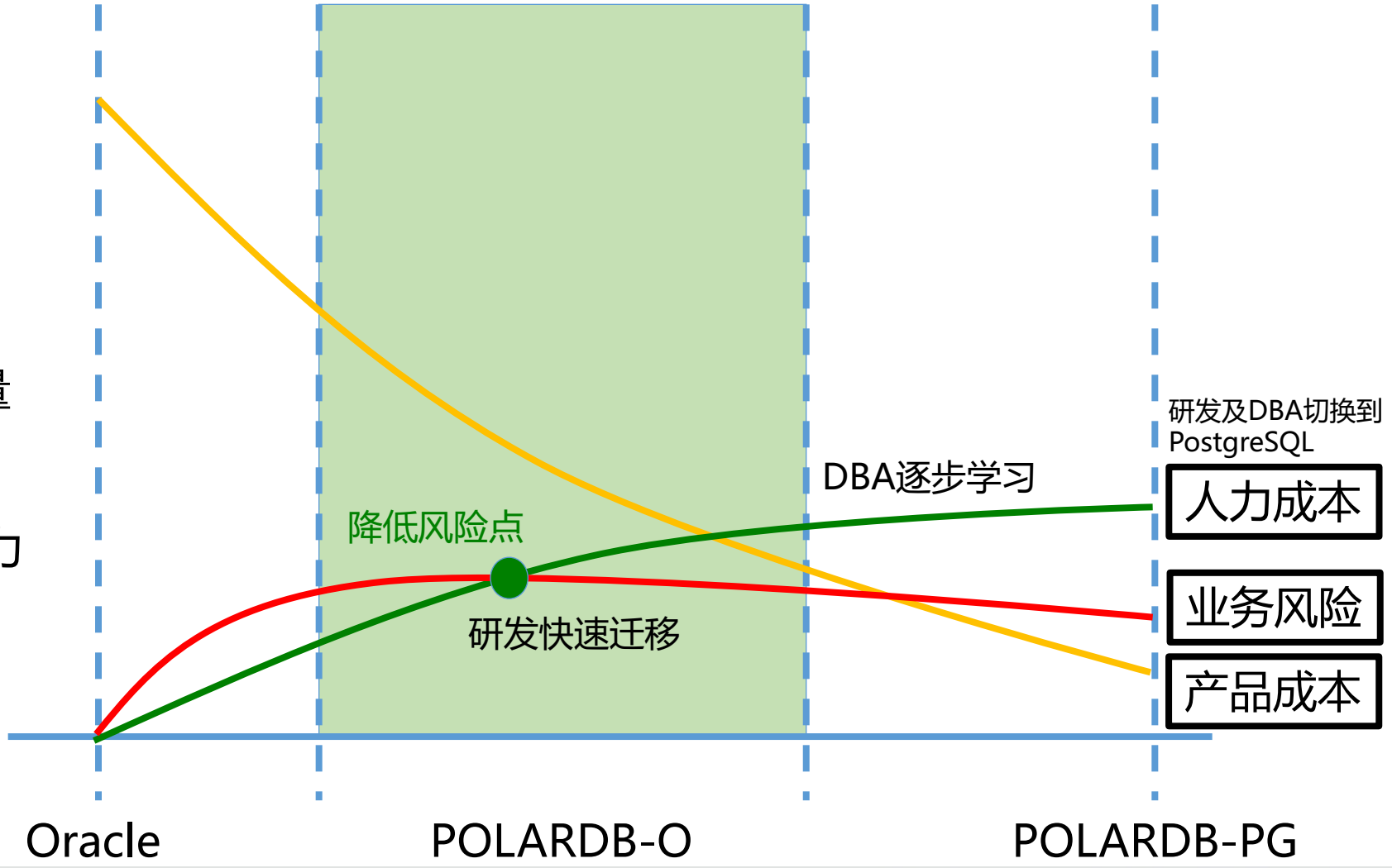
- 研发：重新学习
- DBA：重新学习
- 代码：话法重写
  甚至业务架构重写



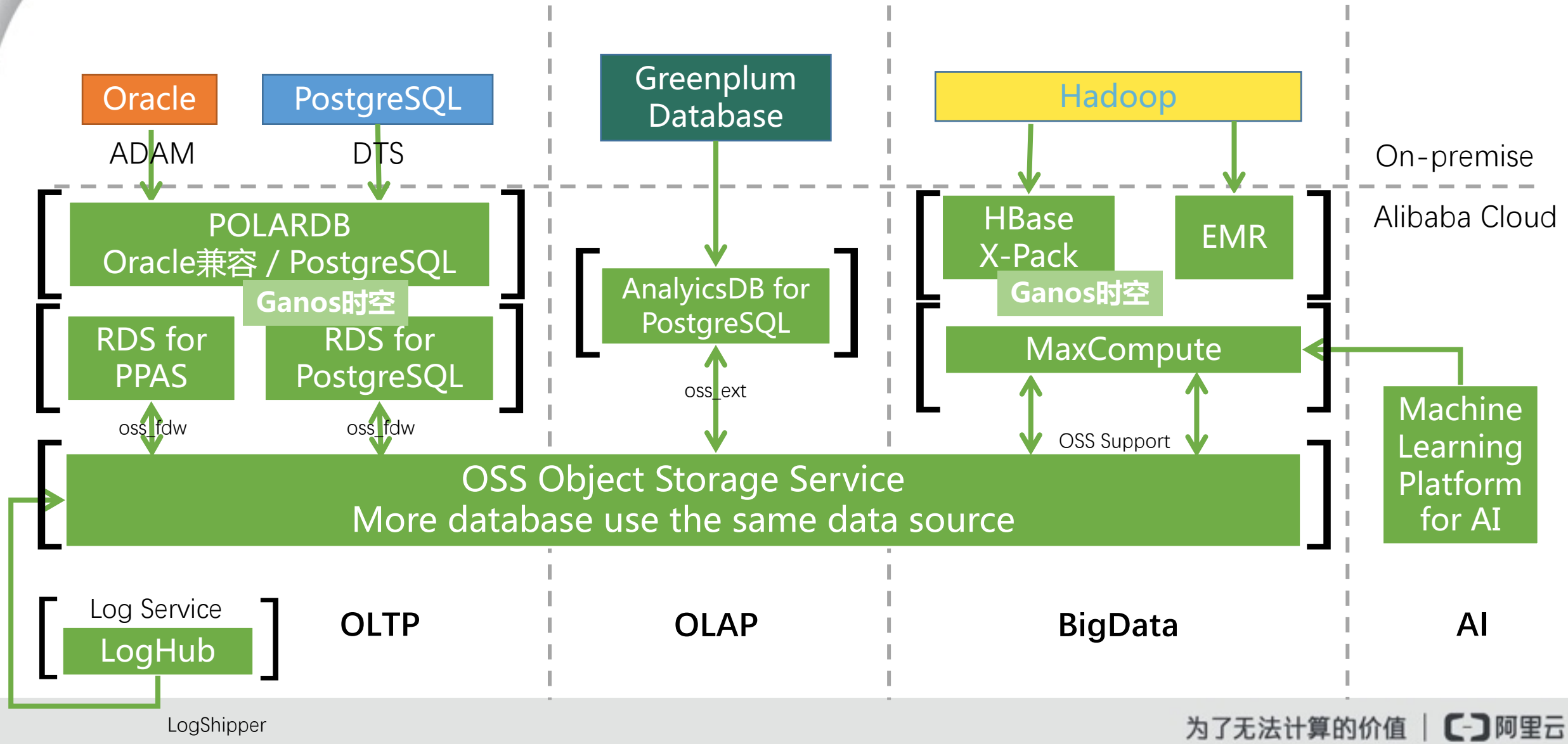
POLARDB + ADAM - 提高Oracle迁移上云成功率

- Oracle -> POLARDB
  语法几乎不改

- 研发：可继续写Oracle语法
  降低迁移难度及工作量
- DBA：阿里云自动运维
  提高DBA SQL优化能力
- 代码：几乎不用修改
  ADAM协助精准分析



Postgres Ecosystem: OLTP -> OLAP -> BigData -> AI



OSS 如何配合 Postgres, EMR, LogHub 及 MaxCompute 工作

在RDS for PostgreSQL中使用OSS:

- https://help.aliyun.com/document_detail/44461.htm

在HybridDB for PostgreSQL中使用OSS:

- https://help.aliyun.com/document_detail/35457.htm

在EMR中使用OSS:

- https://help.aliyun.com/document_detail/42799.html

LogHub数据导入OSS:

- https://help.aliyun.com/document_detail/29002.html

MaxCompute通过OSS导入:

- https://yq.aliyun.com/articles/110948

为了无法计算的价值 | 阿里云