

文章编号: 1005-8451 (2019) 2-0016-06

基于TimesTen内存数据库提升股道现车同步应用性能的研究与测试

王凤琳

(北京经纬信息技术有限公司, 北京 100081)

摘要: 介绍铁路运输信息集成平台股道现车同步应用的现状和存在问题, 针对目前Oracle数据库环境下应用处理的性能, 使用TimesTen内存数据库进行了性能和高可用测试。测试结果表明, TimesTen可大幅提升应用处理效率, 其高可用性保证了业务处理的连续性和数据的完整性, 且运维人员仅需对既有应用进行部分适应性修改就可实现Oracle数据库到TimesTen内存数据库的迁移, 为全面提升股道现车同步应用性能提供了科学依据和有力支撑。

关键词: 股道现车同步; 内存数据库; 性能测试; 高可用性测试

中图分类号: U292 : TP39 **文献标识码:** A

Research and test of improving performance of station track inventory vehicle synchronization application based on TimesTen memory database

WANG Fenglin

(Beijing Jingwei Information Technology Co. Ltd., Beijing 100081, China)

Abstract: This article introduced the current status and problems of the station track inventory vehicle synchronization application on railway transport information integration platform. In view of the performance of application processing in Oracle database environment, TimesTen memory database was used to test its performance and high availability. The test results show that TimesTen can greatly improve the efficiency of application processing, and its high availability ensures the continuity of business processing and the integrity of data. Moreover, the migration of Oracle database to TimesTen memory database can be implemented only by modifying the existing applications partially, which provides a scientific basis and strong support for the overall improvement of the performance of station track inventory vehicle synchronous application.

Keywords: station track inventory vehicle synchronization; memory database; performance testing; high availability testing

铁路运输信息集成平台的目标是实现列车、车辆、货物、机车、机车乘务员等信息的数据集中与共享, 并实时掌握与动态追踪其位置和状态、推算与预测其变化趋势, 为实现精确调度指挥奠定基础, 同时为货运电子商务和货运组织改革提供必要的技术支撑。

运输信息集成平台建设过程中, 中国铁路总公司级(简称: 总公司级)实时掌握各车站股道现车情况, 以便精确组织运输生产, 是一项重要且迫切的需求。而铁路运输当前的实际情况是, 股道现车数据除本车站外, 仅由车站所在铁路局实时掌握, 铁

路总公司调度部仅能通过各个铁路局的现车系统查询相关信息, 无法从全路角度掌控所有在站车辆的位置、数量及空重状态, 故如何在总公司级实时掌握并管理全路约60万辆股道现车, 是运输信息集成平台建设中亟需解决的问题。

本文旨在分析当前股道现车同步应用存在的主要问题的基础上, 针对新的内存数据库技术进行了学习与研究, 并将其运用于股道现车同步应用测试, 为应用性能的提升提供了科学依据和有力支撑。

1 主要问题及解决思路

1.1 主要问题

中国铁路总公司(简称: 总公司)运输信息集

收稿日期: 2018-04-13

作者简介: 王凤琳, 工程师。

成平台股道现车同步应用运行在 Unix 操作系统, 使用 Oracle 数据库, 运用 Pro*C 编程语言, 实现车站股道现车数据同步至总公司。全路股道现车数据庞大且实时变化, 基于当前的设备能力和技术水平, 很难在总公司级做到真正意义上的同步, 即每一辆车的变化状态都实时反映在总公司运输信息集成平台中。目前采用的方式是: 利用 MQ 消息传输机制, 各铁路局每 30 min 按车站为单位以 XML 文件的形式上报局管内车站股道现车数据至总公司集成平台, 总公司接收文件并解析入数据库, 以数据共享方式向相关应用提供数据支持。

全路已实施车站系统的车站约 6 000 多个, 均需每 30 min 上报本站当前股道现车数据, 具体包括车号、到达时间、到达股道、空重、始发站、终到站、货物、车辆状态等信息^[1]; 对于当前时间没有任何车辆的车站, 仍需上报股道现车报告, 以便在总公司股道现车库中清除上一时间点的车辆信息。频繁的删除和插入操作, 使得总公司股道现车库压力较大, 执行效率较低。为此, 根据各铁路局报告数量, 经过估算测试, 优化为 6 个进程并行处理, 满足 30 min 同步的要求, 每处理 30 min 数据需大约 12 min 左右, 即总公司数据与车站实际情况之间存在约 40 min 的时间差。时间差使得总公司与铁路局数据存在较大的不一致性, 而提高两级数据一致性的首要问题是缩短数据同步时间间隔, 提升数据处理能力。

1.2 解决思路

内存数据库技术具有更快的数据读取速度, 能够大大提升数据处理效率, 且该项技术在铁路互联网余票查询^[2]、铁路货车追踪^[3]等应用中已取得一定成效。经过对目前主流的各类内存数据库产品进行对比和分析, 基于与既有应用最大的兼容性, 总公司运输信息集成平台股道现车同步应用选取了 Oracle TimesTen 内存数据库, 进行了数据处理性能和高可用性方面的研究与测试。

2 Oracle TimesTen内存数据库简介

Oracle 内存数据库 TimesTen 是一款针对内存进行了优化的关系型数据库, 它为应用程序提供了即

时响应性和非常高的吞吐量。Oracle TimesTen 作为高速缓存或嵌入式数据库部署在应用程序中, 利用标准 SQL 接口对完全位于物理内存中的数据存储进行操作, 对于大规模的查询应用非常有好处^[4]。

目前, Oracle TimesTen 在全球的客户包括 Dell^[5]、美国美林银行^[6]、上海海关^[7]等, 拥有如此众多的客户群并得到业界的广泛认可, 主要得益于以下几项优势^[8]:

- (1) 能够和 Oracle 数据库做无缝集成, 数据可以在 TimesTen 和 Oracle 直接实现实时双向流动;
- (2) TimesTen 可以做成多节点并行提供服务的模式, 数据在多个 TimesTen 之间直接实现实时或非实时的传输, 进一步提高了系统的扩展性和可靠性;
- (3) TimesTen 是符合 RDBMS 标准的独立内存数据库服务, 支持 SQL92, 可运行在 Linux、Windows、AIX 等操作系统平台上, 使用 Java、.NET、C、C++、Pro*C 等语言进行应用开发。

基于以上几点, 并结合总公司运输信息集成平台股道现车同步应用的特点, 运行于 Unix 平台、使用 Oracle 数据库、应用 Pro*C 语言、业务逻辑相对简单但要求快速响应, Oracle TimesTen 内存数据库是合适的选择。

3 Oracle TimesTen测试过程

3.1 测试目的

本次测试的主要目的是验证固定场景下内存数据库在数据处理上的效率, 为评估内存数据库技术在运输信息集成平台的使用提供依据。

3.2 测试环境及步骤

3.2.1 软硬件环境配置

测试环境的软硬件配置情况如表 1 所示, 两台主机分别安装 TimesTen11, 建立 Active Standby Pair 主从复制关系, 使用 Oracle Cluster Ware 作为集群管理软件, 监控 TimesTen 及错误切换、监控应用和文件系统及应用和文件系统切换。系统逻辑架构如图 1 所示。测试文件位于 /u01/testdata 目录, TimesTen 软件安装于 /home/oracle/TimesTen/tt1122 目录, 测试程序和脚本位于 /home/oracle/poc 目录。

3.2.2 测试数据说明

表1 软硬件环境配置

主机名	dp29-08	dp29-07	N/A
IP地址	10.1.242.140	10.1.242.139	10.1.227.18
型号	HP BL460c G6	HP BL460c G6	SUN主机
操作系统	RHEL 6	RHEL 6	Solaris
CPU	2CPU 16核, Intel(R) Xeon(R) CPU X5550 @ 2.67 GHz	2CPU 16核, Intel(R) Xeon(R) CPU X5550 @ 2.67 GHz	3 GHz
内存	126 G	126 G	32 G
硬盘	300 GB	300 GB	
用户	Oracle	Oracle	
数据库软件	TimesTen11.2.2.8 Linux x86_64版	TimesTen11.2.2.8 Linux x86_64版	Oracle Enterprise Edition

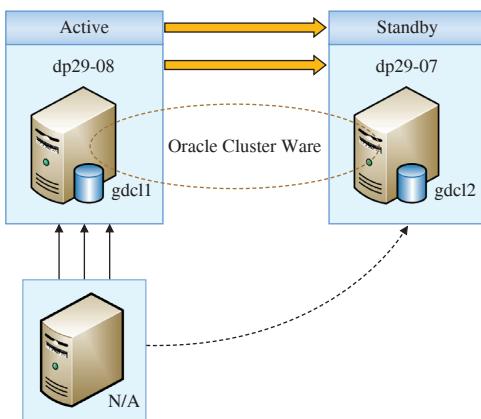


图1 系统逻辑架构

测试所用数据是实际业务数据文件的备份,为 XML 格式文件,描述各铁路局上报的车站股道现车情况,一个车站对应一个或多个文件,没有现车的车站上报一个只有文件头没有文件内容的空文件。为便于测试,将备份文件整理为数据集合 A 和数据集合 B:数据集 A 是从随机备份的实际业务数据的全集中经过筛选后,模拟全路各站每 30 min 上报的一轮数据的集合,并比照目前股道现车同步应用的实际情况,按铁路局分 6 个目录保存,分别为 gdcl_1、gdcl_2、……、gdcl_6;数据集 B 是在 A 的基础上,按铁路局将文件分为 12 个目录保存,分别为 gdcl_1、gdcl_2、……、gdcl_12。同时,为了保证测试的准确性,以股道现车实际生产数据表 BGDCARA 中的数据作为测试基础数据。

测试数据集 A 和 B 中文件数量和占用空间大小描述如表 2 所示。铁路局上报的股道车辆文件格式和股道车辆数据表结构,如图 2 和图 3 所示。

3.2.3 测试步骤

表2 测试数据集基本情况描述

测试数据集A			测试数据集B					
目录	文件数	空间(M)	目录	文件数	空间(M)	目录	文件数	空间(M)
gdcl_1	1 224	56	gdcl_1	483	31	gdcl_7	771	28
gdcl_2	1 273	62	gdcl_2	560	34	gdcl_8	720	31
gdcl_3	912	73	gdcl_3	656	39	gdcl_9	586	42
gdcl_4	1 327	66	gdcl_4	631	33	gdcl_10	696	34
gdcl_5	1 024	64	gdcl_5	532	24	gdcl_11	491	24
gdcl_6	1 140	63	gdcl_6	352	23	gdcl_12	422	41
合计	6 900	384				合计	6 900	384

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<REPORT>
  <BASE>
    <BGBS>报告标识</BGBS>
    <BGZM>报告点代码</BGZM>
    <BGSJ>报告时间</BGSJ>
    <BWMC>报文名</BWMC>
    <SJBS>更正报文事件标识</SJBS>
    <CZBZ>操作标志</CZBZ>
    <BGBB>报告版本号</BGBB>
    <BWCD>报文长度</BWCD>
  </BASE>
  <GDCL>
    <ITEM>
      <ZMLM>车站电报码</ZMLM>
      <GDM>股道码</GDM>
      <CID>车辆标识</CID>
      ...
    </ITEM>
  </GDCL>
</REPORT>

```

图2 股道现车文件格式

Name	Type
RPT_ID	CHAR(4)
RPT_VERSION	VARCHAR2(3)
IN_DB_DATE	CHAR(10)
IN_DB_TIME	CHAR(8)
RPT_NAME	VARCHAR2(14)
RPT_STN_CODE	VARCHAR2(3)
.....	

图3 股道现车数据表结构

- (1) 将 Oracle 数据库中基础数据表的数据导入到内存数据库表 BGDCARA 中。
- (2) 应用按时间序分别读取各文件夹中的报文,进行 XML 报文解析,并更新内存数据库表 BGDCARA。处理过程需保留日志备查。
- (3) 处理某一车站报告时,需删除数据库表中该车站对应的所有车辆,解析文件中每一辆车的详细信息,插入到内存数据库表 BGDCARA 中。
- (4) 正确处理完毕的文件直接删除;如果文件校验有错,将该文件保存到错误文件目录。

(5) 测试完毕, 将内存数据库数据表 BGDCARA 的数据导出到 Oracle 数据库中的数据表。

3.3 测试场景及结果

3.3.1 性能测试

性能测试是为了验证系统是否能够达到用户提出的性能指标, 同时发现系统中存在的性能瓶颈, 起到优化系统的目的^[9]。本测试仅涉及 dp29-08 一台主机, 基于数据集 A 和 B, 以 OCI 和 Java 两种方式分别进行性能测试, 以验证 Oracle TimesTen 数据处理能力。

(1) 测试准备

a. 连接 TimesTen 数据库 gdcl1, 比照集成平台生产环境中股道车辆 Oracle 数据表结构和索引在 TimesTen 数据库中新建数据表 BGDCARA。

b. 将 Oracle 数据库中的数据导入 TimesTen, 记录表行数。数据成功通过 TimesTen 缓存特性从 Oracle 数据库导入, 记录数为 626 610。

c. 在 /u01/testdata 目录下新建子目录 A 和 B, 解压测试数据集 A 到 /u01/testdata/A 目录下, 解压测试数据集 B 到 /u01/testdata/B 目录下。

(2) OCI 测试

a. A 数据集测试。比照既有生产环境下的模式, 启动 6 个 Pro*C 进程分别同时处理数据集 6 个文件夹中的数据, 以最后完成的进程运行时间作为总执行时间。为了保证测试的准确性, 共进行 10 轮测试, 记录每次运行时间, 并计算一轮平均运行时间为 16 s。登录 TimesTen gdcl1 数据库, 查看数据表 BGDCARA 中的数据及记录数。数据表总记录数为 620 356, 各字段显示正确。/u01/testdata/A 目录下正确处理完毕的 XML 文件成功删除, 解析校验不正确的文件共 4 个, 被挪至 /u01/testdata/errdata 目录下。

b. B 数据集测试。与既有生产环境下的模式不同, 测试启动 12 个 Pro*C 进程分别同时处理数据集 12 个文件夹中的数据, 以最后完成的进程运行时间作为总执行时间。为了保证测试的准确性, 共进行 10 轮测试, 记录每次运行时间, 并计算一轮平均运行时间为 18 s。登录 TimesTen gdcl1 数据库, 查看数据表 BGDCARA 中的数据及记录数。数据表总

记录数为 620 356, 各字段显示正确。/u01/testdata/B 目录下正确处理完毕的 XML 文件成功删除, 解析校验不正确的文件共 4 个, 被挪至 /u01/testdata/errdata 目录下。

(3) Java 测试

a. A 数据集测试。比照既有生产环境下的模式, 启动 6 个 Java 进程分别同时处理数据集 6 个文件夹中的数据, 以最后完成的进程运行时间作为总执行时间。为了保证测试的准确性, 共进行 10 轮测试, 记录每次运行时间, 并计算一轮平均运行时间为 22 s。登录 TimesTen gdcl1 数据库, 查看数据表 BGDCARA 中的数据及记录数。数据表总记录数为 620 356, 各字段显示正确。/u01/testdata/A 目录下正确处理完毕的 XML 文件成功删除, 解析校验不正确的文件共 4 个, 被挪至 /u01/testdata/errdata 目录下。

b. B 数据集测试。与既有生产环境下的模式不同, 测试启动 12 个 Java 进程分别同时处理数据集 12 个文件夹中的数据, 以最后完成的进程的运行时间作为总执行时间。为了保证测试的准确性, 共进行 10 轮测试, 记录每次运行时间, 并计算一轮平均运行时间为 24 s。登录 TimesTen gdcl1 数据库, 查看数据表 BGDCARA 中的数据及记录数。数据表总记录数为 620 356, 各字段显示正确。/u01/testdata/B 目录下正确处理完毕的 XML 文件成功删除, 解析校验不正确的文件共 4 个, 被挪至 /u01/testdata/errdata 目录下。

(4) 测试结果

性能测试结果如表 3 和表 4 所示。表 3 是测试过程中执行 DELETE 和 INSERT 操作的次数, 表 4 汇总了各类测试方式的处理效率。从两个表中可以看出, OCI 方式数据处理效率明显高于 Java 方式; 频繁的 DELETE 与 INSERT 操作并未对数据处理效率有明显影响; 当并行进程数由 6 调整到 12 时, OCI 与 Java 方式的处理时间均有所增加, 这说明并行度也有一个平衡点, 并非并发度越高越好。

3.3.2 高可用性测试

本文高可用性测试采用主从方式, 即主机工作, 备机处于监控准备状况, 当主机宕机时, 备机接管主机的一切工作, 待主机恢复正常后, 按使用者的设

表3 数据表操作统计表

处理文件数量	DELETE操作 执行次数	INSERT操作 执行次数	BGDCARA 记录行数
6 900	6 900	620 356	620 356

表4 处理效率汇总表

数据集	测试方式	正确报文数	错误报文数	处理时间 (s)	每千报文平均处理 时间 (s)
A	OCI	6 896	4	16	2.32
	JAVA	6 896	4	22	3.19
B	OCI	6 896	4	18	2.61
	JAVA	6 896	4	24	3.45

定以自动或手动方式将服务切换到主机上运行, 数据的一致性通过共享存储系统解决^[10]。

本测试涉及 dp29-08 和 dp29-07 两台主机, 数据库为 gdcl1 (dp29-08) 和 gdcl2 (dp29-07), 目的是为验证 TimesTen 对业务连续性的支持。

(1) 测试准备

a. 清空 dp29-08 主机 gdcl1 数据库 BGDCARA 数据表中的测试数据; 在主机 dp29-07 gdcl2 数据库中新建数据表 BGDCARA。

b. 从 Oracle 数据库导入数据到 gdcl1, 记录数为 626 610, 登录 gdcl2 可以看到相同记录数, 确认两个数据库之间数据复制正常。

c. 为便于测试, 将 6 个文件夹中的数据汇总到一个文件夹中, 以 OIC 方式启动一个应用进程处理数据。

(2) 高可用性测试之模拟内存数据库故障

a. 启动应用, 默认连接位于主机 dp29-08 的数据库 gdcl1, 进程运行正常, 实时往数据表中写入记录。模拟内存数据库故障, 手动 Kill 数据库守护进程, 应用立即检测到数据库错误, 并成功切换至 gdcl2 数据库继续处理文件、插入数据。

b. 文件全部处理完成后, 检查两数据库表中记录数。gdcl2 数据库表中记录数为 620 356, gdcl1 数据库表中记录数为 379 703, 为数据库 gdcl1 故障之前应用写入数据表的记录数。

c. 重新启动数据库 gdcl1, 复制关系自动建立, 最终 gdcl1 中的数据由 379 703 变为 620 356, 两数据库表数据完全一致。

(3) 高可用性测试之模拟主机故障

a. 启动应用, 默认连接位于主机 dp29-08 的数据库 gdcl1, 进程运行正常, 实时往数据表中写入记录。模拟主机 dp29-08 故障, reboot 重启主机, 应用停顿约几十秒后检测到错误, 并成功切换到 gdcl2 数据库继续处理文件、插入数据。

b. 文件全部处理完成后, 检查两数据库表中记录数。gdcl2 数据库表中记录数为 620 356, gdcl1 数据库表中记录数为 268 915, 为主机 dp29-08 重启之前应用写入数据表的记录数。

c. 主机 dp29-08 重启后, 启动数据库 gdcl1, 复制关系自动建立, 最终 gdcl1 中的数据由 268 915 变为 620 356, 两数据库表数据完全一致。

(4) 测试结果

从表 5 汇总情况可知, 无论数据库故障还是主机故障, 应用均无中断, 都可无缝切换到备机数据库, 继续处理数据, 并在故障恢复后, 主备机数据库可自动恢复复制关系, 保证主备机数据库表数据的完整性和一致性。该测试证明了 Oracle TimesTen 具有很好的高可用性, 为业务处理的连续性提供了有力的保证。

表5 高可用性测试结果汇总表

测试方式	应用情况监测	数据情况核查
内存数据库故障	应用立即检测到数据库错误, 并自动切换至备机数据库	数据库故障解除后, 两主机间数据库复制关系自动建立, 最终两数据库表数据完全一致。
主机故障	应用停顿约几十秒待连接超时后检测到错误, 并自动切换至备机数据库	故障主机成功重启并启动数据库后, 两主机间数据库复制关系自动建立, 最终两数据库表数据完全一致。

4 结语

通过本次测试, 可以看到使用 Oracle TimesTen 内存数据库进行铁路局股道现车数据同步的处理效率比 Oracle 数据库本身有了质的飞跃, 处理一轮数据所需时间从 12 min 缩短到 16 s, 提升了约 45 倍, 完全能够实现在总公司实时掌握车站车辆的现场情况。TimesTen 的高可用性, 更有力保证了业务处理的连续性和数据的完整性。同时, 由于 TimesTen 支持 SQL92 并支持 Pro*C 语言进行应用开发, 使得本次 OCI 测试应用程序修改量不大, 仅做了部分适应性修改, 程序主体基本沿用目前生产环境中的既有应用, 便于开发运维人员学习使用。

因此，使用 Oracle TimesTen 内存数据库提升股道现车同步应用的性能是可行的解决方案。

参考文献：

- [1] 中国铁路总公司. 铁路运输信息集成平台 1.0 (铁路局级) 暂行技术规范 : 铁总运 [2014]294 号 [S]. 北京 : 中国铁路总公司, 2014.
- [2] 梅巧玲, 王明哲, 张志强, 等. 内存数据库在互联网余票查询中的应用 [J]. 铁路计算机应用, 2014, 23 (3) : 41-44.
- [3] 颜昌盛, 范娟娟, 海洋, 等. 基于内存数据库提升铁路货车追踪应用性能的研究 [J]. 铁路计算机应用, 2015, 24 (6) : 26-30.
- [4] Jason Yang. TimesTen In-Memory Database FAQ[EB/OL]. [2017-10-24]. <https://blogs.oracle.com/database4cn/timesten-v1>.
- [5] Juan Garza. Dell TimesTen Use Case[EB/OL]. [2012-01-03]. <http://www.oracle.com/technetwork/database/database-technologies/timesten/overview/dell-customer-case-1446274>.

pdf?ssSource-SiteId=otncn.

- [6] Wayne Wilson. Oracle TimesTen use within Bank of America Merrill Lynch[EB/OL].[2010-09-23]. <http://www.oracle.com/technetwork/database/database-technologies/timesten/overview/boba-customer-case-1446280.pdf?ss-SourceSiteId=otncn>.
- [7] Oracle®. 上海海关构建自贸区海关监管信息系统平台, 借助内存数据库技术提升系统性能达 43 倍 [EB/OL]. [2015-05-22]. <https://www.oracle.com/cn/customers/shanghai-customs-5-db-cn.html>.
- [8] Sam Drake. What is TimesTen?[EB/OL]. [2011-05-06]. <https://blogs.oracle.com/timesten/what-is-timesten>.
- [9] 孟祥丰. 软件性能测试解析与优化 [J]. 电子设计工程, 2011, 19 (10) : 26-28.
- [10] 陈满砚. 高可用性软件测试的方法及应用 [D]. 武汉 : 华中科技大学, 2010.

责任编辑 陈蓉

(上接 P15)

踪间隔时间的计算值及变化规律, 为信号控制系统的选 择提供依据。在今后的工作中, 应更深入的研究列车牵引制动全过程, 同时考虑线路参数对模型的影响, 进一步细化模型, 使得结果更能贴近实际工程, 更具说服力。

参考文献：

- [1] 邢红霞. 城市轨道交通列车自动控制系统闭塞制式的分析 [J]. 铁路计算机应用, 2008, 17 (7) : 49-51.
- [2] 陈卫华, 刘江, 李聪. 点-连式信号系统在市域铁路的可行性研究 [J]. 中国铁路, 2017 (6) : 43-47.
- [3] 张济民, 吴汶麒, 张树京. 准移动闭塞列车安全间隔时间的计算 [J]. 铁道学报, 1999, 21 (3) .
- [4] 高嵘华. 城市轨道交通点式 ATC 系统的改进研究 [J]. 铁道标准设计, 2015, 59 (5) : 158-160.
- [5] 赵跟党, 李乐. 点式列车自动保护 (IATP) 模式下车载信号丢失定位、无推荐速度故障分析及处理 [J]. 城市轨道交通, 2013 (10) : 103-105.
- [6] 何宗华. 城市轨道交通运营组织 [M]. 北京 : 中国建筑工业出版社, 2003.
- [7] 黄显. 区间信号与列车运行控制系统 [M]. 北京 : 中国铁道出版社, 2008.
- [8] 王瑞峰, 高继祥. 铁路信号运营基础 [M]. 北京 : 中国铁道出版社, 2014.

出版社, 2008.

- [9] 黄超. 高速列车追踪到达车站间隔时间计算方法初探 [J]. 铁路运输与经济, 2009, 31 (7) : 85-88.
- [10] 中华人民共和国住房和城乡建设部. 地铁设计规范 : GB 50157—2013 [S]. 北京 : 中国建筑工业出版社, 2013.

责任编辑 陈蓉

《铁路计算机应用》

2018年合订本 (限量版) 出版发行



合订本为大16开精装本, 全彩印刷, 每册定价 **160** 元。



**限量发行 100 套
从速订阅**

订购热线: 010-51849236
<http://www.tljsjyy.com>

微信公众订阅号: tljsjyy