

DB2 for z/OS Version 8.1

驱动业务价值

Julian Stuhler
Triton Consulting
2004 年 8 月

目录

摘要	3
关于作者	4
执行摘要	5
业务获益概览	5
投资回报	9
结论	11
简介	12
DB2 UDB for OS/390 Version 7 – 当今企业的基础	13
性能	13
可用性	13
Web 支持	14
应用程序开发	14
DB2 for z/OS V8 – 驱动业务价值	15
可伸缩性和 zSeries 增效作用	15
数据可用性	20
应用程序移植	24
Web 支持	27
提高性能并降低操作成本	29
开发人员/DBA 生产力	32
全球集成和部署	
数据安全	34
Version 8 业务获益总结	36
DB2 for z/OS Version 8 客户案例研究: Univar USA	40
验证升级	43
示例客户环境	43
场景 1: OLTP 客户	44
场景 2: SAP 客户	48
场景 3: Java 客户	52
结束语	56
附录 A – 参考文献	57
附录 B – 致谢	58

摘要

本文从 IT 执行官角度出发，给出了 IBM DB2 UDB for z/OS Version 8.1 主要新特性的高层次概览，其中重点介绍新版本能够交付的基础业务价值。本文还分析了在 Version 7 版本基础上更新的业务案例，包括能够轻松适应大多数特定客户环境的一般 ROI 模型。

注：本文中“Version 7”指“IBM DB2 UDB for OS/390 Version 7”，“Version 8”指“IBM DB2 UDB for z/OS Version 8.1”。

关于作者

Julian Stuhler 就职于英国 **Triton Consulting** 公司，担任首席顾问，该公司主要面向欧洲客户提供 **DB2** 配置咨询、教育和管理服务。**Julian Stuhler** 具有 17 年关系数据库经验，曾经为多个行业的客户提供服务，包括保险业、电信业、银行业、金融服务业和制造业。**Julian** 曾就广泛的 **DB2** 主题在英国和欧洲发表讲座，曾荣获 2000 International **DB2 User Group European Conference** 会议的“**Best Overall Speaker**”奖。他是 **IBM Redbook on Java Stored Procedures** 的作者之一，经常在 **DB2 Times** 和 **DBAZine** 等业内刊物上发表文章。

Julian 是一位 **IBM DB2** 金牌顾问，也是 **IDUG** 董事会的成员。您可以通过电子邮件 julian.stuhler@triton.co.uk 与他联系。

执行摘要

DB2 for z/OS Version 8 是 IBM 旗舰关系数据库管理系统中的最重要版本之一。新产品是在 Version 7 的稳固基础上构建的，经过全面增强和重新架构，它直接应对 IBM 最苛刻企业客户的众多关键挑战。

业务获益概览

以下部分总结了迁移到 DB2 Version 8 能够带来的业务获益。

可伸缩性和 zseries 增效作用

特性	业务获益	页码
64 位支持	Version 8 允许 IBM 最新一代硬件的客户完全利用 64 位存储器模型的庞大存储容量，而无需更改应用程序，进而允许 DB2 系统缓存更多数据并直接跃升到更高的水平，从而减少系统开销并更高效地使用有限的机器资源。在某些情况下，这些增强功能允许客户延迟仅与可伸缩性有关的数据共享的实现。虽然迁移到 64 位环境可能对 CPU 有更高的要求，但新版本显著的性能和生产力提高马上会抵偿 CPU 的这部分开销。	15
分区增强	Version 8 为数据库设计人员提供了功能更强的分区选项，允许在给定的机器资源条件下创建支持更高事务负载的高效 DB2 数据库设计。借助于新的动态模式更改功能，可以在现有数据库设计基础上实现这些获益，并最大限度地减少对可用性的影响。	18
恢复日志增强	Version 8 允许在线和离线保持更多恢复日志数据，从而允许 DB2 系统处理更大的工作负载，同时保持数据的可恢复性。	19
其他 zSeries 增效作用	Version 8 采用了 z/OS 和 zSeries 硬件中的特定功能，以提供更高效率的字符转换和加密服务。	20

数据可用性

特性	业务获益	页码
动态模式更改	Version 8 提供了完善的初始化功能集，能够确保在更改数据库结构的同时保证应用程序可以访问数据。客户可以利用这些功能实现更频繁的数据库结构更改，以便对不断变更的市场条件或意外性能问题做出迅速响应。由于更改过程更加简单，可以极大地提高数据库管理员的效率（同时显著降低了与数据库更改有关的风险）。	21
分区增强	Version 8 引入的全新分区模型首次实现了真正的独立分区。这项特性减少了表重组所产生的可用性影响，允许管理员更好地组织数据，因此从根本上提高了应用程序的性能。	23
动态系统参数	Version 8 中额外的动态系统参数降低了整个 DB2 系统在配置参数更改时可能遇到的风险，从而提高了整体可用性，并允许系统管理员对不断变更的应用程序要求做出更快的响应。	23
系统级备份和恢复	在拥有正确的 DASD 子系统后，新的系统级备份和恢复功能能够减少由系统备份造成的 DB2 数据不可用的时间。更重要的是，及时将系统快速恢复到给定点意味着系统在发生重要问题后能够以最短时间恢复可用性。	24

应用程序移植

特性	业务获益	页码
SQL 增强	这些 SQL 增强能够降低将数据库定义和应用程序代码移植到 DB2 Version 8 所需的工作量、复杂度和成本，使其成为一个更具吸引力的平台，可用于供应商应用程序移植、基础设施整合项目和 Windows、Unix 或 Linux 平台上的开发。	25
DB2 Universal Driver	新的 Universal Driver 提供了适用于所有平台（包括 DB2 for z/OS V7 和 V8）的单一驱动程序，从而在将应用程序移植到 zSeries 时无需额外的代码集更改，同时提供了有用的性能和功能增强。	26

Web 支持

特性	业务获益	页码
Java 增强	DB2 Universal Driver 中新的 Java 功能提供了一些有价值的生产力和性能增强，整合了主要的 Java zSeries 编程语言。	27
XML 增强	虽然我们仍需要使用 XML Extender 来分解和存储 DB2 中的 XML 数据，但 Version 8 内置的 XML 发布功能提供了一些友好的性能改进，并为将来的广泛的本机 XML 支持指出了方向。	28

性能和操作成本

特性	业务获益	页码
多行读取和插入	多行插入和读取命令是一项显著的性能改进，它能够潜在地使许多类型的应用程序获益，其中对于某些操作可节省 50% CPU 资源。对于分布式应用程序来说，可以用单次调用替代耗时的多次网络传输，因此获得更大的性能改进。	29
优化增强	DB2 领先的 SQL 优化器在 Version 8 中得到进一步增强，从而为许多公共查询提供了更高的效率和更低的 CPU 成本。	30
物化查询表 (MQT)	MQT 已经在 DB2 for Linux、Unix and Windows 中发挥了强大功效，它是大幅度缩短查询响应时间和减少资源消耗的有力方式，而无需查询用户付出任何工作。它是一项可用于决策支持/数据仓库环境的有价值的特性。	31
其他性能增强	这些性能增强在很大程度上减少了 CPU 需求并增大了客户工作负载的吞吐量。	31

开发人员/数据库管理员生产力

特性	业务获益	页码
生产力增强	Version 8 引入了大量重要的新功能，这些功能让 DB2 管理员和应用程序开发人员能够提高日常工作效率。随着目前人员成本所占比例不断提高，这些增强功能成为向 Version 8 迁移的极为重要的成本因素。	32

全球集成和部署

特性	业务获益	页码
Unicode 支持	Version 8 是基于以前版本所实现的初始 Unicode 支持构建的，删除了许多妨碍客户完全利用 Unicode 跨国应用程序环境的约束。	33

数据安全

特性	业务获益	页码
行级安全性	Version 8 的行级安全特性从根本上改变了 DB2 访问权限的管理方式。实现这些特性需要进行仔细的计划和析，但客户最终能够获得显著提高的灵活性，从而在安全管理和数据访问方面实现更高级别的控制和功能。	34

投资回报

以下几幅图总结了三种典型的 Version 8 客户环境的总体投资回报，它们分别是传统的 OLTP 环境、SAP 环境和 Java 环境。有关更详细的信息，请参阅第 43 页“验证升级”。

注意，虽然下图中使用的是实际的硬件和软件成本，但这些数据是在许多因素和假设的基础上概括得到的，对于不同的客户这些因素和假设可能大不相同。但这里所呈现的模型在分析潜在的财务状况方面，对于许多客户具有参考价值。

OLTP Customer

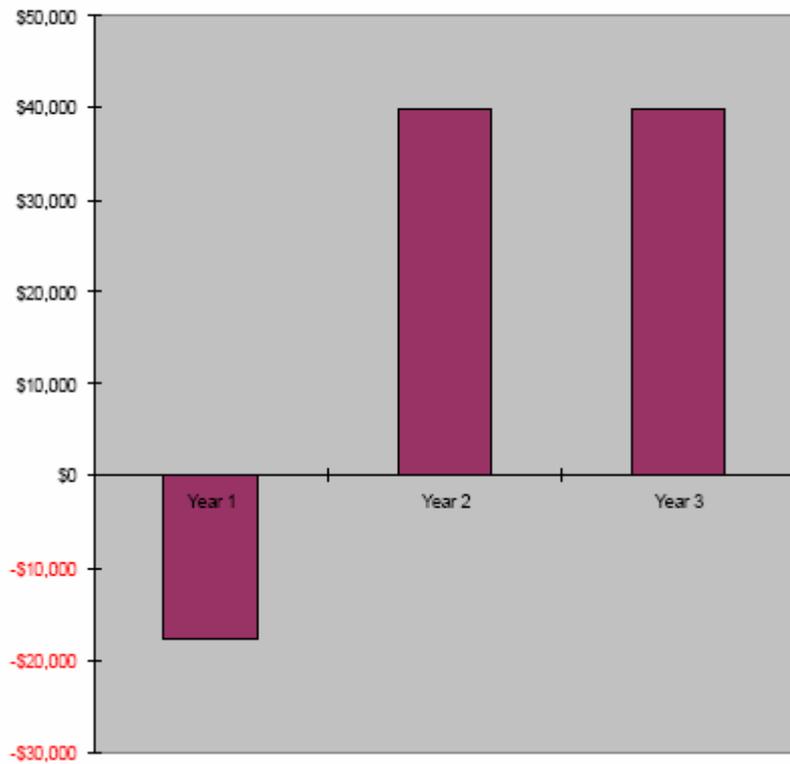


图 1 OLTP 客户总成本节约

SAP Customer

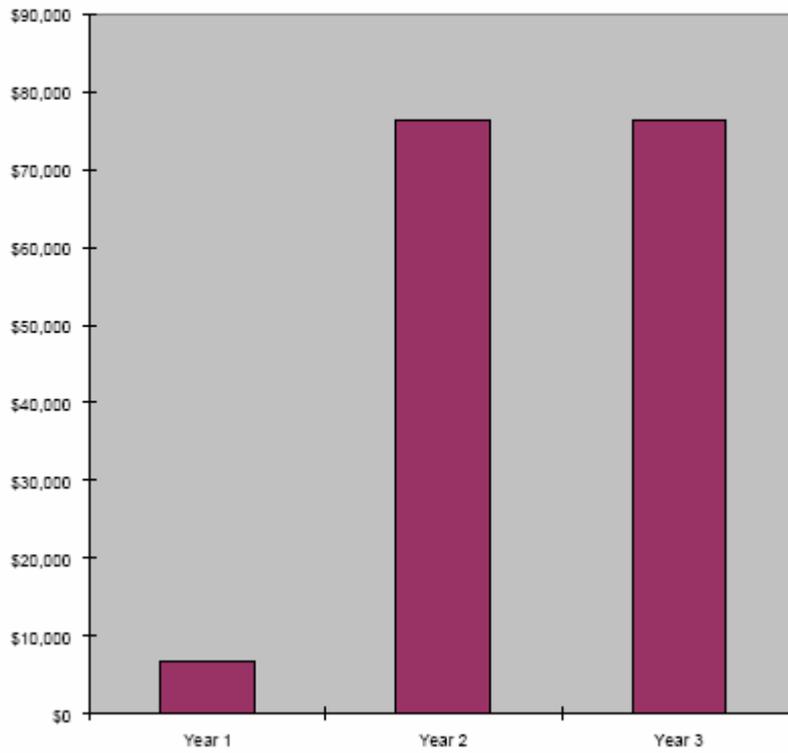


图 2 SAP 客户总成本节约

Java Customer

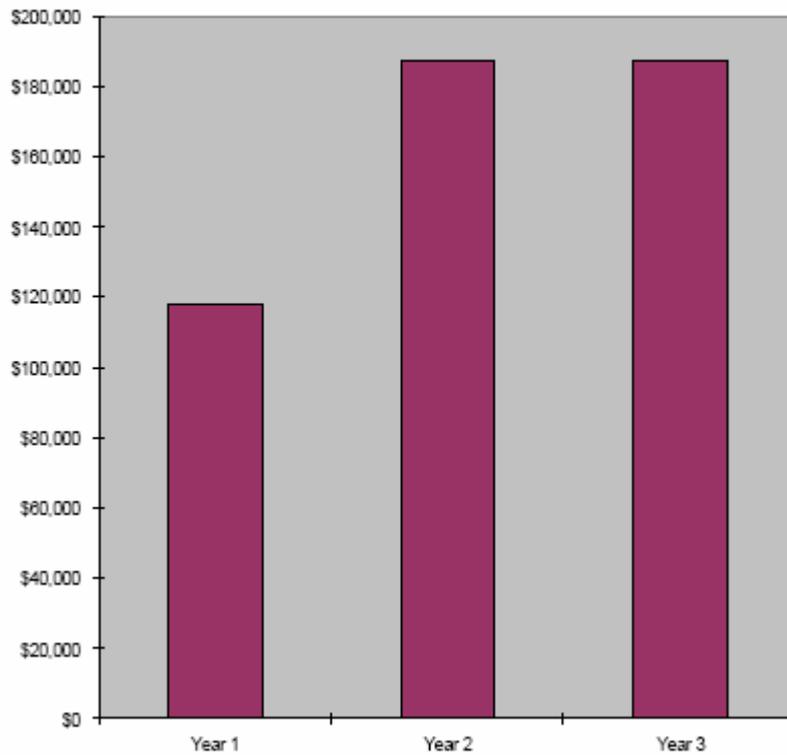


图 3 Java 客户总成本节约

结论

DB2 for z/OS Version 8.1 有望成为 IBM 旗舰数据库管理产品的最重要版本之一。它是首个突破 31 位计算限制的中间件产品，它利用了 IBM 的 64 位 zSeries 机器的强大可伸缩性，允许客户驱动更大的工作负载，并且“以更少资源完成更多工作”。

新产品在数据可用性方面也做了大幅改进，进一步减少了计划内宕机时间并能够满足当今 24/7 应用程序的更长时间的运行要求。利用这些特性，许多客户可以更频繁地更改数据库，从而允许客户对不断变更的市场条件或意外性能问题做出更快响应。

虽然向 64 位环境迁移引入了其固有的 CPU 要求，但 Version 8 的重要性能增强无需更改（或只需较少更改）应用程序即可实现，因此所获得的性能提高很容易抵消增加的 CPU 开销。性能的增强降低了 DB2 应用程序的总体拥有成本，令每项事务的执行成本更低廉，并允许服务器处理更大的工作负载。

新版本还根除了许多妨碍非 DB2 应用程序向 zSeries 平台移植的壁垒。如第 40 页的案例研究所示，移除这些壁垒后为以具有竞争力的成本创建服务器整合项目开辟了一条途径。

最后，由于产品具有更完善的自动化和智能，因此能够实现显著的生产力节约。这将经验丰富的 DB2 管理员和开发人员从低级例行任务中解放出来，投入更多时间来编写更好的、更高效的应用程序。

总之，这些新特性代表了 DB2 功能的重要改进，在迁移到新版本后，客户将获得更大的财务收益和技术收益。

简介

数据是任何组织的血液，随着当今电子商务应用程序日益增加的可用性和性能要求，企业数据库的选择变得越来越重要。

DB2 for z/OS Version 8 是 IBM 旗舰关系数据库管理系统中最重要的版本之一。新产品是在 **Version 7** 的稳固基础上构建的，经过全面增强和重新架构，它直接应对 IBM 最苛刻企业客户的众多关键挑战。

大量资料介绍了 **Version 8** 中的技术变更，但有关这些新特性如何改进业务方面只有很少的研究。本文将从 IT 执行官视角出发，给出 **IBM DB2 UDB for z/OS Version 8.1** 主要新特性的高层次概览，其中重点介绍新版本能够交付的基础业务价值。本文还分析了在 **Version 7** 版本基础上更新的业务案例，包括能够轻松适应大多数特定客户环境的一般 ROI 模型。

DB2 UDB for OS/390 Version 7 — 当今企业的基础

今天, DB2 作为 IBM 主机架构的主要数据库系统已经为人们普遍接受。虽然该平台也使用其他产品¹, 但在过去 20 年中, DB2 一直是大多数业务关键型主机 IT 应用程序的心脏。

自 DB2 Version 7 2001 年初面世以来, 超过 2/3 的 IBM DB2 用户使用该版本。IBM zSeries 平台的可伸缩性和可靠性使其成为客户的高容量、任务关键型应用程序的最佳选择。

注意, Version 7 以前的版本不能直接迁移到 Version 8, 因此客户要想利用 Version 8 的新特性, Version 7 是重要的基础。

本小节首先简要回顾一下 Db2 UDB for OS/390 Version 7, 并介绍一下它如何帮助客户获取竞争优势。

性能

提高性能并由此降低支持给定工作负载所需的资源是任何新版本的关键目标。Version 7 在这方面交付了许多重要的增强功能, 包括:

- **SQL 增强。**一系列与性能有关的 SQL 增强有助于减少资源开销和工作负载成本, 其中包括 DB2 对超大量插入过程的处理能力的增强, 它能将结果集限定为行的集合, 并避免不必要的排序操作。
- **滚动光标。**许多应用程序允许用户在浏览数据时前后翻页。用户每次前后翻页时通常要求执行单独的 SQL 语句。Version 7 引入了新滚动光标功能, 它允许应用程序开发人员在结果集中前后导航, 而只需从表中进行一次检索。

可用性

减少计划内宕机时间并增强可用性能够帮助 DB2 更好地支持高要求的电子商务应用程序。Version 7 在这方面交付了一些重要的新功能。

- **在线子系统参数。**Version 7 率先实现了这项重要功能, 它允许系统管理员在无需停止和重启 DB2 系统的情况下, 修改许多重要的系统配置参数。Version 8 进一步增强了此功能 (请参阅第 23 页的“动态系统参数”)。

¹ 包括 IBM 自己的 DB2 for Linux、Unix and Windows, 在 Linux for zSeries 平台上运行。

- **Restart Light。** 这项增强功能能够减少偶尔发生的 DB2 成员共享数据组失败而导致的数据不可用时间。

Web 支持

Version 7 在 DB2 电子商务应用程序中向前跃进了一大步。此版本交付了以下基于 Web 的功能：

- **Unicode 支持。** Version 7 率先引入了 Unicode² 存储支持，允许以 Unicode 格式在 DB2 表中存储用户数据。这项增强功能使跨国应用程序能够非常简单地存储和显示正确的数据，而不用考虑用户的本机语言。Version 8 进一步扩展了 DB2 的 Unicode 功能，请参阅第 33 页“全球集成和部署”。
- **网络监控。** 分布式应用程序所引入的关键挑战之一是如何在需要改进性能时缩短总体响应时间。Version 7 引入了一项新功能，允许 DB2 将有用的性能信息向回传递给调用的客户机。这使数据库管理员能够更快更简单地诊断和解决分布式应用程序的性能问题。
- **支持 JDBC 2.0。** JDBC 是 Java 应用程序与关系数据库之间进行连接的行业标准。DB2 Version 7 引入了 JDBC 2.0 标准支持。这实现了大量的功能和性能改进，包括新数据类型、滚动光标的支持和连接池。

应用程序开发

提高开发人员的生产力对应用程序总体开发和维护成本有直接和显著的影响。Version 7 在这方面引入了以下几项增强功能：

- **Java 存储过程。** 存储过程能够促进代码重用并缩短许多应用程序的响应时间。Version 7 引入了高性能的 Java 存储过程支持。以前的版本在 Java 代码能够执行之前需要开发人员执行复杂的编译过程，但 Version 7 不再需要这些工作。
- **SQL 存储过程。** Version 7 显著增强³了一项新类型的存储过程，它是完全用 ISO 标准的 SQL 语言编写的存储过程，而不是采用传统的编程语言编写。这些存储过程更容易在平台之间移植，与采用传统语言编写的存储过程相比，它们更易于编写和实现。
- **新的扩展程序。** Version 7 在以前的 Version 6 的对象关系功能基础上提供了 XML 的扩展程序。扩展程序允许用户从 DB2 数据生成 XML 文档，以及分解 XML 文档并将分解结果存储在 DB2 中。Version 8 中通过新的本机 XML 功能扩展了这些功能，请参阅第 28 页的“XML 增强”。

² Unicode — 全球字符编码标准，旨在允许应用程序正确地存储和显示字符，而不管平台、语言或程序。

³ DB2 Version 5 和 Version 6 也具有 SQL 存储过程功能（需要单独安装）。

DB2 for z/OS V8 — 驱动业务价值

本小节详细介绍 Version 8 的主要特性，以及 IBM 的最大型企业客户如何能够利用这些特性来为他们的企业交付 IT 服务。

可伸缩性和 zSeries 增效作用

可伸缩性是轻松地增加系统功能以满足要求的能力，可伸缩性一直是一项重要的能力。当今的 IT 系统通常与机构的合作伙伴、供应商和客户紧密集成，必须对客户要求、新的市场机遇或竞争威胁迅速做出响应。IBM 将这样的应用程序称为“按需应变”的应用程序，它们必须时刻处理处于高峰期的事务吞吐量，在几年前这样的吞吐量是前所未闻的。

在这种环境下，以快速和经济高效的方式扩展核心基础设施（例如数据库系统）的处理能力是至关重要的。

64 位支持

在降低 IT 成本和交付更大价值的努力中，任何客户都在不断尝试以更少资源完成更多工作。他们希望能够在不增加 IT 相关人员的情况下，管理更少的机器，维护更少的数据库，以及支持更大更的工作负载。这些驱动因素意味着 DB2 系统需要处理不断增多的工作量。

目前，存储约束可能成为单一 DB2 Version 7 系统的可伸缩性的最大制约因素。该系统中并发运行的每个进程都需要一定的存储空间，因此系统需要处理的工作负载越多，存储需求就越大。近年来，单个 DB2 系统的存储需求每年增长 20 - 30%，以支持系统上运行的不断增加的工作负载要求。

自从 90 年代中期 DB2 Version 4 引入数据共享以来，通过将多个 DB2 系统添加到单个的“数据共享组”中，可以横向地扩展存储能力，其中所有组成员共享相当的数据集访问，并通过 Coupling Facility 进行协调（参见图 4 “DB2 数据共享架构”）。

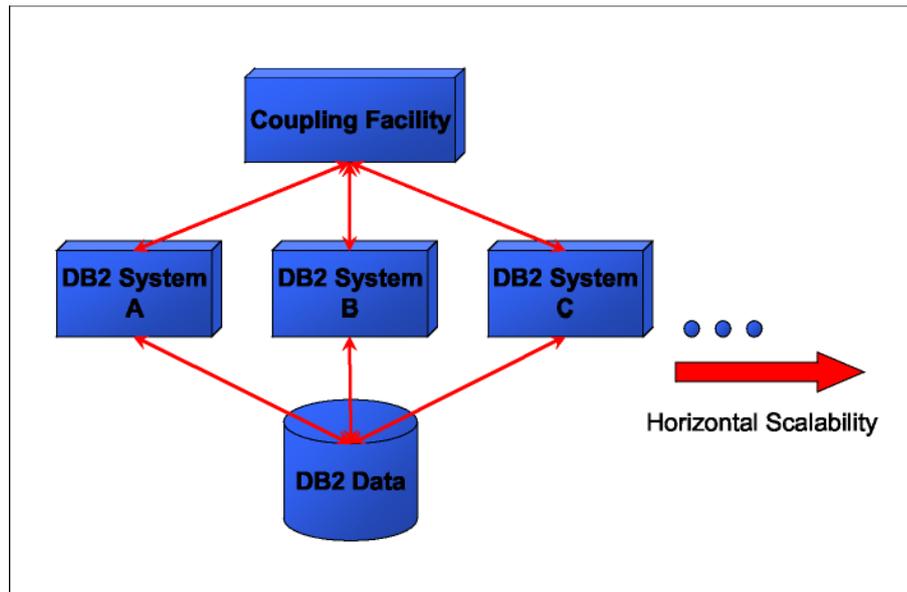


图 4 DB2 数据共享架构

这种数据共享方法提供了优秀的可伸缩性和弹性。但向该数据共享环境迁移并不是轻易之举，因为可能存在潜在的性能问题和操作环境的复杂性，同时还要考虑财务问题，例如购买耦合设施和软件许可证的成本。

由于这些原因，大多数 DB2 客户尝试在数据共享组中最大限度地减少子系统的个数。一些对弹性要求不高的客户甚至可以不使用这样的数据共享，而是采用纵向方式增加单个 DB2 系统的个数。

那么，妨碍这种纵向可伸缩性的存储约束是什么呢？

归根结底并不是存储器本身的价格问题。zSeries 增量存储器成本已稳步下降，目前已降至 10,000 美元/GB，仅相当于 2003 年年中 z990 发布时的价格的 1/4。

因此，存储器本身的成本已不再成为问题，问题在于 DB2 充分利用其存储器的能力。未采用最新的 64 位模式的 zSeries 硬件⁴的客户仅限于 31 位实际存储寻址能力（参见图 5 “z/OS 64 位内存映射”）。实际上，这意味着任何应用程序（包括 DB2）只能直接访问最大为 2 GB 的“中央”存储器。任何额外的存储器（随着存储器价格的快速下滑，8 GB 以上的存储器已较为常见）必须作为单独的“扩展”存储器来使用，虽然 DB2 能够使用这部分存储器来存储只读数据，但与“中央”存储器相比，它们的访问效率较低（因此更昂贵）。虽然对于 DB2 来说，从扩展存储器中提取数据比从磁盘提取数据更高效，但扩展存储器的使用还具有其他限制，从而使其不能成为一个理想的解决方案。

如果您正在使用 64 位模式下的最新 zSeries 机器和正确的 OS/390 或 z/OS 版本⁵，那么您将不受这些限制，且理论上可达到 16 Exabytes 的寻址能力⁶（ 2^{64} 字节）。这彻底排除了“扩展”存储器的概念，并允许 DB2 直接使用更大的内存⁷，进而允许以更高效的方式进行数据访问。

⁴ 撰写本文时仅限于 IBM eServer 模型 z800、z890、z900、z990。

⁵ OS/390 V2R1 或更高版本。

⁶ 但是，撰写本文时任何硬件配置的最大可用存储器为 256 GB（在 IBM eServer Z990 机器上）。

⁷ 最高达 32 GB。

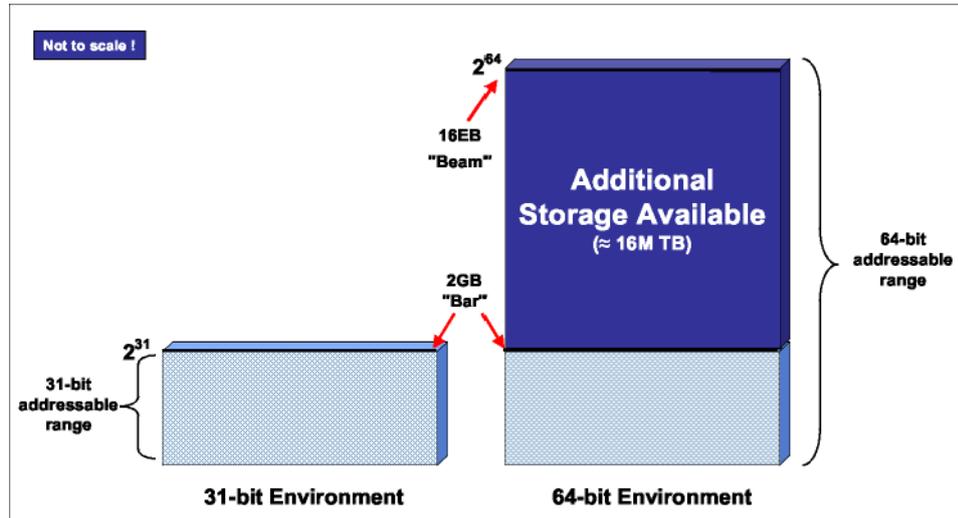


图 5 z/OS 64 位内存映射

但是，虽然 DB2 Version 7 能够缓存更多数据，但从内部来讲它不能完全采用新的 64 位内存模型。关键的内部存储区域仍然限制在 2 GB 内存以下，这样限制了 DB2 系统能够支持的并发事务数。因此，即使对于具有很大存储容量的系统，纵向可伸缩性仍然受到限制，并且客户可能不得不继续进行横向伸缩。对于最高可配置 256 GB 的大型 eServer 模型来说，32 GB 的最大数据缓存区尚不能满足这些模型的要求。

Version 8 经过了全面的重新架构以解决此问题，它完全采用最新的 64 位硬件和操作系统。Version 8 的总体构想之一是“突破限制”，在其存储器容量上充分体现了这一点，DB2 目前可直接寻址 zSeries 机器中的所有可用存储器。

随着处理器以超过磁盘子系统的速度持续增长，执行 I/O 的相关成本正不断增加，因此最小化 I/O 成为改进性能的重要目标。Version 8 允许的 DB2 缓存数据量从以前的 32 GB 增加到 1 Tb⁸。这能够极大地降低支持给定工作负载所需的 I/O 总量，从而允许在不升级 CPU 的情况下实现更大的吞吐量。在某些情况下，这能够将 I/O 密集进程（例如批处理程序）的响应时间减少一个数量级。缓存的增加还简化了 DB2 的存储器使用，将所有数据缓存区整合为一个单个 z/OS 地址空间，因此简化了 DB2 系统管理和操作。

除数据缓存功能增强之外，DB2 的内部代码和存储利用率也得到提高，从而使其能够充分利用 64 位计算环境所提供的更大的可用存储空间。因此，通过单个子系统的能力显著增强，解决了许多原有的工作量限制问题（例如，任意时刻可以打开的数据集的最大数目从 32,000 增加到 100,000）。

⁸ 注意这是 Version 8 的架构限制，实际上数据缓存应完全受 zSeries 服务器存储器支持，因此完全配置的 IBM eServer z990 的当前限制为 256 GB 或 0.25 TB。

综合来看, 这些 64 位增强功能将允许更多客户减少支持特定工作负载所需的 DB2 系统总数, 因此相应降低了系统开销。实现这些获益无需更改任何应用程序, 而只需升级到 Version 8。

但是, 务必注意迁移到 64 位环境并非不需任何成本。在 64 位环境下 DB2 需要执行额外的工作, 某些客户可能需要将 CPU 的速度提高约 5 -10%。但正如本文其他部分所述, 性能和生产力的显著提高足以抵消 CPU 的开销。全面考虑所有这些成本和获益对于最后制订迁移决策是非常重要的 (参见第 43 页 “升级决策”)。

底线: Version 8 允许 IBM 最新一代硬件的客户完全利用 64 位存储器模型的庞大存储容量, 而无需更改应用程序, 进而允许 DB2 系统缓存更多数据并直接跃升到更高的水平, 从而减少系统开销并更高效地使用有限的机器资源。在某些情况下, 这些增强功能允许客户延迟仅与可伸缩性有关的数据共享的实现。虽然迁移到 64 位环境可能对 CPU 有更高的要求, 但新版本显著的性能和生产力提高马上会抵偿 CPU 的这部分开销。

分区增强

一直以来, DB2 允许设计人员将大容量的表分割为一些较小的片段 (称为分区), 以方便进行管理。在 Version 8 版本之前, 最大分区数限制为 254。

这种限制成为数据库设计人员的一大障碍, 设计者经常需要更多的分区支持 (例如, 对于时间序列数据的存储需求来说, 通常要求在每个分区中存储 1 天的数据, 在线数据需要保存 7 年, 这样共需 2,548 个分区)。由于分区限制的缘故, 设计人员往往要对数据库设计采取折衷方案, 而这通常导致数据库无法达到最佳查询性能。在某些情况下, 数据库还需要更多的干扰性维护工作, 从而影响了数据可用性。

DB2 Version 8 突破了 254 分区限制, 允许单个表最多有 4,096 个分区。这让数据库设计人员具有更强的灵活性, 能够针对给定的表选择最高效的分区方案。

Version 8 的另一项重要分区增强功能是关于表中的数据物理排序方式。在以前的版本中, 必须使用与数据分区相同的属性对数据进行物理排序。在下面图 6 显示的示例中, 按照客户号对保存客户数据的表进行了分区, 因此每个分区中的数据必须按照客户号进行排序。同样, 这种限制迫使数据库设计人员在设计中采取折衷方案, 因为最优的分区属性与最优的物理排序属性常常是不同的。

Customer Table					
Customer No	Acct Type	Customer No	Acct Type	Customer No	Acct Type
1	A	101	B	201	C
2	B	102	B	202	A
3	A	103	C	203	A
4	C	104	A	204	C
...		
Customers 1-100		Customers 101 - 200		Customers 201-300	

图 6 Version 7 的分区

Version 8 首次突破了这种限制，允许数据库设计人员根据分区和物理排序各自的特点选择最高效的排序属性。在下面图 7 显示的示例中，按照客户号对表进行了分区，而每个分区中的数据按帐户类型进行物理排序。这项增强功能允许更高效地运行查询，从而减少了给定工作负载所需的机器资源。

Customer Table					
Customer No	Acct Type	Customer No	Acct Type	Customer No	Acct Type
1	A	104	A	202	A
3	A	101	B	203	A
2	B	102	B	201	C
4	C	103	C	204	C
...		
Customers 1-100		Customers 101 - 200		Customers 201-300	

图 7 Version 8 的分区

我们可以使用 Version 8 新的动态模式更改功能在现有表上实现这两项分区增强功能，这能够极大地降低风险、减少工作量并降低对数据可用性的影响（有关更详细的信息请参阅第 21 页）。

底线： Version 8 为 DB2 数据库设计人员提供了功能更强的分区选项，允许在给定的机器资源条件下创建支持更高事务负载的高效 DB2 数据库设计。借助于新的动态模式更改功能，可以在现有数据库设计基础上实现这些获益，并最大限度地减少对可用性的影响。

恢复日志增强

随着单个 DB2 子系统吞吐量的增加，子系统所生成的恢复日志数据总量也随之增加。保持恢复日志数据的在线可用性是至关重要的，因为这能够加速恢复操作并最小化数据可用性影响。

DB2 Version 7 最多允许 31 个在线恢复日志数据集⁹。Version 8 将这个限制提高到 93 个日志数据集¹⁰，允许 DB2 恢复日志数据在线保存更长时间，完全能够满足由上面所描述的可伸缩性增强而可能生成的更大吞吐量。

相关的增强功能也增加了离线恢复日志的数目，DB2 Version 8 能够使用 1,000 - 10,000 个数据集。这增加了 DB2 可以访问的日志数据的范围¹¹，允许减小某些数据的备份频度，从而节省 CPU 成本并减少日常的维护工作。

底线： Version 8 允许在线和离线保持更多恢复日志数据，从而允许 DB2 系统处理更大的工作负载，同时保持数据的可恢复性。

其他 zSeries 和 z/OS 增效作用

zSeries 硬件、z/OS 操作系统和 DB2 Version 8 之间的其他增效作用示例包括：

- **Unicode 转换服务。** 特定的 zSeries 机器¹²具有特定的硬件指令，这些指令在 Unicode 转换中起到辅助作用，z/OS V1.4 中的相关增强功能利用这些指令来大幅改进字符转换的性能。Version 8 也能够执行某些内部转换，并进一步流线化转换过程（第 33 页“全球集成和部署”详细介绍了迁移到 Unicode 的优势）。
- **加密支持。** Version 8 采用 z/OS Integrated Cryptographic Service Facility 来提高敏感信息向分布式客户机传输过程中的安全性。

底线： Version 8 采用了 z/OS 和 zSeries 硬件中的特定功能，以提供更高效的字符转换和加密服务。

数据可用性

zSeries 平台所提供的弹性和可靠性是其成为当今任务关键型应用程序平台的主要原因。不幸的是，虽然该平台最大限度地减少了意外宕机事件，但仍需要一定的计划内停机来执行必需的系统维护。这些计划停机成为越来越突出的问题，在维护过程中所有批处理窗口都需关闭，这些窗口的维护使许多客户不堪重负。全球大多数按需应变¹³的应用程序要求数据必须具有全天候的在线可用性（24×7×365）。在这样的环境中，任何形式的计划停机都将成为主要的业务问题。

在数据库维护中，有三种主要原因导致数据不可用：

⁹ 如果日志定义为最大容量，这些数据集约等于 124 GB 恢复日志信息。

¹⁰ 如果日志定义为最大容量，这些数据集约等于 372 GB 恢复日志信息。

¹¹ 从 4 Tb 增加到 40 Tb。

¹² 这些指令最初是从 IBM eServer z900 引入的，在 IBM eServer Z990 上进行了进一步增强。

¹³ 按需应变是 IBM 的 IT 系统的标志，这些系统与机构的合作伙伴、供应商或客户进行了紧密集成，必须对客户要求、新的市场机遇或竞争威胁迅速做出响应。

- **数据整理。**针对数据库运行特殊的实用程序以进行备份、收集数据库统计信息，以及重新组织数据以便提高访问效率，等等。
- **系统配置更改。**DB2 是高度可配置的数据库管理系统，带有大量控制系统操作方式的系统参数（称为 DSNZPARMS）。许多参数的更改要求系统停止运行并重启才能生效，这导致系统级停机，从而对各种不同的应用程序造成影响。
- **模式更改。**随着新版本应用程序进入产品环境，数据库结构经常需要进行相关修改。性能问题的解决可能也需要修改数据库结构，根据修改的性质不同，有些修改可能导致长时间的系统停机。

在数据整理方面 IBM 取得了极大进步，在最近几个版本中，IBM 增强了 DB2 的实用程序功能，允许在更多数据库维护过程中保证应用程序可以读取和更新数据。虽然这个问题尚未完全解决，但在设计完善的应用程序中常常可以将其影响控制在可管理的范围内。数据整理中最难应对的是表的分区问题¹⁴。分区的最重要原因之一是减小日常数据整理操作的影响，将表分区之后，可以对单个分区进行数据整理，因此允许应用程序访问其他分区，从而提高数据可用性。但是，大多数表所定义的索引并不具有这种“分区独立性”，因此即使是对某个分区进行维护，但在在数据维护期间，对整个表的访问仍会受到限制。

Version 7 的增强功能组合允许动态修改系统配置和数据共享，从而解决了大部分与系统配置修改相关的问题（重新配置数据共享组的一部分成员时，其他成员中运行的应用程序可以继续访问数据）。

但由于模式更改引起的停机仍然是一个主要问题，并导致许多客户无法访问数据。

动态模式更改

任务关键型的按需应变应用程序的标志性特征是应用程序开发周期很短。企业需要对不断变更的市场条件和竞争对手的举措迅速做出响应，这通常意味着要对应用程序进行修改并在产品环境中实现新版本。应用程序的修改常常需要同时对数据库结构进行修改。

即使应用程序本身相对稳定，但由于操作上的原因，数据库管理员往往需要更改数据库的定义，例如修改数据库以改进关键业务流程的性能，或在分区的表空间内部调整数据分布以适应有可能出现的增长模式。

¹⁴ 由于性能、可用性或可管理性的需要，将表（通常是大量表）分为多个部分。

在 **Version 8** 之前，在不影响数据可用性的情况下，DB2 仅允许对现有结构做很少的修改，例如可以向现有数据库添加新表和索引，或在现有表的结尾处添加新列，也可以修改某些与存储有关的参数，但任何更重要的修改（例如删除列或将某列从一种数据类型转换为另一种数据类型）需要放弃整个表并重新创建。必须首先卸载表中的数据，修改之后再重新加载。这个过程（如图 8 所示）可能需要几小时，大的表甚至需要几天，期间任何应用程序均无法访问表中的数据。

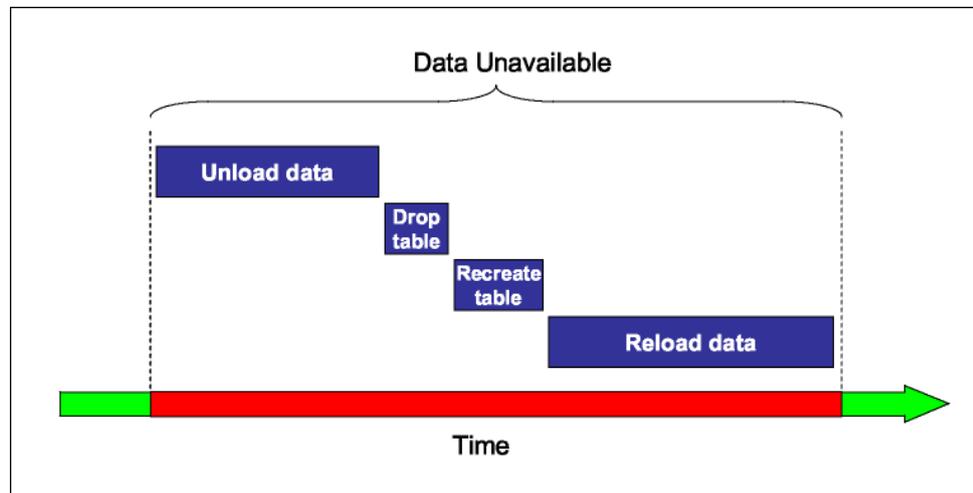


图 8 在更改模式期间数据不可用

Version 8 引入了 IBM 对这个问题的首个解决方案，即“动态模式更改”。该解决方案的概念很简单，即允许对数据库结构进行动态更改，同时保证应用程序可以完全访问数据。

IBM 确定了目前导致数据不可用的最常见的修改，并针对这些修改在 **Version 8** 实现了增强功能。新功能包括：

- 更改列数据类型和长度。这是一项很常见的修改，例如表示客户号的数字类型的列最初定义为 8 位数字，但由于意外的高业务量，必须将它增至 10 位。
- 将列添加到索引。这是在性能调优过程中常见的需求，让管理员不必删除和重建索引。
- 更改数据存储表的物理排序，这是由于性能原因而经常需要实现的一项修改。
- 将新的分区添加到表中，并在现有分区表中重新分布数据。由于数据的增长方式常常与数据库设计时预期方式不同，因此这是经常需要进行的一项更改。

这些新特性极大地减少了动态模式更改中的数据不可用时间，允许数据保持更长时间的可访问性。虽然我们仍面临一些挑战（初始功能集中尚不支持其他一些模式更改，在线更改后会有短期的性能下降），但 IBM 已率先在正确的方向的迈进了重要的一步，在将来版本中还会对这些特性进行扩充。

“DB2 for z/OS 的每个新版本中的可用性增强对于 GAD 来说都是至关重要的，最新引入的 Online Schema Evolution 赢得了所有数据库管理员的好评，他们正在期待该产品将来版本的进一步增强。”

Claus Weidenfeller, Lead DBA and DB2 Early Program 项目经理, GAD

我们已对分区旋转 (partition rotation)、添加和数据的重新分布进行了测试，这些特性将提高可用性并降低修改风险

Version 8 Beta 版用户

底线: Version 8 提供了完善的初始化功能集，能够确保在更改数据库结构的同时保证应用程序可以访问数据。客户可以利用这些功能实现更频繁的数据库结构更改，以便对不断变更的市场条件或意外性能问题做出迅速响应。由于更改过程更加简单，可以极大地提高数据库管理员的效率（同时显著降低了与数据库更改有关的风险）。

分区增强

正如第 15 页“可伸缩性”小节所讨论的一样，Version 8 引入了一些重要的分区增强功能。除提高总体可伸缩性外，这些增强功能还提供了一些非常有用的可用性优势。

使用 Version 8 引入的新分区方法，每个分区能够真正独立于其他分区。这允许对给定分区进行数据维护时，应用程序可以完全访问其他表分区的数据。

此外，Version 8 允许创建更多的分区，因此每个分区可以更小，从而进一步将维护活动的影响控制在更小的范围内。

底线: Version 8 引入的全新分区模型首次实现了真正的独立分区。这项特性减少了表重组所产生的可用性影响，允许管理员更好地组织数据，因此从根本上提高了应用程序的性能。

动态系统参数

IBM 继续增强了 Version 7 的动态系统参数特性，允许在 DB2 运行中更改许多 DB2 系统配置参数 (DSNZPARMS)。Version 7 允许以这种方式修改约 60 个参数，Version 8 在此基础上添加了超过 40 个参数（包括 Version 8 引入的新的参数，以及原有的参数）。

底线: Version 8 中额外的动态系统参数降低了整个 DB2 系统在配置参数更改时可能遇到的风险，从而提高了整体可用性，并允许系统管理员对不断变更的应用程序要求做出更快的响应。

系统级备份和恢复

数据库管理员的一个最重要目标是确保正确备份 DB2 数据，以便在数据丢失或系统崩溃时恢复数据。不同的数据备份技术也会对数据可用性产生很大影响。

Version 8 引入了通过单一实用程序备份整个 DB2 系统的功能（包括所有应用程序数据）。Version 8 的另一个实用程序允许将整个系统恢复到用户定义的时间点。这项特性使用 z/OS 中的新功能来调用 ESS Flashcopy，ESS Flashcopy 能够非常迅速地进行 DB2 系统所有数据的卷级（volume-level）完全备份。虽然恢复整个 DB2 系统的情况非常少见，但这项功能对于运行 CRM/ERP 供应商应用程序（这些应用程序通常驻留在它们自己的子系统中）的 DB2 系统来说将产生极大的获益。

底线：在拥有正确的 DASD 子系统后，新的系统级备份和恢复功能能够减少由系统备份造成的 DB2 数据不可用的时间。更重要的是，及时将系统快速恢复到给定点意味着系统在发生重要问题后能够以最短时间恢复可用性。

应用程序移植

随着与主机相关的硬件、软件和环境成本的不断下降，zSeries 逐渐成为一些最初为其他平台设计开发的应用程序的主机平台。对于许多客户来说，将现有应用程序从其他平台移植到 zSeries 平台已成为当务之急，以便可以充分利用该平台行业领先的伸缩性和弹性。有些客户还倾向于在分布式环境中（例如 Windows）开发新应用程序以提高开发人员生产力，然后在最后的测试和实现阶段将程序移植到主机平台。ERP 和 CRM 供应商还希望将其现有应用程序代码移植到运行 DB2 的主机平台上。

不管出于何种原因，应用程序移植过程都面临大量问题，因此移植过程也随之变得更加复杂、风险更高，成本也相应增加。这些问题包括：

- **DB2 for LUW 数据库不兼容性。**DB2 在 zSeries 上的实现与分布式平台有所不同，这些区别可能要求移植时对应用程序代码和/或数据库结构定义进行修改。
- **ISV 数据库不兼容性。**DB2 与其他数据库（例如 Oracle 和 SQL Server）之间也存在区别，这些不兼容性可能是主要障碍，因此相应需要更多措施来克服这些问题。

从应用程序角度来看，IBM 在最近几个主机和分布式 DB2 产品开发周期中逐步减少了这种内部不兼容性。但是，但是仍然存在某些重要的区别，根据应用程序规模和复杂性不同，这些区别可能需要大量的努力来克服。典型的例子是表名称的最大长度问题。分布式平台上的 DB2 允许表名称最长可为 128 字符，而 DB2 for OS/390 & z/OS Version 7 的表名称最长只能为 18 字符。当将应用程序移植到主机上时，此限制可能需要对数据库定义和应用程序代码进行大量修改。

IBM 不断向（直接或间接）提供类似于其竞争 ISV 数据库产品功能的 DB2 添加新功能。每次添加新功能后，移植过程就变得更容易一些，但到 Version 7 为止，数据库迁移仍然需要克服许多重要挑战。

Version 8 突破了大多数可移植性壁垒，为系统移植提供了更多途径，包括以前耗资巨大的大规模服务器整合。请参阅第 40 页的“DB2 for z/OS Version 8 客户案例研究”，其中详细介绍了这些方法在实际项目中所能交付的财务获益。

SQL 增强

与 DB2 Version 7 相比，许多其他数据库系统（包括 DB2 for Linux、Unix and Windows 的当前版本）支持更长的表名称和列名称。这是应用程序移植的主要障碍之一，因为无论是在数据库定义中，还是在应用程序代码的每条 SQL 语句中，必须将这些长名称缩短，这是一项耗时的过程。Version 8 将表名称长度从 18 字符增加到 128 字符，列名称长度从 18 字符增加到 30 字符（这个限制与 Version 8 of DB2 for Linux, Unix and Windows 中的限制相同），这样在从大多数数据库系统进行移植时，无需修改表名称和列名称。

新版本还增加了单条 SQL 语句的最大大小，从 Version 7 的 32 K 增至 Version 8 的 2 MB。这项更改不仅能够适应由长名称引起的 SQL 语句长度的增加，而且允许移植为其他平台编写的长 SQL 语句，而避免了手工重新编码所需的成本和时间延迟。

Version 8 允许在一次连接操作中最多引用 225 个表（Version 7 中的上限为 15 个）。这允许大量 SQL 语句只需较少修改（或无需修改）即可在 DB2 for z/OS 平台上执行，而不再像以前一样需要重新编码。这项增强功能对 ERP/CRM 供应商和程序具有特别的价值，因为这些应用程序中广泛使用了大型表的连接。

将应用程序移植到 zSeries 平台的过程中，排序的区别是隐蔽性较强的问题之一（也称为“collating sequence”）。由于针对主机和非主机字符的不同编码模式，相同的 SQL 语句在 DB2 for z/OS 上执行所返回的数据顺序可能与分布式平台所返回顺序不同。例如，典型的主机查询将首先返回小写字符，后跟大写字符，然后是数字。在分布式平台上，相同的查询将首先返回数字，后跟大写字符，然后才是小写字符。在某些应用程序中这可能引起意外的结果，并需要额外的编码来解决。正如第 33 页所描述的，Version 8 大量使用了 Unicode，并在内部采用 Unicode 格式来处理所有 SQL 语句。因为大多数分布式应用程序要么在本机使用 Unicode，要么使用 ASCII（其排序与 Unicode 相同），因此两个平台上使用的排序序列是相同的，从而使移植更简单。

除提高开发人员开发新应用程序的效率之外，Version 8 的大量其他 SQL 增强功能均有助于提高应用程序的可移植性，这些增强功能包括：

- 引入了大量新的 SQL 功能，这些功能直接等效于 DB2 for LUW 和其他供应商所提供的功能（在许多情况下语法上是等同的）。
- 索引关键字的最大长度显著增加¹⁵。正如前面所介绍的 SQL 名称长度增加一样，这避免了 DB2 for z/OS 移植中对数据库设计的更改需求。
- 许多非 DB2 应用程序使用称为“sequence”的对象来自动生成惟一的数字。DB2 for z/OS 的以前版本支持类似的概念，称为“identity column”，但它们的灵活性较低并且需要特殊处理。Version 8 引入了序列支持，从而增强了与其他供应商数据库的兼容性。
- DB2 for Linux, Unix & Windows 目前支持递归 SQL（也称为“bill of materials”），Version 8 中也支持这些查询，在将应用程序移植到 zSeries 过程中这避免了不必要的重新编码的复杂任务。

GAD 通过将现有 UNIX DB2 环境移植到 DB2 for z/OS 平台，成功地验证了 Version 8 对 DB2 产品家族的集成支持，移植过程中未出现任何问题，只是需要创建某些未自动生成的索引。

Claus Weidenfeller, Lead DBA and DB2 Early Program 项目经理, GAD:

底线：这些 SQL 增强能够降低将数据库定义和应用程序代码移植到 DB2 Version 8 所需的工作量、复杂度和成本，使其成为一个更具吸引力的平台，可用于供应商应用程序移植、基础设施整合项目和 Windows、Unix 或 Linux 平台上的开发。

DB2 通用驱动程序

Java 客户机应用程序需要“驱动程序”服务以访问服务器上的数据库资源。在 Version 8 之前，DB2 的三个不同产品家族提供了这些驱动程序（DB2 运行时客户机、DB2 应用开发客户机和 DB2 Connect），每种驱动程序分别针对特定的平台。这些驱动程序使用不同的内部协议，并需要在 Java 应用程序中进行不同处理，当将 Java 应用程序在平台之间进行移植时，需要修改代码，这是一项繁重的额外工作。

Version 8 引入了新的 Universal Driver，它使用与目标平台（本地或远程）无关的单一协议¹⁶和代码基。因为对于所有平台应用程序接口均相同，所以显著增强了 Java 应用程序的可移植性。使用新的驱动程序后，性能也得到很大改进（因为内存占用较小），并且还交付了许多有用的增强功能（参见下面的“Web 支持”）。

底线：新的 Universal Driver 提供了适用于所有平台（包括 DB2 for z/OS V7 和 V8）的单一驱动程序，从而在将应用程序移植到 zSeries 时无需额外的代码集更改，同时提供了有用的性能和功能增强。

¹⁵ 从 255 字节增加到 2,000 字节。

¹⁶ DRDA，它是 Open Group 的一项标准。

Web 支持

采用 Web 技术

电子商务为客户带来了许多机遇，因此越来越多的传统主机产品工作负载正向基于 Web 的方向迁移。但是，Web 技术的采用引入了其自己的挑战：

- **使用 Java。**Java 作为 zSeries 开发语言正越来越普及。IBM 在 DB2 中较早地引入了 Java 开发支持，Version 7 在这方面提供了健壮的功能集（例如通过 SQKJ 标准支持 Java 程序中的高性能静态 SQL）。但是，虽然 Version 7 具有这些优势，但在 Java 程序中的动态（JDBC）和静态（SQLJ）SQL 的性能方面还有待于大幅改进。
- **应用程序更改。**不管应用程序是使用哪种语言编写的，要在基于 Web 实现现有应用程序常常意味着必须重新编写或添加额外的代码。随着硬件和软件成本的下降，开发人员生产力在任何应用程序的总成本中成为关键因素，并且快速实现新代码变得越来越重要。
- **使用 XML。**近年来，XML 迅速成为互联网和公司内部网数据交换的事实标准，并成为许多基础 B2B 应用程序的关键技术。这些应用程序的公共需求是以 XML 格式发布 DB2 数据库中保存的数据。虽然 Version 7 提供了这项功能，但实现该解决方案要么需要大量的开发人员工作，要么会影响系统性能。
- **安全问题。**允许客户和/或供应商访问操作数据显然需要健壮的、可审计的访问控制机制，并且要具有限制数据库中特定内容的访问权限的灵活性。在 Version 7 中只能通过使用视图来实现此目的，但实现起来较为困难。

zSeries 平台的可伸缩性和弹性使其成为任务关键型按需应变应用程序的最理想选择。在 Version 8 中，IBM 继续关注 Web 支持功能，扩展了 DB2 的 Java 和 XML 功能，同时巩固了其对于大容量电子商务引擎的领先地位。

Java 增强

前面介绍的新 DB2 Universal Driver 除提高可移植性之外，还交付了许多有用的增强功能。该驱动程序与 JDBC 3.0 行业标准完全兼容，并引入了大量有价值的新功能，包括：

- **滚动光标支持。**允许应用程序开发人员在查询结果集中前后导航，而无需对程序代码重新排序，从而提高了开发人员生产力和应用程序性能。
- **批处理更新。**此功能允许应用程序对多个数据库更新操作进行分组，然后在单次网络交换中向 DB2 提交。对于特定的应用程序来说，这能够实现极大的性能改进。

- **增强的 SQL 错误消息。** 标准错误处理的扩展程序允许将更具体的 SQL 错误信息传递给应用程序。这允许开发人员不必编写额外的代码来确定错误信息，从而提高了生产力。
- **Java 客户机信息。** Java API 的扩展程序允许客户机将用户定义的信息传递给后端 DB2 服务器。这允许在服务器上为独立的客户机分配特定的执行优先级，从而改进工作负载管理并允许业务关键型事务具有更高的吞吐量。
- **存储点支持。** 存储点 (savepoint) 允许应用程序请求其本身定义的一致性点 (consistency point)，从而能够及时地将数据恢复到较早的点。此功能使开发人员不必手工编写代码，因此提高了生产力。

底线： DB2 Universal Driver 中新的 Java 功能提供了一些有价值的生产力和性能增强，整合了主要的 Java zSeries 编程语言。

XML 增强

XML 正逐渐成为 IT 系统内、外部信息交换的重要事实标准。IT 系统的公共需求是以 XML 格式表示 DB2 数据库中保存的数据（此过程称为 XML 发布）。

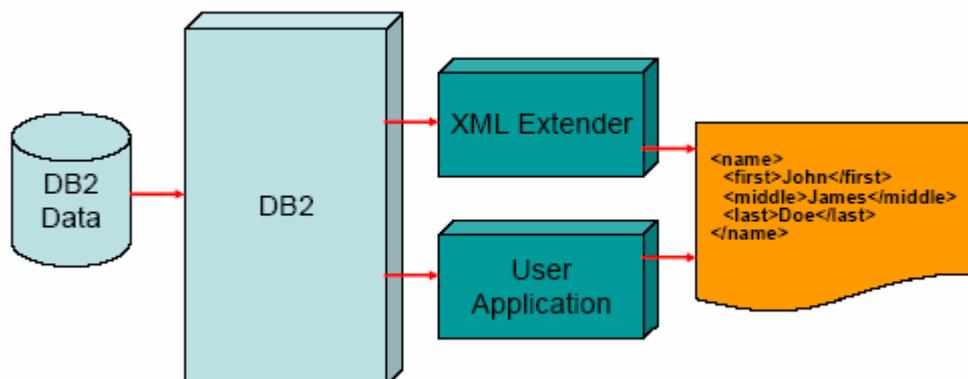


图 9 Version 7 XML 发布架构

Version 7 提供了 XML Extender 来实现此功能，但与其他 User Defined Function 不同，它在与主 DB2 代码分离的区域中运行，因此与 DB2 内部功能相比效率较低（参见上面图 9）。

Version 8 实现了作为内部 SQL 功能的完全的 XML 发布功能集合，如图 10 所示。此实现允许标准 SQL 语句以高性能返回本机 XML 数据流。

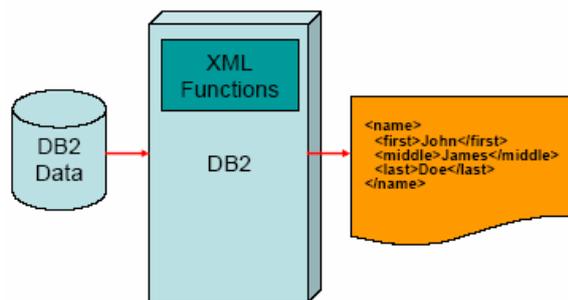


图 10 Version 8 XML 发布架构

底线：虽然我们仍需要使用 XML Extender 来分解和存储 DB2 中的 XML 数据，但 Version 8 内置的 XML 发布功能提供了一些友好的性能改进，并为将来的广泛的本机 XML 支持指出了方向。

提高性能并降低操作成本

不断提高性能是所有关系数据库客户的重要目标。过去实现此目标主要是为了满足不断提高的服务水平，而现在的主要目标是“以更少资源完成更多工作”和利用给定硬件投资支持更大的工作负载（从而降低总体操作成本）。

DB2 的每个版本均交付了能够降低 DB2 事务资源要求的显著性能增强，Version 8 同样也提供了这样的增强功能。下面部分介绍新版本所交付的与性能有关的主要功能。这些功能是 Version 8 的重要功能，它们将补偿与迁移到 64 位环境有关的额外 CPU 开销（有关更多信息请参阅第 15 页的“64 位支持”）。

多行插入和读取（ Insert&Fetch）

插入和读取是 DB2 表的最常用的两个操作。通常，程序循环中要多次执行这两个操作，当应用程序要向数据库添加多行或执行多行检索时，每次只能执行一行操作。

Version 8 首次引入了在单次操作中读取或插入多行的功能。当执行插入操作时，开发人员只需简单地将所有要插入的行存储到程序存储器中，然后将它们作为一个组传递到 DB2。同样，多行读取操作只需指定程序存储器区域和要检索的行数，然后 DB2 可以通过单次操作将要检索的行存储到存储器中。

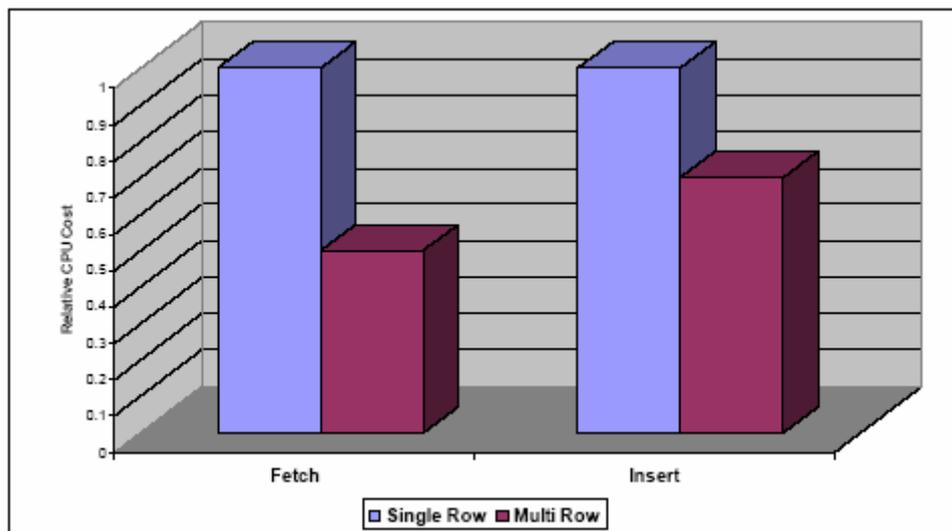


图 11 多行插入/读取能够节省 CPU 资源

如图 11 所示，IBM 最初的测试显示了在非分布式应用程序中使用此功能的获益，对于读取和插入操作分别节省 50% 和 30% CPU 资源。Version 8 的早期客户验证了这一结果，他们通常能够节约 30-40% CPU 资源。在分布式环境中由于避免了网络延迟而使资源的节约更为突出，其中延迟时间最高缩短 8 倍，CPU 节省 4 倍。

这些增强功能除应用于为客户编写的应用程序之外，已经集成到 IBM 提供的实用程序和示例代码中，同时也可用于分布式数据访问。

底线：多行插入和读取命令是一项显著的性能改进，它能够潜在地使许多类型的应用程序获益，其中对于某些操作可节省 50% CPU 资源。对于分布式应用程序来说，可以用单次调用替代耗时的多次网络传输，因此获得更大的性能改进。

优化增强

优化器是负责选择以最高效的方式执行给定 SQL 语句的组件。DB2 的行业领先的优化器在 Version 8 中提供了很多方面的增强功能，包括：

- **不一致的数据类型处理。**开发人员经常遇到的传统的 DB2 性能“陷阱”是 SQL 中的不一致的数据类型。例如，在 SQL 语句中可能指定了小数，而将其与表中的整数值进行比较。DB2 可以执行此 SQL 语句，但由于转换开销的原因且 DB2 不能高效使用这类查询的索引，将导致性能下降。虽然可以在编写 SQL 查询代码时采用一些良好实践来避免此问题，但并非总是有效（例如，Java 不支持固定长度的字符串，而 DB2 表中通常需要使用这样的数据类型）。Version 8 对这样的不匹配具有更强的处理能力，允许更高效地处理这样的查询。

- **增强的分布式统计。**Version 8 中增强了 DB2 收集数据统计信息的能力。这允许 DB2 确定最高效的数据访问方式，从而能够改进查询性能。
- **并行排序。**在 Version 8 之前，DB2 尽可能尝试进行并行排序。在某些情况下（例如小的排序），并行排序并不是高效的方法。Version 8 优化器能够根据要排序的数据量做出更明智的决策，确定是否进行并行排序。这能够提高特定查询的 CPU 利用率。

底线：DB2 行业领先的 SQL 优化器在 Version 8 中得到进一步增强，从而为许多公共查询提供了更高的效率和更低的 CPU 成本。

物化查询表（MQT）

在决策支持/数据仓库环境中，公共的查询需求是在更高的水平上总结表中的详细信息。例如，零售公司在对包含详细销售数据进行查询时，可能需要按地区进行销售汇总。

这样的汇总查询可能需要很大开销，因为查询涉及表中的大量详细数据。执行这类查询的频率越高，相应的成本也越高。

MQT 针对这个问题的解决办法是只构建一次汇总结果表，并将其存储在一个特殊的表中，即物化查询表（MQT）。当用户下次对详细表进行查询时，MQT 可提高查询效率，DB2 将在内部自动重写查询，并根据 MQT 来执行查询。查询过程中无需用户干预，并且响应时间可以缩短 10 - 1,000 倍。

底线：MQT 已经在 DB2 for Linux、Unix and Windows 中发挥了强大功效，它是大幅度缩短查询响应时间和减少资源消耗的有力方式，而无需查询用户付出任何工作。它是一项可用于决策支持/数据仓库环境的有价值的特性。

其他性能增强

Version 8 还引入了其他很多性能增强功能，包括：

- **分区增强。**独立指定分区属性和数据的物理排序（参见第 23 页的“分区增强”），这对于特定查询可实现极大的性能提高（在某些测试中最高可节省 20 倍 CPU 资源）。
- **双向索引。**在 Version 8 之前，DB2 只能向前读取索引。在某些情况下表中可能需要定义两个相近的相同索引，只是索引关键字的存储顺序不同。Version 8 允许清除这些多余的索引，因为 Version 8 允许向前和向后两个方向读取索引。这节省了 DASD 空间并避免了维护这些多余索引（插入、更新和删除操作）所带来的 CPU 和延迟时间开销。

- **数据共享锁定。**这是一项在数据共享环境中处理锁定的增强功能，它允许 Version 8 本机系统对锁定请求进行管理，而不必将请求传递给 Coupling Facility。经过 IBM 测试，这项增强功能在 IBM 的 Relational Warehouse Workload 的事务中可节省 6% CPU 时间¹⁷。
- **增强的动态语句缓存。**动态语句缓存允许 DB2 记住为以前执行的动态 SQL 语句选择的访问路径。内存分配方式的改进使事务吞吐量得到很大提高，在一些 IBM 测试中高达 45%。
- **页面修复缓冲池。**缓冲池是 DB2 的主要数据缓存，Version 8 引入了一些增强功能，它们允许 DB2 更高效地利用存储器。测试表明，对于标准的 IBM Relational Warehouse Workload，这些增强功能可减少 7% 总体 CPU 时间。

底线：这些性能增强在很大程度上减少了 CPU 需求并增大了客户工作负载的吞吐量。

开发人员/DBA 生产力

随着硬件和软件成本的不断下降，IT 人员成本相对变得越来越高。提高生产力并让相同人员管理更多系统成为关键因素，这决定了 IT 系统是否能够持续交付业务价值。

IBM 一直致力于提供额外的功能，这些功能旨在提高 DB2 应用程序开发人员和数据库管理员的生产力。Version 8 的新的生产力特性包括：

- **动态模式更改。**动态模式更改（参见第 21 页）极大地减少了更改 DB2 对象结构所需的工作。在具有大量应用程序更改的环境中，这项增强功能允许 DBA 支持和管理比 Version 7 系统中更多的数据库。
- **自动二次空间分配。**空间管理作为监控和手工调整表所使用的磁盘空间的过程，是数据库管理员日常工作的一项重要内容。这项增强功能允许 DB2 自动处理接近达到最大容量的表。虽然这并未完全使管理员脱离空间管理工作，但它显著减少了在管理繁忙的开发或产品环境中所需的时间。在较少涉及前瞻性空间管理策略的环境中，此特性还能减少由“空间溢出”引起的应用程序失败和数据不可用性。
- **分区平衡。**数据重组实用程序中的新功能允许管理员通过单条命令将数据均匀分配到每个表分区中。新版本之前这需要管理员的大量手工工作。

¹⁷ 使用双向数据共享

- **智能实用工具默认功能。**DB2 数据整理实用程序提供了更多的智能默认值，数据库管理员只需更少的手工调优即可定义高效的过程。
- **系统级备份和恢复。**此功能（参见第 24 页）允许使用很少的命令来备份和恢复整个 DB2 系统。除系统和灾难恢复这些明显的优势外，这些功能还可以提高开发环境中的生产力，此外可用于快速克隆开发系统，从而节省大量的工作时间。
- **递归 SQL。**许多应用程序要求对给定的表执行递归 SQL 语句（通常称为“bill of material”）。在新版本之前，只有通过相对复杂的程序编码和/或数据库设计才能满足这些需求，因此需要大量的开发人员和管理员工作来构建和维护。Version 8 引入了递归 SQL 支持，允许直接由 DB2 来处理这些查询。
- **Sequence。**当在表中生成一个新行时，通常要求应用程序生成一个充当关键字的惟一数字（例如发票号）。虽然 Version 7 的 Identity Column 支持这项需求，但在实现过程中产生的一系列操作问题限制了它的使用。Version 8 引入了 Sequence，它是一个非常灵活的选项，允许自动生成惟一数字，因此极大地提高了开发人员的生产力。

除这些新功能之外，新版本还广泛增强了 IBM 的 DB2 Tool¹⁸，以促进新版本的采用，并进一步提高数据库管理员和开发人员的生产力，从而让他们能够关注具有更高价值的任务，并帮助公司提高数据中心的投资回报。

底线：Version 8 引入了大量重要的新功能，这些功能让 DB2 管理员和应用程序开发人员能够提高日常工作效率。随着目前人员成本所占比例不断提高，这些增强功能对于向 Version 8 的迁移的成本方面成为极为重要的因素。

全球集成和部署

按需应变的企业应突破国界限制，在当今的全球化经济中，越来越多的 DB2 客户使用面向全球的 IT 系统。这种全球化引入了由于地区差异而导致的可用性挑战，此外分布于各个不同地域的客户机系统还面临更重要的字符表示方式问题。

¹⁸ 一个需要单独购买的综合工具集，它用于补充特殊的 DB2 需求（例如非常大型的企业和复杂的配置），有关更详细的信息，请参考 <http://www.ibm.com/software/data/db2imstools/>

从传统上讲，每种语言环境都具有其自己的字符编码方式，字符编码后成为可以存储于计算机中的数字表示形式。这些编码模式（称为代码页）常常相互冲突，特别是对于“非标准”字符。例如，使用标准的 US 代码页，192 表示“{”符号，但在标准的 German 代码页中，相同的数字表示“ä”符号。这些冲突通常导致在未经正确配置的 3270 客户机上显示错误的结果。我们可以选择在所有客户机上强制使用单一的代码页来解决此问题，但这样本地用户不能访问其自己的本国语言字符。

为此人们引入了 Unicode 来解决这一问题，通过提供每个字符与平台和语言无关的表示形式，可以避免上述冲突。此标准的最新版本（Version 4.0）包含 96,248 个不同字符，这个数字还会继续增加。

Version 7 引入了最初的 Unicode 支持，允许以 Unicode 格式存储表数据，而不使用传统的主机 EBCDIC 编码模式。这允许多个国家的客户在相同的表中存储来自不同地区的数据，但 Version 7 仍存在一些挑战（例如根据 Unicode 表执行 SQL 语句的复杂性，以及不能将 Unicode 表连接到使用不同编码模式的其他表）。

Version 8 提供了对 DB2 Version 7 最初实现的 Unicode 支持的增强功能，排除了该版本中的许多内在约束。

新版本的变更包括：

- **通过单条 SQL 语句连接多种编码模式。** Version 8 允许通过单条 SQL 语句来连接具有不同编码模式的表。这允许为每个表选择能达到最佳性能的编码模式，而不会影响现有应用程序的功能。
- **以 Unicode 格式处理 SQL。** 这项增强功能允许 SQL 语句本身包含 Unicode 引用，在处理之前 SQL 会在内部转换为 Unicode。
- **Unicode 格式的 DB2 目录。** 迁移到 Version 8 后，由于 SQL 语句可以在处理之前转换为 Unicode，因此 DB2 内容表中的大部分信息可以转换为 Unicode。

底线： Version 8 是基于以前版本所实现的初始 Unicode 支持构建的，删除了许多妨碍客户完全利用 Unicode 跨国应用程序环境的约束。

数据安全

数据是机构的宝贵资产，因此必须将保护数据安全性作为机构的最重要任务之一。DB2 一直具有非常健壮的安全实现，通常能够很好地满足传统 OLTP 工作负载的安全需求。

但是，当今的许多按需应变的应用程序需要更灵活的方法，尤其是需要限制对表中特定数据的访问。DB2 传统的安全机制已不能直接满足此需求，虽然可以采取一些措施缓解此问题，但它们都会带来一定程度上的性能影响。

Version 8 全面改进了 DB2 的安全方法，提供了实现“多级安全”的选项，可用于替代传统的 DB2 方法。多级安全基于 RACF V1.5 中的功能，它提供了在任何所需级别保护 DB2 资源的灵活性，从整个表到个别的行。新选项还提供了高级的分层安全功能。例如，可以简单地定义工资表的访问控制，使员工只能查看自己的数据，而经理（安全层次结构中的上一层）能够查看该部门的所有数据。

底线：Version 8 的行级安全特性从根本上改变了 DB2 访问权限的管理方式。实现这些特性需要进行仔细的计划和析，但客户最终能够获得显著提高的灵活性，从而在安全管理和数据访问方面实现更高级别的控制和功能。

Version 8 业务获益总结

本小节总结了 Version 8 提供的主要业务获益。注意不是每位客户都会获得这些收益，但下表所列的业务获益在许多方面会带来重大改进。

可伸缩性和 zseries 增效作用

特性	业务获益	页码
64 位支持	Version 8 允许 IBM 最新一代硬件的客户完全利用 64 位存储器模型的庞大存储容量，而无需更改应用程序，进而允许 DB2 系统缓存更多数据并直接跃升到更高的水平，从而减少系统开销并更高效地使用有限的机器资源。在某些情况下，这些增强功能允许客户延迟仅与可伸缩性有关的数据共享的实现。虽然迁移到 64 位环境可能对 CPU 有更高的要求，但新版本显著的性能和生产力提高马上会抵偿 CPU 的这部分开销。	15
分区增强	Version 8 为数据库设计人员提供了功能更强的分区选项，允许在给定的机器资源条件下创建支持更高事务负载的高效 DB2 数据库设计。借助于新的动态模式更改功能，可以在现有数据库设计基础上实现这些获益，并最大限度地减少对可用性的影响。	18
恢复日志增强	Version 8 允许在线和离线保持更多恢复日志数据，从而允许 DB2 系统处理更大的工作负载，同时保持数据的可恢复性。	19
其他 zSeries 增效作用 (synergy)	Version 8 采用了 z/OS 和 zSeries 硬件中的特定功能，以提供更高效率的字符转换和加密服务。	20

数据可用性

特性	业务获益	页码
动态模式更改	Version 8 提供了完善的初始化功能集，能够确保在更改数据库结构的同时保证应用程序可以访问数据。客户可以利用这些功能实现更频繁的数据库结构更改，以便对不断变更的市场条件或意外性能问题做出迅速响应。由于更改过程更加简单，可以极大地提高数据库管理员的效率（同时显著降低了与数据库更改有关的风险）。	21
分区增强	Version 8 引入的全新分区模型首次实现了真正的独立分区。这项特性减少了表重组所产生的可用性影响，允许管理员更好地组织数据，因此从根本上提高了应用程序的性能。	23
动态系统参数	Version 8 中额外的动态系统参数降低了整个 DB2 系统在配置参数更改时可能遇到的风险，从而提高了整体可用性，并允许系统管理员对不断变更的应用程序要求做出更快的响应。	23
系统级备份和恢复	在拥有正确的 DASD 子系统后，新的系统级备份和恢复功能能够减少由系统备份造成的 DB2 数据不可用的时间。更重要的是，及时将系统快速恢复到给定点意味着系统在发生重要问题后能够以最短时间恢复可用性。	24

应用程序移植

特性	业务获益	页码
SQL 增强	这些 SQL 增强能够降低将数据库定义和应用程序代码移植到 DB2 Version 8 所需的工作量、复杂度和成本，使其成为一个更具吸引力的平台，可用于供应商应用程序移植、基础设施整合项目和 Windows、Unix 或 Linux 平台上的开发。	25
DB2 Universal Driver	新的 Universal Driver 提供了适用于所有平台（包括 DB2 for z/OS V7 和 V8）的单一驱动程序，从而在将应用程序移植到 zSeries 时无需额外的代码集更改，同时提供了有用的性能和功能增强。	26

Web 支持

特性	业务获益	页码
Java 增强	DB2 Universal Driver 中新的 Java 功能提供了一些有价值的生产力和性能增强，整合了主要的 Java zSeries 编程语言。	27
XML 增强	虽然我们仍需要使用 XML Extender 来分解和存储 DB2 中的 XML 数据，但 Version 8 内置的 XML 发布功能提供了一些友好的性能改进，并为将来的广泛的本机 XML 支持指出了方向。	28

性能和操作成本

特性	业务获益	页码
多行读取和插入 (Fetch & Insert)	多行插入和读取命令是一项显著的性能改进，它能够潜在地使许多类型的应用程序获益，其中对于某些操作可节省 50% CPU 资源。对于分布式应用程序来说，可以用单次调用替代耗时的多次网络传输，因此获得更大的性能改进。	29
优化增强	DB2 领先的 SQL 优化器在 Version 8 中得到进一步增强，从而为许多公共查询提供了更高的效率和更低的 CPU 成本。	30
物化查询表 (MQT)	MQT 已经在 DB2 for Linux、Unix and Windows 中发挥了强大功效，它是大幅度缩短查询响应时间和减少资源消耗的有力方式，而无需查询用户付出任何工作。它是一项可用于决策支持/数据仓库环境的有价值的特性。	31
其他性能增强	这些性能增强在很大程度上减少了 CPU 需求并增大了客户工作负载的吞吐量。	31

开发人员/数据库管理员生产力

特性	业务获益	页码
生产力增强	Version 8 引入了大量重要的新功能，这些功能让 DB2 管理员和应用程序开发人员能够提高日常工作效率。随着目前人员成本所占比例不断提高，这些增强功能对于向 Version 8 的迁移的成本方面成为极为重要的因素。	32

全球集成和部署

特性	业务获益	页码
Unicode 支持	Version 8 是基于以前版本所实现的初始 Unicode 支持构建的，删除了许多妨碍客户完全利用 Unicode 跨国应用程序环境的约束。	33

数据安全

特性	业务获益	页码
行级安全性	Version 8 的行级安全特性从根本上改变了 DB2 访问权限的管理方式。实现这些特性需要进行仔细的计划和分折，但客户最终能够获得显著提高的灵活性，从而在安全管理和数据访问方面实现更高级别的控制和功能。	34

DB2 for z/OS Version 8 客户案例研究：Univar USA

Univar USA 是美国一家领先的化学销售商，在美国的化学及相关销售服务市场占据最大的市场份额。

一直以来，Univar 的 IT 环境基于多种技术的组合：多台 Unix 服务器运行 Oracle 应用程序并提供了业务信息环境，同时由一台 IBM 9672 RB6 主机运行多个传统的 VSAM 应用程序（参见图 12）。

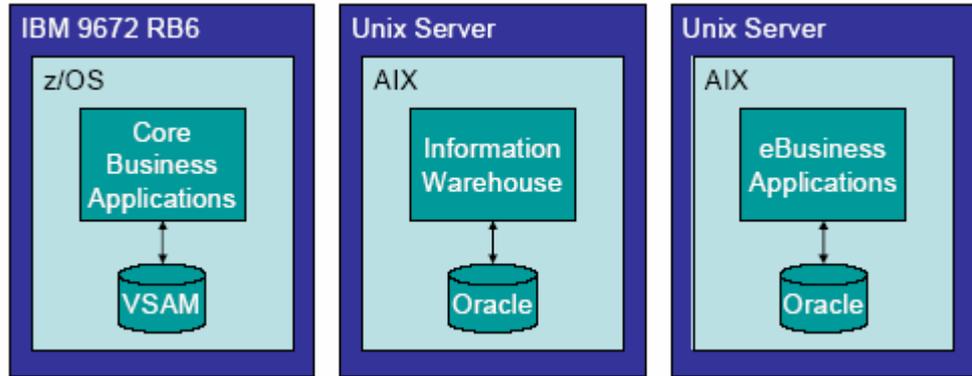


图 12 以前的 Univar IT 环境

Version 8 迁移项目的主要推动者，Univar 应用架构师 Kevin Campbell 指出：“多年以来我们一直希望将现有 VSAM 转换为使用 DB2。我们的主要目的是为了提高可用性并能够轻松地与外部应用程序进行集成，这对于我们来说是当务之急的问题，但投资回报却成为一个难题，所节省的成本可能无法抵偿 DB2 许可证费用。”

就 Unix 方面来说，Univar 很早就认识到 Unix 主机环境提供了超级的系统管理功能，并且平均事务成本低于其他平台，但重写 Unix/Oracle 应用程序和将它们移植到 zSeries 平台的成本成为迁移的最大阻力。“我们拥有在 Oracle 上运行的重要的、成熟的信息仓库，它的迁移计划总是搁浅，因为很难对所有 OLAP 基础设施进行更改，让它们适应更短的列名称。每年创造数百万美元收入的电子商务应用程序也在使用 Oracle，这也成为移植的一大障碍。”

DB2 for z/OS Version 8 扫清了所有障碍。Version 8 支持更长的列名称和更大的 SQL 语句，它能够极大地降低 Univar 的应用程序移植的成本和风险，为公司在 zSeries 平台上进行大规模服务器整合项目创造了机遇。同时，主机上现有的 DB2 允许对 VSAM 应用程序进行转换，并且能够实现关键的业务线应用程序的集成和可用性改进，这正是公司所必需的。

Kevin 指出：“DB2 Version 8 为我们创造了绝佳的机遇：我们可以获得核心事务处理系统的长期可用性和稳定性，并可以通过将 Oracle 应用程序迁移到 DB2 for z/OS 实现‘硬性成本’节约。在此过程中我们还能从根本上流线化和简化计算环境，并且将主机的简单可恢复性推广到更大范围。Version 8 的更长的列名称支持对我们来说无疑是最有价值的特性，在将应用程序移植到主机时这将显著减少相关工作并降低风险，允许我们将多个 Unix 系统整合到单一的 z/Linux 环境中（参见图 13）”。

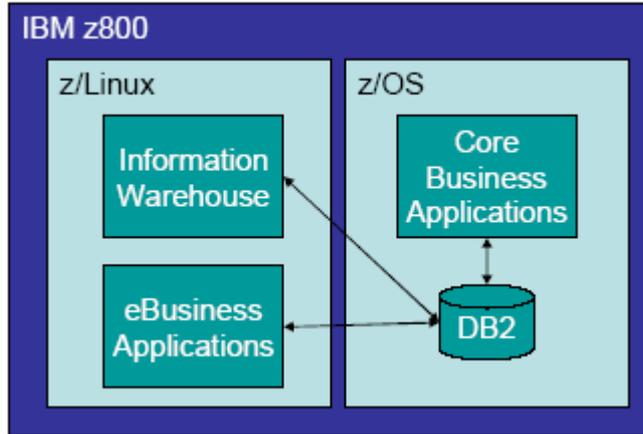


图 13 Univar 的新的 IT 环境

为评估迁移的可行性，Univar 参加了 IBM 的 Version 8 前期发布计划。Kevin 说：“这使我们获得了非常宝贵的 Version 8 经验，并且验证了 Unix 应用程序移植的可行性”。Univar 在前期发布计划所获得的经验起到了积极推动作用，并促使公司做出向 DB2 Version 8 迁移的决策。

公司于 2004 年 4 月开始了迁移过程，首先是将主机升级到 64 位容量的 eServer z800。Kevin 对升级过程给予了高度评价：“在我们的技术服务团队的控制下，整个升级过程只用了几小时，几乎未对用户产生影响，他们只是在第二天发现响应时间变短了！”

硬件升级完成之后，Univar 的技术服务团队在应用程序相关工作就绪后开始构建新的环境。Beta 版系统仅使用 8 天，就被最终的 Version 8 代码取代，这使得 Univar 成为最先正式采用新版本的客户之一。在“DB2 lab”的帮助下，仅用 2 天时间系统即安装就绪。

然后一支由 25 位开发人员组成的团队开始将 Univar 应用程序转换为使用 DB2，其中由 3 人组成一个 DB2 管理小组，负责将 VSAM 文件迁移到 DB2 for z/OS 数据库。在撰写本文时，Univar 正处于将转换后的应用程序迁移到产品环境（预计在 2004 年 9 月）之前的压力测试阶段。

预计到 2004 年底，迁移将进入最后一个步骤，即将 Unix 应用程序从 Oracle 移植到 DB2，以实现许可证和基础设施的成本节约。Kevin 坚信这将对 DB2 管理团队产生深远影响。他指出：“迁移到 DB2 能够将数据库管理员从低级 Unix 系统管理活动（例如手工备份）中解放出来。他们将有更多时间高效地利用他们的技能，将更多精力投入优化和 SQL 调优这样的工作中，从而为机构创造更大价值。”

除应用程序移植增强功能之外，Version 8 还具有很多其他优势，其中动态模式更改功能是一项关键特性。许多要移植的应用程序是电子商务系统，这些系统不允许有计划内停机，并且要求能够动态更改数据库结构。虽然 Univar 是一家跨国机构，但每个国家均使用其自己的 IT 系统，因此 Unicode 支持不是主要问题。Kevin 说：“但是，我们希望在这些系统之间进行数据整合，以便将来可以进行报告和分析。因此，Unicode 支持对我们来说是很有价值的。”

因此迁移到 Version 8 对于 Univar 具有重要的技术意义，但在成本方面是什么情况呢？Kevin 表示：“对于我们来说，迁移到 DB2 Version 8 的决策首先是一个财务决策。迁移可以带来三个方面的显著成本节约。首先，我们可以不必再支付 Unix 服务器的 Oracle 许可证费用和基础设施支持费用。第二，在 Unix 服务器到期之后，我们不必再续交租赁费。第三，我们从根本上减少了填充信息仓库的相关工作和成本，因为这将成为一个 DB2 到 DB2 的过程，而不再是一个 z/OS 上的 VSAM 到 Unix 上的 Oracle 过程。综合来看，这些成本节约将迅速补偿 DB2 的迁移投资。

验证升级

本小节将给出一些示例场景，以证明从 Version 7 迁移到 Version 8 的成本/获益情况。我们重点关注与升级有关的可量化的财务成本和获益，而不讨论前面已经描述的许多战略性或非量化的获益。

在 Version 8 迁移的主要客户类型中，我们选择了 3 种典型客户：

- OLTP 客户，他们运行传统的 CICS/DB2 工作负载
- SAP 客户，他们仅使用 DB2 for z/OS 作为 ERP 工作负载的后台数据库服务器
- Java 客户，他们运行高容量的基于 Web 的事务工作负载

以下部分对这些客户进行了详细定义，给出了每类客户的成本和获益估算。

注意，虽然这里使用的是实际的硬件和软件成本，但这些数据是在许多因素和假设的基础上概括得到的，对于不同的客户这些因素和假设可能大不相同。但这里所呈现的模型在分析潜在的财务状况方面，对于许多客户具有参考价值。

示例客户环境

为了使示例具有代表性，我们选择了运行 eServer z990¹⁹ 的客户，其运行速度约为 330 MSU²⁰（例如 Model 2064-2C8），其运行平台为 z/OS 1.4。该系统的总运行成本（包括硬件、操作系统、中间件和 ISV 软件）为 5,573,000 美元/年。

目前客户每年支付 410,000 美元 DB2 Version 7 的 PSLC²¹ 许可证费用（包含在上图所示的拥有成本中），升级到 Version 8 后每年还需支付额外的 61,000 美元。

此外，由于迁移到 64 位环境需要 7% 的 CPU 增加，这项额外的成本每年为 387,000 美元²²。

升级所需的人员成本估算如下：

- 规划（每客户）— 20 天系统程序员工作，20 天 DBA 工作，5 天应用程序开发人员工作。
- 实现（每 DB2 系统）— 5 天系统程序员工作，10 天 DBA 工作，10 天开发人员工作

在计算项目成本和获益时，所有人员按 60 美元/小时计算。

¹⁹ 如果向 z990 平台的迁移是整个迁移的一部分，那么客户还应考虑到额外的“technology dividend”获益，这是 IBM 在 2003 年 9 月公布的一项获益，这里并未将此获益计算在内，因为我们假设客户在向 Version 8 迁移之前已经使用了 z990 平台。

²⁰ MSU — Million Service Unit。用于 IBM S/390 和 zSeries 系统的性能/容量比。

²¹ PLAC — Parallel Sysplex Licence Charge

²² 计算方法是总体拥有成本（5,573,000 美元）的 7% 减去磁盘存储成本（46000 美元），因为磁盘存储不受迁移影响。

场景 1: OLTP 客户

在场景 1 中我们考虑运行传统 OLTP 工作负载的客户，日间在线工作负载由高容量 CICS/DB2 事务组成，每天夜间从 8.00pm - 6.00am 采用批处理窗口。

客户运行 3 个独立的 DB2 系统，其中包括两个开发系统和一个位于独立逻辑分区上的产品系统。除 OLTP 工作负载外，产品系统还运行一个小型的管理信息系统，使用从主操作数据库中提取的目录。

客户的升级成本如下表所示：

	第 1 年	第 2 年	第 3 年
额外的 DB2 许可证成本	\$61,500	\$61,500	\$61,500
CPU 成本 — 性能衰退	\$386,890	\$386,890	\$386,890
人员成本 — 升级	\$57,600	\$0	\$0
总成本	\$505,990	\$448,390	\$448,390

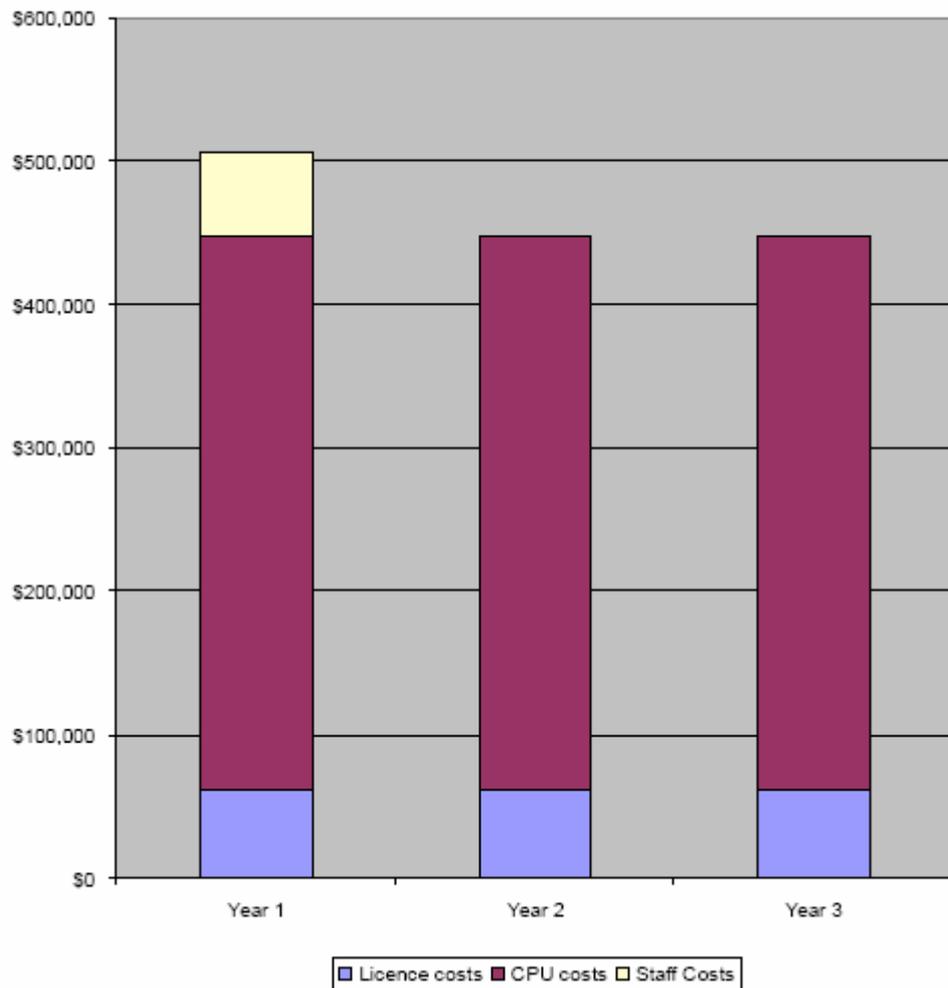


图 14 OLTP 客户升级成本

注

- 无硬件成本，因为客户已拥有 64 位容量的服务器。
- DB2 许可证和 64 位衰退 CPU 成本按第 43 页“示例客户环境”所示计算。
- 人员成本按第 43 页“示例客户环境”所示计算。

客户的财务获益如下表所示：

	第 1 年	第 2 年	第 3 年
可伸缩性	\$167,190	\$167,190	\$167,190
多行插入/读取	\$111,460	\$111,460	\$111,460
物化查询表	\$55,730	\$55,730	\$55,730
优化改进	\$111,460	\$111,460	\$111,460
人员成本 — 动态模式更改	\$21,600	\$21,600	\$21,600
人员成本 — PIT 恢复	\$2,880	\$2,880	\$2,880
人员成本 — 自动空间管理	\$14,400	\$14,400	\$14,400
人员成本 — 实用程序增强	\$3,600	\$3,600	\$3,600
总计	\$488,320	\$488,320	\$488,320

注

- 采用 64 位架构内存后由于吞吐量增大和效率提高获得的可伸缩性成本节约约占总体拥有成本的 3%。
- 多行插入/读取允许使用更少的单独指令并减小了远程操作的网络开销，这项成本节约约占总体拥有成本的 2%。
- 物化查询表能够避免在管理信息环境中对大容量表进行扫描，这项成本节约约占总体拥有成本的 1%。
- 优化允许 DB2 为现有应用程序选择更高效的访问路径，这项成本节约约占总体拥有成本的 2%。
- 人员成本节约估算如下：
 - 动态模式更改 — 由于能够更快实现数据库结构的更改，每月节约 30 小时（10 项更改 x 3 小时/项）。
 - 时间点恢复 — 由于备份/恢复过程更快更简单，每月节约 4 小时（2 次恢复 x 2 小时/次）。
 - 自动空间管理 — 不必再修复常规的空间溢出条件，每月节约 20 小时（20 天 x 1 小时/天）。

- 实用程序增强 — 由于具有更智能的默认设置，实用程序调优每月节约 5 小时（20 天 x 0.25 小时/天）。
- 这里未计算预期的数据可用性提高带来的获益，但对于某些客户来说这是一项可量化的财务获益。
- 这里未计算应用程序移植能力提高带来的获益，某些客户在 Windows、Unix 或 Linux 平台上进行应用程序开发，而产品在 DB2 for z/OS 运行，对于这些客户来说可能获得显著的财务获益。

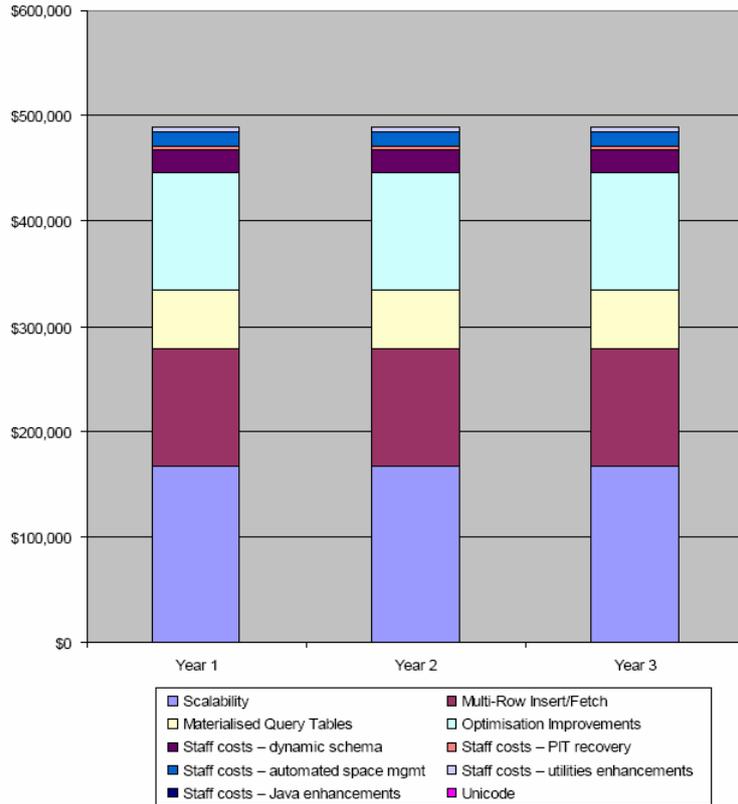


图 15 OLTP 客户财务获益

投资回报

以上所述成本和获益可在 18 个月内实现投资回报，如下表所示：

	第 1 年	第 2 年	第 3 年
成本	\$505,990	\$448,390	\$448,390
获益	\$488,320	\$488,320	\$488,320
节约	-\$17,670	\$39,930	\$39,930
总计	-\$17,670	\$22,260	\$62,190

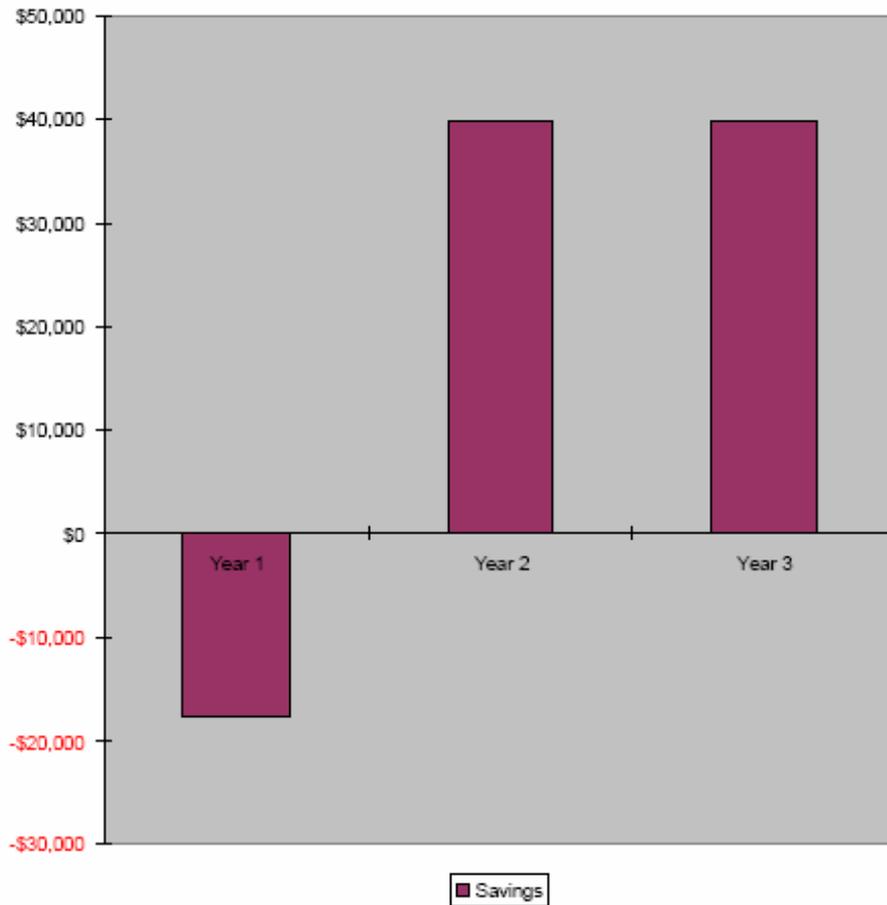


图 16 OLTP客户总成本节约

场景 2: SAP 客户

在场景 2 中我们看一下单纯使用 DB2 来支持 zSeries SAP 系统的客户。该客户使用 SAP 业务信息仓库功能来分析关键业务信息。

客户运行 4 个单独的 DB2 系统，其中包括两个开发系统和两个位于独立逻辑分区上的产品系统，分别用于支持 SAP 操作和业务信息仓库工作负载。

客户的升级成本如下表所示：

	第 1 年	第 2 年	第 3 年
额外的 DB2 许可证成本	\$61,500	\$61,500	\$61,500
CPU 成本 — 性能衰退	\$386,890	\$386,890	\$386,890
人员成本 — 升级	\$69,600	\$0	\$0
总计	\$517,990	\$448,390	\$448,390

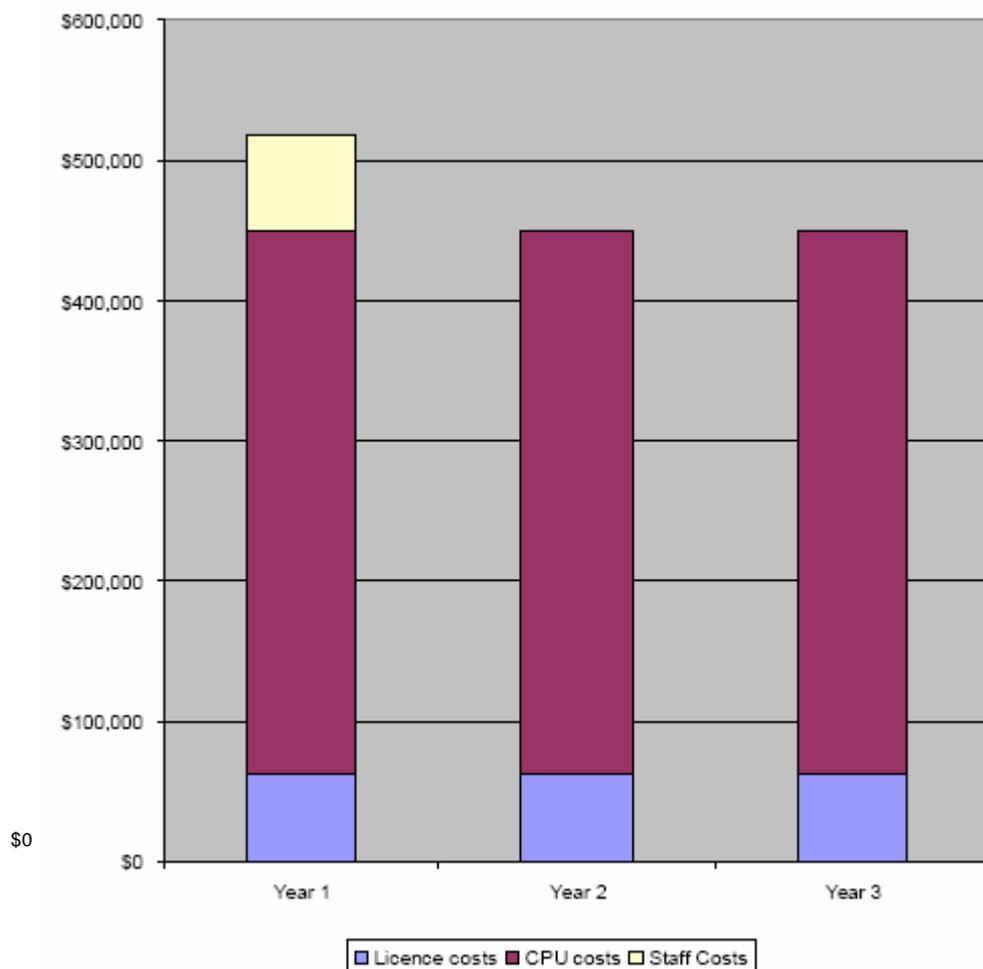


图 17 SAP 客户升级成本

注

- 无硬件成本，因为客户已拥有 64 位容量的服务器。
- DB2 许可证和 64 位衰退 CPU 成本按第 43 页“示例客户环境”所示计算。
- 人员成本按第 43 页“示例客户环境”所示计算。

客户的财务获益如下表所示：

	第 1 年	第 2 年	第 3 年
可伸缩性	\$278,650	\$278,650	\$278,650
多行插入/读取	\$83,595	\$83,595	\$83,595
物化查询表	\$5,573	\$5,573	\$5,573
优化改进	\$111,460	\$111,460	\$111,460
人员成本 — 动态模式更改	\$5,760	\$5,760	\$5,760
人员成本 — PIT 恢复	\$14,400	\$14,400	\$14,400
人员成本 — 自动空间管理	\$21,600	\$21,600	\$21,600
人员成本 — 实用程序增强	\$3,600	\$3,600	\$3,600
总计	\$524,638	\$524,638	\$524,638

注

- 采用 64 位架构内存后由于吞吐量增大和效率提高获得的可伸缩性成本节约约占总体拥有成本的 5%。
- 多行插入/读取允许使用更少的单独指令并减小了远程操作的网络开销，这项成本节约约占总体拥有成本的 1.5%。
- 物化查询表能够避免在管理信息环境中对大容量表进行扫描，这项成本节约约占总体拥有成本的 1%²³。
- 优化允许 DB2 为现有应用程序选择更高效的访问路径，这项成本节约约占总体拥有成本的 2%。
- 人员成本节约估算如下：
 - 动态模式更改 — 由于能够更快实现数据库结构的更改，每月节约 8 小时（2 项更改 x 4 小时/项）。
 - 时间点恢复 — 由于备份/恢复过程更快更简单，每月节约 20 小时（10 次恢复 x 2 小时/次）。

²³ SAP 业务信息仓库已经实现了聚合表的概念，因此在此环境中 MQT 可能不会产生明显的获益。但因为不能在主 SAP 操作数据存储上定义聚合，因此实现 MQT 将产生有限的获益。

- 自动空间管理 — 不必再修复常规的空间溢出条件,每月节约 30 小时(20 天 x 1.5 小时/天)。
- 实用程序增强 — 由于具有更智能的默认设置,实用程序调优每月节约 5 小时(20 天 x 0.25 小时/天)。
- 这里未计算预期的数据可用性提高带来的获益,但对于某些客户来说这是一项可量化的财务获益。

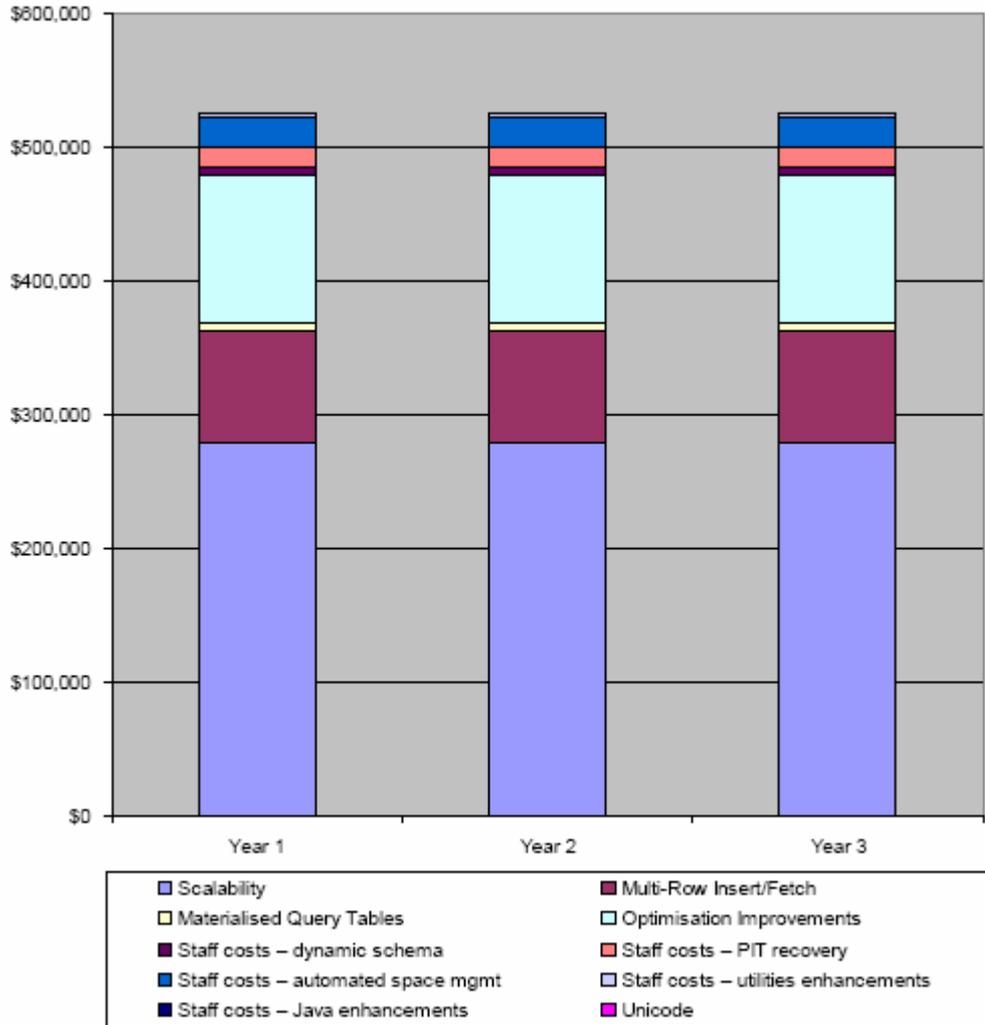


图 18 SAP 客户财务获益

投资回报

以上所述成本和获益可在 18 个月内实现投资回报，如下表所示：

	第 1 年	第 2 年	第 3 年
成本	\$517,990	\$448,390	\$448,390
获益	\$524,638	\$524,638	\$524,638
节约	\$6,648	\$76,248	\$76,248
总计	\$6,648	\$82,896	\$159,144

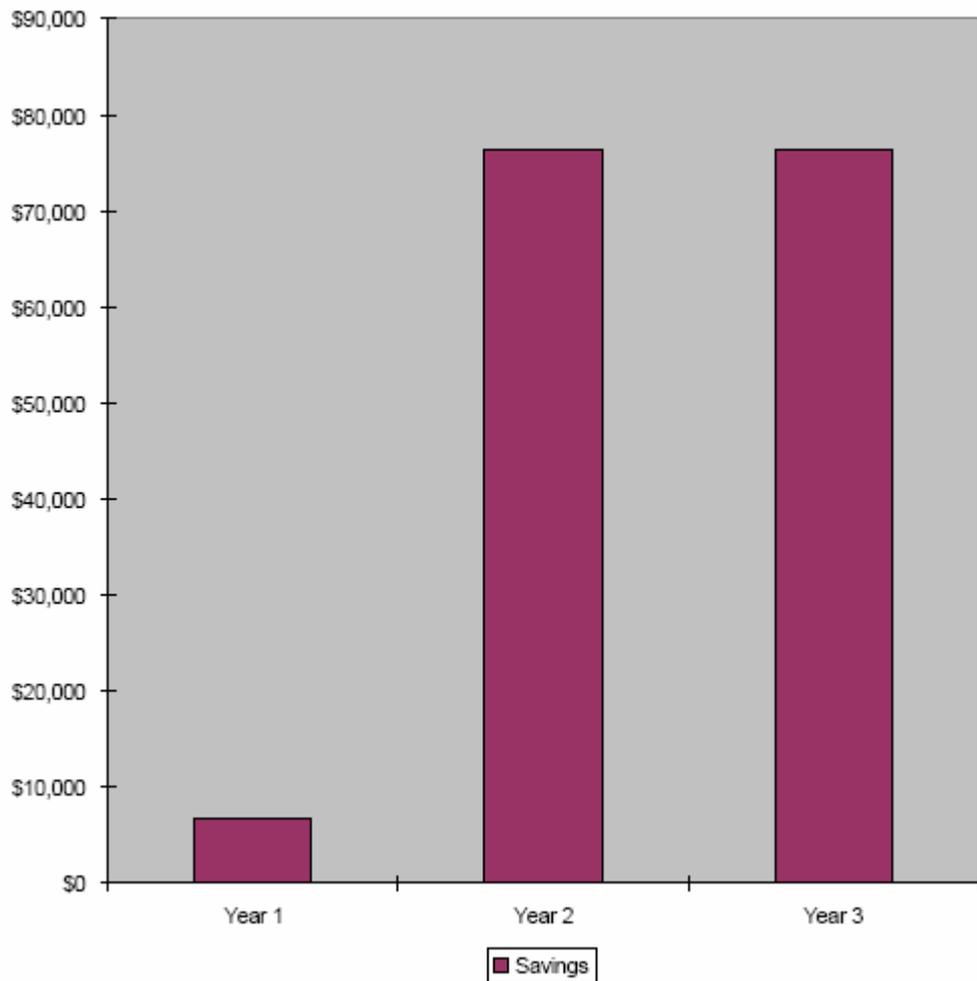


图 19 SAP 客户总成本节约

场景 3: Java 客户

在最后一个场景中我们集中关注典型的 Java 客户，该客户将 DB2 作为核心，使用优秀的 Web 功能来运行高容量的基于 Web 的事务处理系统。

客户运行 4 个单独的 DB2 系统，其中包括两个开发系统和两个采用数据共享配置的产品系统，以提供额外的弹性。

客户的升级成本如下表所示：

	第 1 年	第 2 年	第 3 年
额外的 DB2 许可证成本	\$61,500	\$61,500	\$61,500
CPU 成本 — 性能衰退	\$386,890	\$386,890	\$386,890
人员成本 — 升级	\$69,600	\$0	\$0
总计	\$517,990	\$448,390	\$448,390

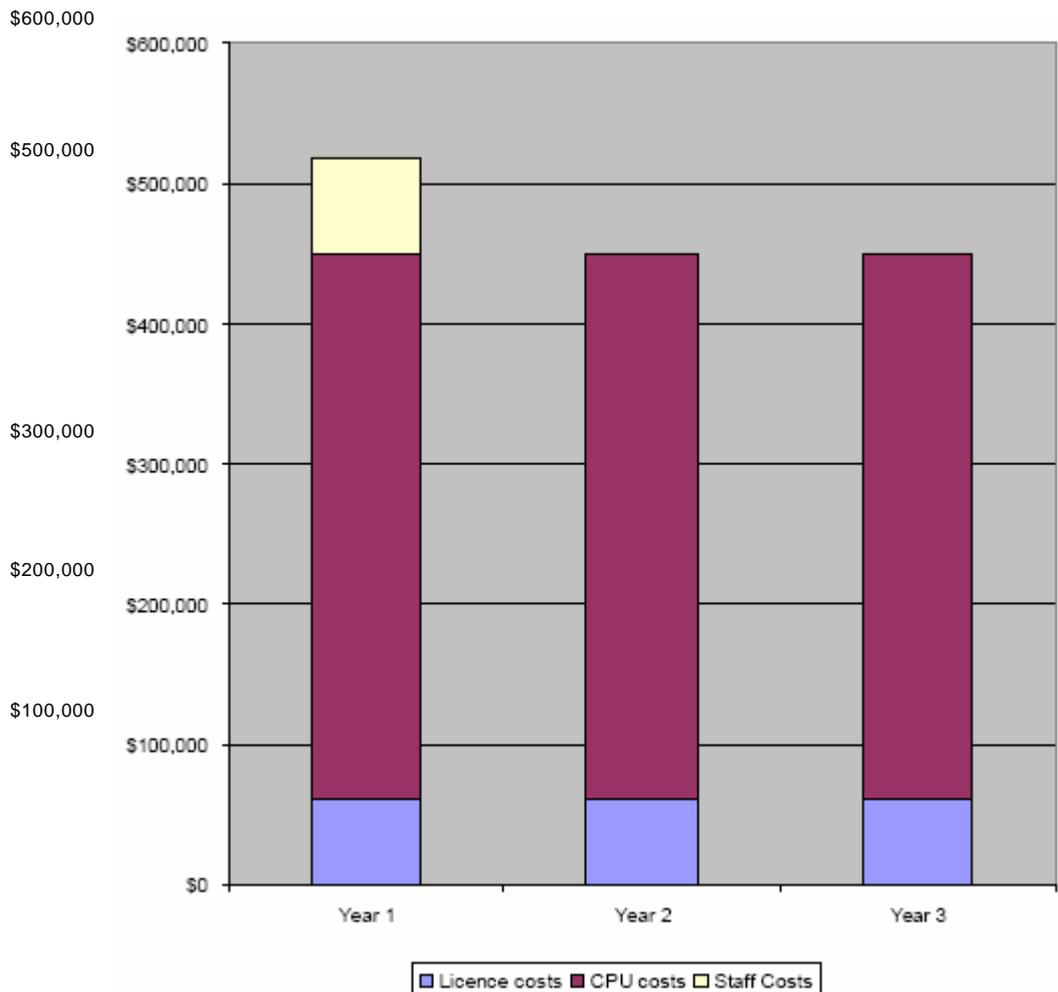


图 20 Java 客户升级成本

注

- 无硬件成本，因为客户已拥有 64 位容量的服务器。
- DB2 许可证和 64 位衰退 CPU 成本按第 43 页“示例客户环境”所示计算。
- 人员成本按第 43 页“示例客户环境”所示计算。

客户的财务获益如下表所示：

	第 1 年	第 2 年	第 3 年
可伸缩性	\$222,920	\$222,920	\$222,920
多行插入/读取	\$55,730	\$55,730	\$55,730
物化查询表	\$0	\$0	\$0
优化改进	\$278,650	\$278,650	\$278,650
人员成本 — 动态模式更改	\$43,200	\$43,200	\$43,200
人员成本 — PIT 恢复	\$2,880	\$2,880	\$2,880
人员成本 — 自动空间管理	\$14,400	\$14,400	\$14,400
人员成本 — 实用程序增强	\$3,600	\$3,600	\$3,600
人员成本 — Java 增强	\$14,400	\$14,400	\$14,400
总计	\$635,780	\$635,780	\$635,780

注

- 采用 64 位架构内存后由于吞吐量增大和效率提高获得的可伸缩性成本节约约占总体拥有成本的 4%。
- 多行插入/读取允许使用更少的单独指令并减小了远程操作的网络开销，这项成本节约约占总体拥有成本的 1%。
- 此环境中不涉及物化查询表的成本节约。
- 优化允许 DB2 为现有应用程序选择更高效的访问路径，这项成本节约约占总体拥有成本的 5%。
- 人员成本节约估算如下：
 - 动态模式更改 — 由于能够更快实现数据库结构的更改，每月节约 60 小时（20 项更改 x 3 小时/项）。
 - 时间点恢复 — 由于备份/恢复过程更快更简单，每月节约 4 小时（2 次恢复 x 2 小时/次）。
 - 自动空间管理 — 不必再修复常规的空间溢出条件，每月节约 20 小时（20 天 x 1 小时/天）。

- 实用程序增强 — 由于具有更智能的默认设置，实用程序调优每月节约 5 小时（20 天 x 0.25 小时/天）。
- 这里未计算预期的数据可用性提高带来的获益，但对于某些客户来说这是一项可量化的财务获益。

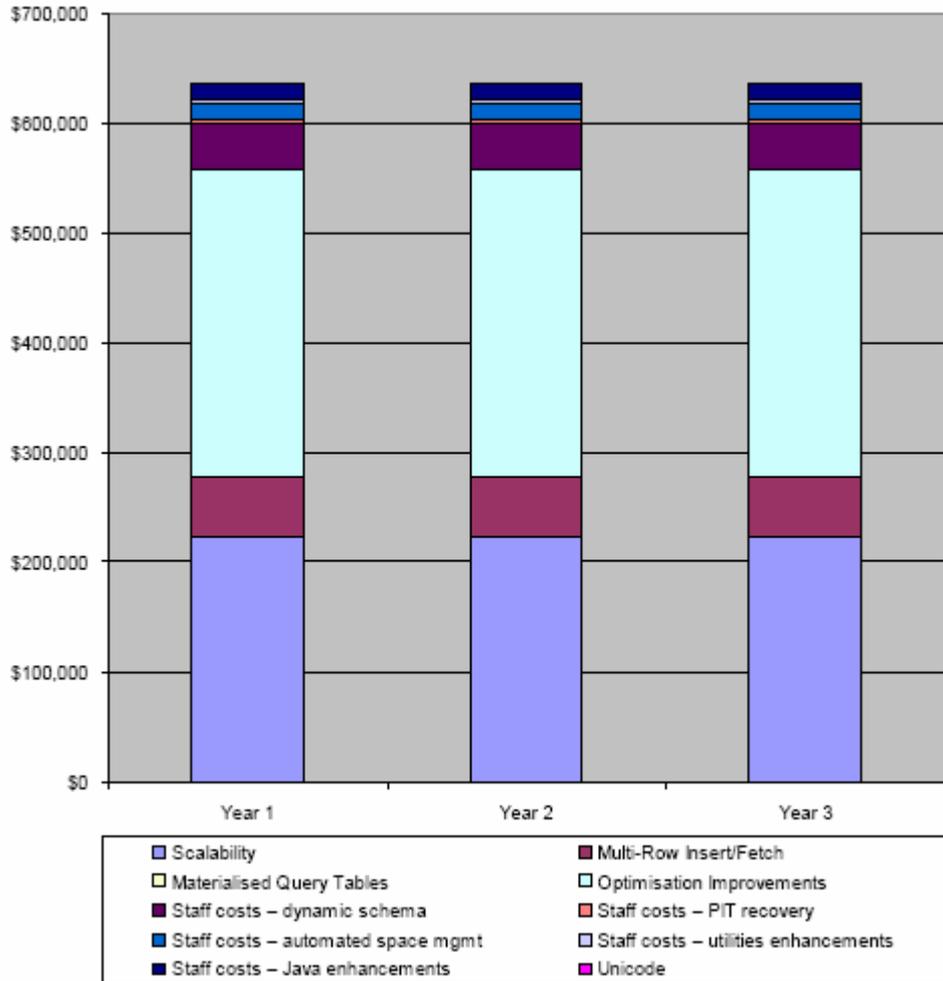


图 21 Java 客户财务获益

投资回报

以上所述成本和获益可在 18 个月内实现投资回报，如下表所示：

	第 1 年	第 2 年	第 3 年
成本	\$517,990	\$448,390	\$448,390
获益	\$635,780	\$635,780	\$635,780
节约	\$117,790	\$187,390	\$187,390
总计	\$117,790	\$305,180	\$492,570

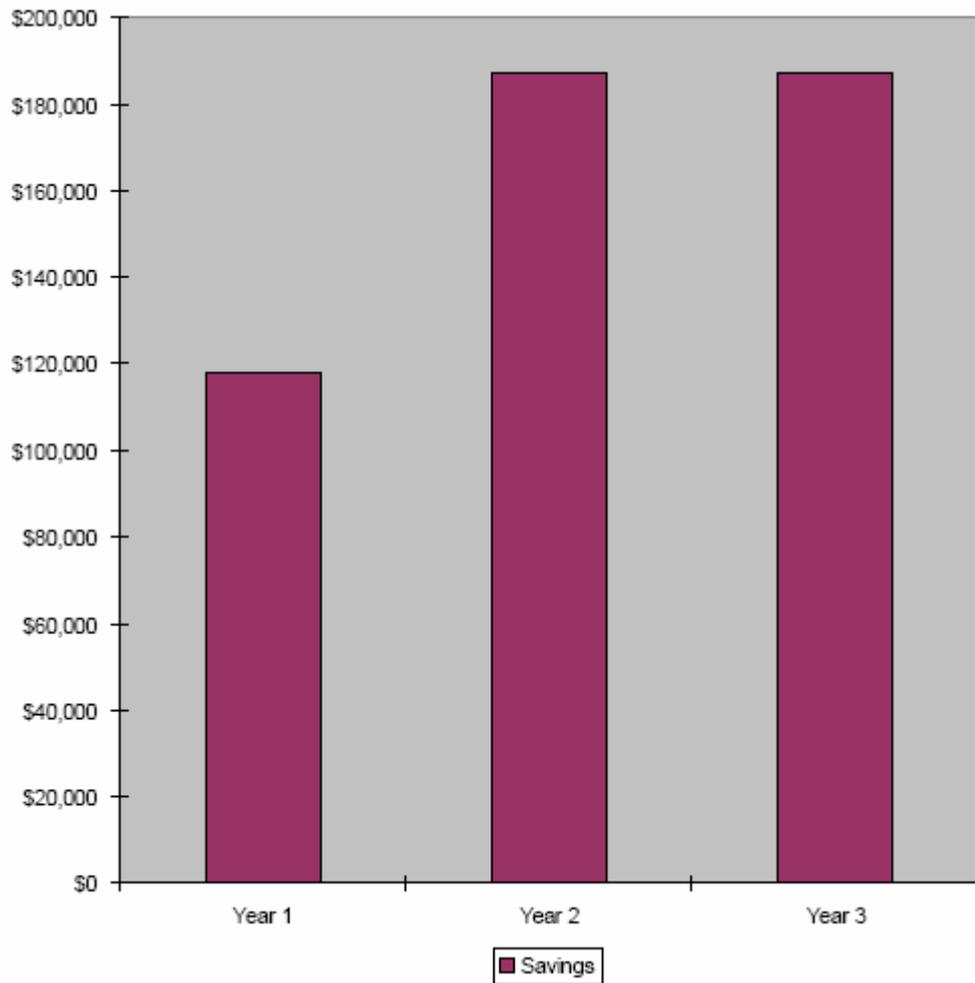


图 22 Java 客户总成本节约

结束语

DB2 for z/OS Version 8.1 有望成为 IBM 旗舰数据库管理产品的最重要版本之一。它是首个突破 31 位计算限制的中间件产品，它利用了 IBM 的 64 位 zSeries 机器的强大可伸缩性，允许客户驱动更大的工作负载，并且“以更少资源完成更多工作”。

新产品在数据可用性方面也做了大幅改进，进一步减少了计划内宕机时间并能够满足当今按需应变应用程序的 24/7 运行要求。利用这些特性，许多客户可以更频繁地更改数据库，从而允许客户对不断变更的市场条件或意外性能问题做出更快响应。

虽然向 64 位环境迁移引入了其固有的 CPU 要求，但 Version 8 的重要性能增强无需更改（或只需较少更改）应用程序即可实现，因此所获得的性能提高很容易抵消增加的 CPU 开销。性能的增强降低了 DB2 应用程序的总体拥有成本，令每项事务的执行成本更低，并允许服务器处理更大的工作负载。

新版本还根除了许多妨碍非 DB2 应用程序向 zSeries 平台移植的壁垒。如第 40 页的案例研究所示，移除这些壁垒后为以具有竞争力的成本创建服务器整合项目开辟了一条途径。

最后，由于产品具有更完善的自动化和智能，因此能够实现显著的生产力节约。这将经验丰富的 DB2 管理员和开发人员从低级例行任务中解放出来，投入更多时间来编写更好的、更高效的应用程序。

总之，这些新特性代表了 DB2 功能的重要改进，在迁移到新版本后，客户将获得更大的财务收益和技术收益。

附录 A — 参考文献

- SG24-6871 DB2 UDB for z/OS Version 8 Technical Preview
- GC18-7428 DB2 UDB for z/OS V8: What's New?
- SG24-6079 DB2 UDB for z/OS V8: Everything You Ever Wanted To Know
- SG24-7088 DB2 UDB for z/OS V8: Through the Looking Glass and What SAP Found There

附录 B — 致谢

作者在此谨向为本文做出卓越贡献的以下人员表示衷心感谢：

John Campbell	杰出工程师 DB2 for z/OS Development, IBM Silicon Valley Laboratory
Kevin Campbell	应用架构师 Univar USA
Jackie Chu	SWG Pricing, IBM Software Group - Data Management Division
Catalin Comsia	高级 IT 专家 DB2 for z/OS
Curt Cotner	IBM 院士 DB2 for z/OS Development
Malcolm Nolan	营销经理, DB2 for z/OS IBM Software Group - Data Management Division
Ian Cook	DB2 for z/OS Product Introduction Center, Europe
Sandy Smith	Data Management 主管 Hewitt Associates
Roger Miller	高级战略专家 DB2 for z/OS Development, IBM Silicon Valley Laboratory
Robert Perih	Database System 经理 Citigroup
Mark Wilson	DB2 for z/OS Product Introduction Center, Americas
Maureen Zoric	项目经理 DB2 UDB for z/OS and OS/390