



目 录

目 录.....	2
第 21 章 基于 Tuxedo 对大机应用的迁移-- ART.....	3
21.1 ART 简介.....	3
21.2 Application Rehosting Workbench.....	3
21.2.1 关键特性.....	3
21.2.2 优点.....	3
21.2.3 流程简介.....	4
21.2.4 详细流程.....	4
21.2.4.1 应用编目和分析 (Cataloger)	4
21.2.4.2 DB2 到 Oracle 的数据迁移.....	5
21.2.4.3 文件的迁移.....	5
21.2.4.4 COBOL 程序迁移.....	6
21.2.4.5 JCL 脚本转换.....	6
21.3 ART for CICS 运行环境.....	7
21.3.1 关键特性.....	7
21.3.2 优点.....	7
21.3.3 流程简介.....	7
21.3.4 详细流程.....	8
21.3.4.1 ART CICS Runtime 简介.....	8
21.3.4.2 ART CICS Runtime 配置.....	10
21.4 ART for Batch 作业运行环境.....	11
21.4.1 流程简介.....	12
21.4.2 详细流程.....	13
21.4.2.1 概述.....	13
21.4.2.2 运行环境.....	13
21.4.2.3 创建一个脚本.....	13
21.4.2.4 创建一个过程.....	13
21.4.2.5 控制脚本的执行.....	14
21.4.2.6 文件使用.....	15
21.4.2.7 用 EJRC 提交一个 JOB.....	16

第 1 章 基于 Tuxedo 对大机应用的迁移-- ART

1.1 ART 简介

ART 是 Application Runtime 的简称，是一个专业解决 IBM 大型机应用迁移的工具。将大型机应用程序迁移到开放系统避免了应用程序重写的成本和风险，降低大型机 MIPS 消耗，保持大型机级别的服务质量，借助灵活的、支持 SOA 的环境以提高业务敏捷度。

ART 组件包括：

- Oracle Tuxedo Application Runtime for CICS and Batch
 - CICS API
 - 终端仿真和服务
 - 批处理作业执行环境
- Oracle Tuxedo Application Rehosting Workbench
 - 应用程序信息库与编目器
 - 语言迁移工具：COBOL、JCL
 - 数据迁移工具：VSAM 和普通文件、DB2

1.2 Application Rehosting Workbench

1.2.1 关键特性

1. 自动化：提高生产力
2. 精确性：降低错误率
3. 高效：自动转换数千万行代码
4. 可重复性：确保多次重复的结果一致
5. 可扩展性：方便修改和添加转换规则

1.2.2 优点

1. 一致性、高效率
2. 将项目成本和风险降至最低
3. 处理大规模应用资产
4. 统一的转换，易于维护
5. 支持用户特定的转换，易于扩展

1.2.3 流程简介

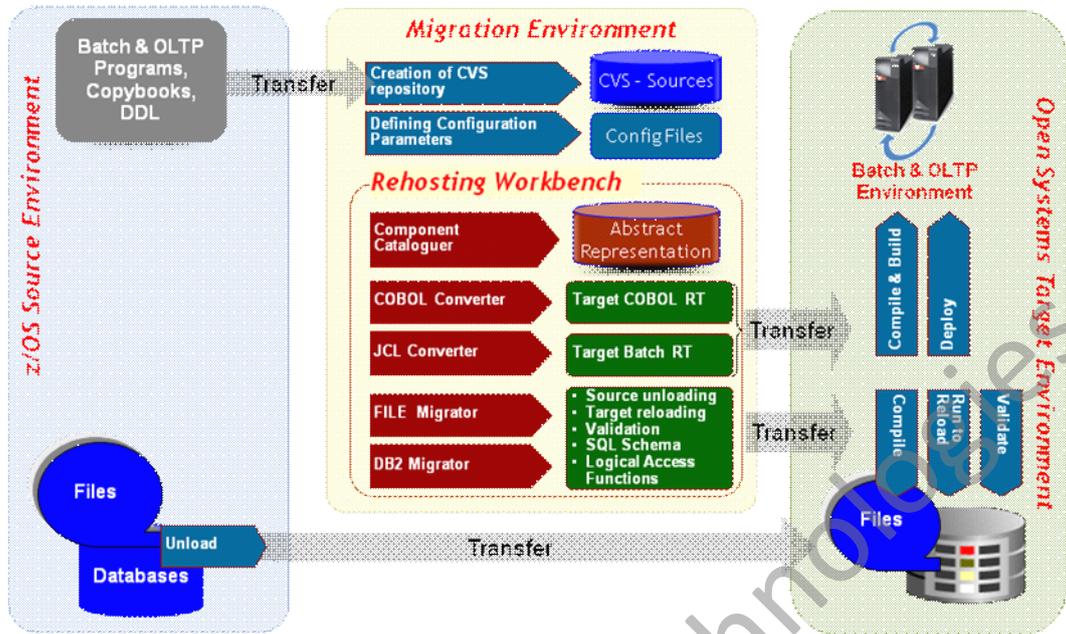


图 21-1

1.2.4 详细流程

1.2.4.1 应用编目和分析 (Cataloger)

1、 功能

- 1) 分析所有资源组件
 - 探查内在的不一致
 - 为组件创建内部表示方法
- 2) 分析组件之间的引用关系
 - 探查资源级别的不一致
 - 分析组件之间的交叉引用
- 3) 报告资源清单以及不一致之处
 - 包括未使用或者丢失的组件

2、 配置

- 1) 系统描述文件
 - 描述了迁移平台所有的源码文件的位置和类型
 - 描述引用资源的位置
 - 描述各种转换以及分析的选项
- 2) Hint 参数文件
 - 补充的一些引用信息

- 可以克服静态分析的一些局限性而获得资源的真实结构图

3、 命令

1) Catalog

- 和 preparse, analyze, fast-final 相结合的命令

2) preparse/preparse-files

- 转换资源生成内在的.pob 描述文件 (Persistent Object Base)
- 可并行运行

3) analyze

- 分析.pob 文件
- 构建 Cataloger 描述表 (生成 symtab-<ProjectName>.pob 文件)

4) fast-final

- 运行 post-analysis 以及生成报告
- 分析组件之间相互引用关系
- 指出引用关系之间的不一致
- 生成报告

1.2.4.2 DB2 到 Oracle 的数据迁移

4、 功能：转换 z/OS DB2 对象到开放平台 Oracle 对象

1) 迁移所有的 SQL 对象

2) 支持所有常用数据类型

3) 处理保留字 (重命名)

4) 相关处理过程：

- 使用 DSNTIAUL 下载 DB2 表，
- 用逻辑名自动处理 8 字符长名字限制；
- 转码，
- 导入数据

5、 配置

1) 系统描述文件

2) 数据库参数文件

- 列出要重名的表或者列

1.2.4.3 文件的迁移

6、 功能：转移 z/OS 文件 (VSAM, QSAM, GDG) 到目标平台的 ISAM 文件或者 ORACLE 数据库

1) 下载物理文件到目标平台

2) 转码 (EBCDIC-to-ASCII) 以及导入数据

3) 执行初步检查

4) 将文件转换到相关的 DBMS

- 翻译成关系模型
- 优化关系模型
- 转换应用程序代码来连接关系型数据库

7、配置:

1) 系统描述文件

2) 数据库参数文件

3) 其它配置

- 对于每一个要迁移的文件, 都要在 COBOLcopybook 中描述
- Datamap-<configuration name>.re 列出所有物理文件名, 以及每个文件的类型
- mapper-<configuration name>.re 列出所有文件, 指定相应 COBOL 描述以及区别的规则

1.2.4.4 COBOL 程序迁移

8、功能: 转换 IBM COBOL 源程序到开放平台的 COBOL 程序

1) 转换 IBM Enterprise COBOL 程序为 MicroFocus COBOL 程序 (或其它开放平台 COBOL 程序)

2) file-to-Oracle 转换匹配, 组件重命名

3) 嵌入式 SQL 转换

4) 把程序中的 EXEC CICS 命令正规化

9、配置

1) 系统配置文件

2) 主配置文件:

- 各种参数, 例如是否在转换后合并 copybook

3) 从属配置文件, 可能是手写的也可能是工具生成的:

- 组件重命名
- 重命名标识符
- File-to-Oracle 转换信息

1.2.4.5 JCL 脚本转换

10、功能

1) 转换 IBM JCL jobs (主文件, PROCs, INCLUDEs, SYSIN 文件...)到目标平台的脚本:

- Ksh 脚本语言
- ART 脚本运行环境的组件调用（程序执行，文件分配，文件处理，应用调用等）

2) 生成相关源程序

- JCL: 主 JCL 文件，这些文件定义了一个或多个 JCL job.
- JCL-Lib: JCL 引用文件，定义 EXEC 调用的过程，或者 INCLUDE 的引用
- JCL-Sysin: 应用程序运用的 SYSIN 文件

11、配置

1) 系统配置文件

2) 主配置文件:

- 各种参数，例如数据文件根目录，排序应用

3) 从属文件，可能是手写的也可能是工具生成的:

- 组件重命名
- 重命名标识符
- File-to-Oracle 转换信息

1.3 ART for CICS 运行环境

1.3.1 关键特性

- 基于成熟的 Tuxedo 产品
- 支持多节点、多平台、多种网络协议
- 线性可伸缩
- 开放、SOA、可扩展
- 集中管理

1.3.2 优点

- 具备大型机的性能（RASP），保持服务质量
- 只需相当于大型机成本的一小部分即可
- 灵活、能够满足未来需求的基础架构
- 更快速、经济的集成，更高的重用率和开发人员效率

1.3.3 流程简介

应用角度:

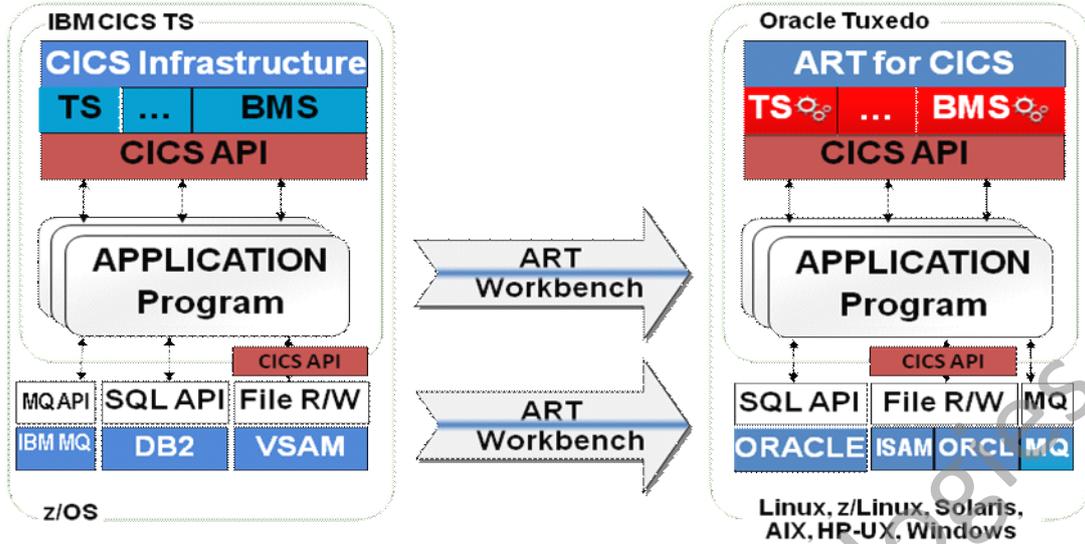


图 21-2

系统角度：

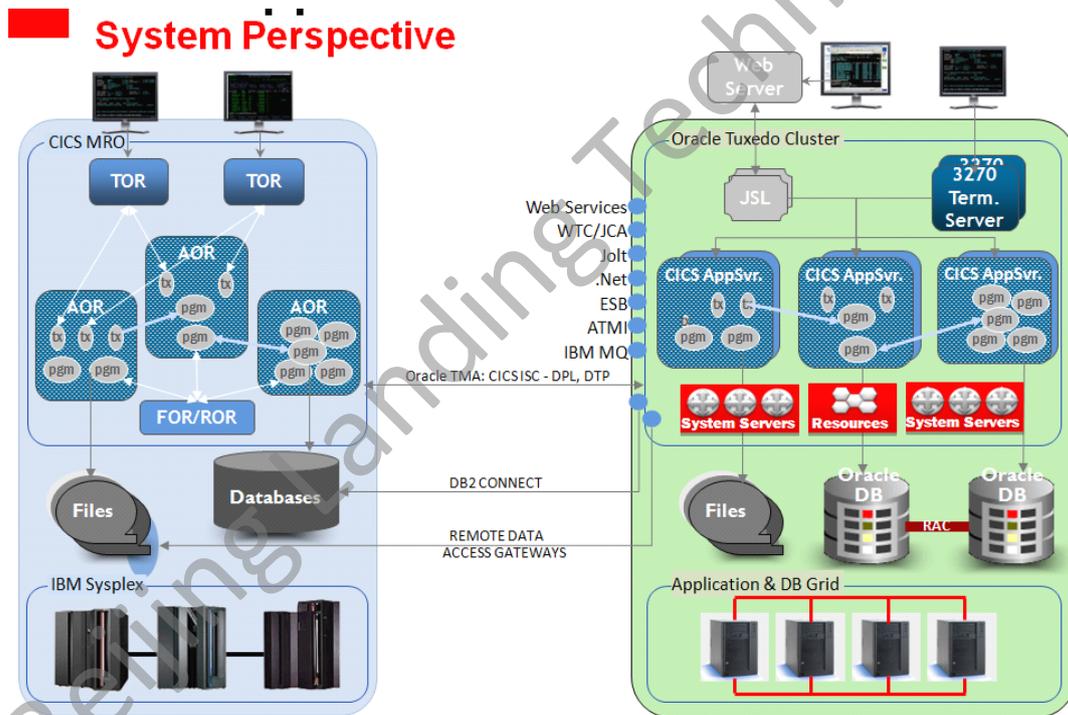


图 21-3

1.3.4 详细流程

1.3.4.1 ART CICS Runtime 简介

1. 架构

- CICS 是一个交易中间件。

- 当使用 ART workbench 从 Z/OS 移植到 UNIX/Linux 平台时, ART CICS Runtime 提供了 CICS 运行环境和 3270 终端仿真。
2. ART CICS 运行环境主要组成部分
 - CICS 预处理和运行库;
 - CICS Runtime Tuxedo 服务以及资源配置文件
 3. ART CICS 运行库
 - 在 Z/OS CICS 应用中,所有的操作和资源管理都是通过 EXEC CICS API 来实现的。
 - ART CICS 提供了执行 EXEC CICS API 的运行库
 4. ART CICS Tuxedo 服务进程

ART CICS 基于 Tuxedo 的服务进程和服务来运行迁移后的 CICS 交易 (Transaction), 管理 3270 仿真终端的连接等:

 - 1) 必要的服务进程
 - 终端连接进程 (ARTTCPH 和 ARTTCPL): 管理 3270 仿真终端接入 ART CICS 应用。
 - 连接进程 ARTCNX: 管理用户的会话以及提供系统 Transaction (CSGM: Good Morning Screen, CESN: Sign On, CESF: Sign off)。
 - 同步交易服务进程 ARTSTRN: 管理标准的 CICS 同步交易。
 - 2) 可选性服务进程
 - 同步交易服务进程 ARTSTR1: 管理不可以同时运行, 但可以顺序执行的 CICS 同步交易。
 - 异步交易服务进程 ARTATRN 和 ARTATR1: 和 ARTSTRN、ARTSTR1 相似, 但是是服务于异步的交易, 即以 EXEC CICS START TRANSID 开始的交易。
 - TS Queue 服务进程 ARTTSQ: 管理 CICS TS queue-通过 CICS 专门的命令管理的文件。
 - TD Queue 服务进程 ARTTDQ: 管理 CICS TD queue
 - 3) 服务进程的配置
 - 这些服务进程都是 Tuxedo SERVER, 配置在 UBB 中。
 - ART CICS 服务进程会读取配置在 CICS runtime 资源配置文件中的信息。
 - 4) ART CICS Runtime 资源管理文件
 - z/0s CICS 资源管理

在 Z/OS 中, 所有 CICS 应用使用的组件 (terminals, transactions, programs, maps, files ...) 都要用一个叫 CSD 的声明文件来声明。

每个资源都必须属于一个资源组, 这样就能让一组资源连接在一起来组成一个功能或应用来管理 (install, delete, copy 到另外的 CSD...)。

 - ART CICS Runtime 资源管理

ART CICS runtime 只管理在 Z/OS 的 CICS CSD 文件中预先定义的资源信息的子集，这些文件都存放在 UNIX 的指定目录中。

CICS 管理以下资源：

a) 交易类 Tranclasses (transclasses.desc 文件)

这个文件包括 CICS 交易相关的所有类，在 ART CICS runtime 中，tranclass 提供一种定义是否几个相同交易的实例能够并行执行的机制。

b) 交易 Transaction(transaction.desc 文件)

一个交易属于一个 transaction class，每个交易要在这个文件中指定其对应的程序名。

c) 程序 Program(program.desc 文件)

这个文件包含交易对应的程序，或通过 LINK 或者 XCTL 调用的程序。

d) TS Queue 模型 (tsqmodel.desc 文件)

包含所有 CICS 程序中用到的 TS queue 所关联的 TS queue 模型。这些模型用作定义 TS queue 是否可恢复，以及存储在用数据库、文件、还是内存。

e) Mapsets (mapsets.desc 文件)

这个文件中包含了 CICS 应用所关联的所有的 mapset。Mapset 是包含一个或多个 screen（用于终端和 CICS 程序之间的交互）的物理组件。这些资源被专门的 CICS 语句像 EXEC CICS SEND 或者 RECEIVE MAP 使用。

f) Typeterms (typeterms.desc 文件)

包含所有 3270 终端的类型，CICS runtime TCP 进程支持这些类型。

1.3.4.2 ART CICS Runtime 配置

1. ART CICS Runtime 配置

- 可以使用 UNIX 系统中的 `/.profile` 文件定义 CICS runtime 使用的或者 Tuxedo 使用的 PATH 以及其它环境变量。
- 也可以在 Tuxedo 的 `envfile` 中为某些服务进程定义特定的环境变量。
- 在 Tuxedo 的 `UBBCONFIG` 中配置所需的 CICS runtime 服务进程。
- 配置 ART CICS runtime 使用的 CICS 资源。

2. ART CICS 使用的环境变量

变量名	取值	用途	环境
TUXDIR	在安装时设置	必须。指定 tuxedo 安装目录，默认 <code>/usr/tuxedo</code>	TUXEDO
TUXCONFIG	在安装时设置	必须。tuxconfig 文件的全路径	TUXEDO
KIXDIR	在安装时设置	必须。CICS Runtime 组件的绝对路径	CICS Runtime
APPDIR	<code>\${KIXDIR}/bin</code>	必须。CICS Runtime Servers 二进制文件的路径	CICS Runtime

KIXCONFIG	在安装时设置	必须。CICS Runtime 资源配	CICS Runtime
KIX_TS_DIR	在安装时设置	必须。不可恢复的 CICS Queue	CICS Runtime
		置文件的路径	
		TS 目录	

表 21-1

3. Tuxedo UBBCONFIG 文件

必须的 ART CICS 服务进程包括 ARTTCPL 和 ARTCNX

```
*SERVERS
ARTTCPL SRVGRP=TCP00 SRVID=101 CLOPT="-o
/home/landingbj/demo/Logs/TUX/sysout/stdout_tcp -e
/home/landingbj/demo/Logs/TUX/sysout/stderr_tcp -- -M 4 -m 1 -L //deimos:2994 -
n //deimos:2992"
ARTCNX SRVGRP=GRP01 SRVID=15 CONV=Y MIN=1 MAX=1 RQADDR=QCX015 REPLYQ=Y
CLOPT="-o /home/landingbj/demo/Logs/TUX/sysout/stdout_cnx -e
/home/landingbj/demo/Logs/TUX/sysout/stderr_cnx -r --"
```

示例 21-1

4. ART CICS runtime 资源配置文件

必须的 ART CICS 资源配置文件包括 typeterms.desc 和 mapsets.desc

- typeterms.desc 配置文件

这个文件用于 ARTTCPL，定义了不同种类的终端属性。

```
[typeterm]
name=IBM-3279-5E
color=YES
defscreencolumn=80
defscreenrow=24
description="IBM 327x family terminal"
highlight=YES
logonmsg=YES
outline=NO
swastatus=ENABLED
uctran=NO
userarealen=0
```

示例 21-2

- mapsets.desc 配置文件

该文件的以下配置用来启动 CSGM 交易画面，

```
[mapset]
name=ABANNER
filename=<KIXDIR>/sysmap/abanner.mpdef
```

示例 21-3

1.4 ART for Batch 作业运行环境

1.4.1 流程简介

Application Perspective

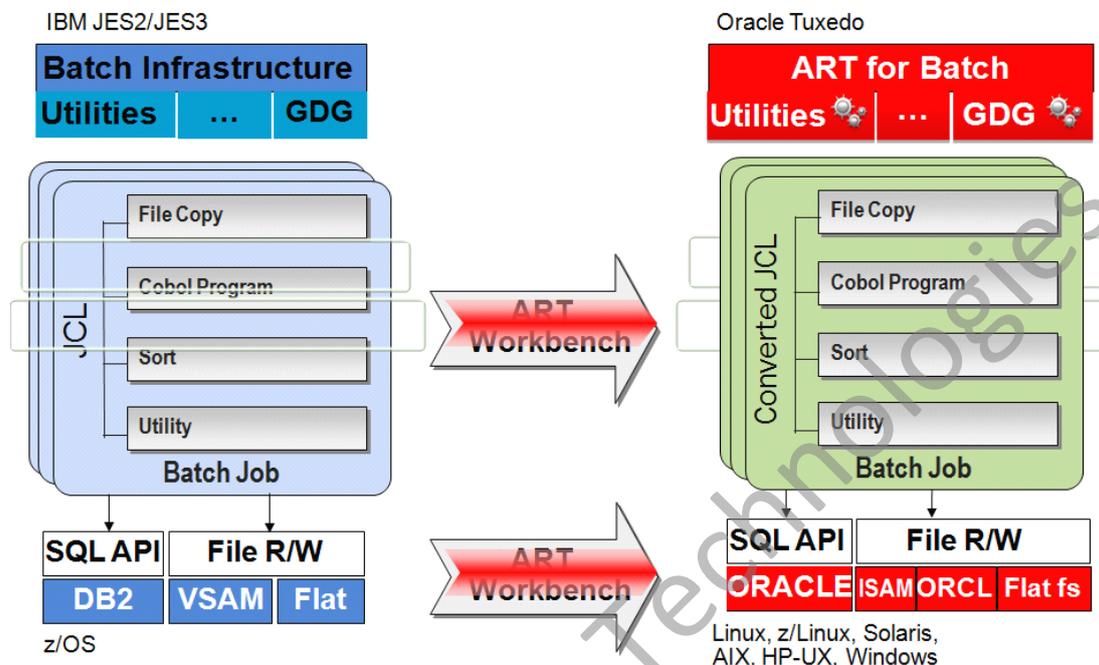


图 21-4

System Perspective

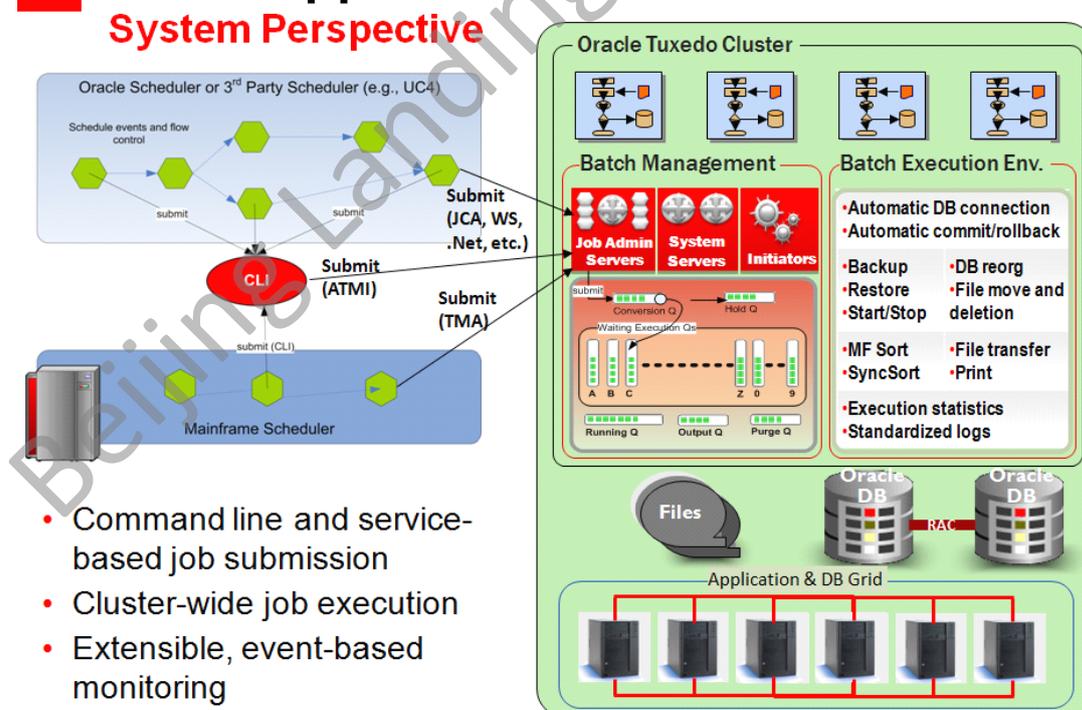


图 21-5

1.4.2 详细流程

1.4.2.1 概述

ART Workbench 将 z/OS JCL 转换成可以在开放平台上运行的 KSH。当在 ART Batch 运行环境中提交一个作业时，这个 Ksh 脚本通过以下步骤执行：

- 输入：在这个步骤中，要分析作业的参数
- 转换：这个过程中，执行以下动作 1. 加入所有的外部 KSH (procedures 和/或 includes) 2. 用当前值代替脚本中用到的变量
- 执行：在运行环境中执行该脚本

1.4.2.2 运行环境

ART for Batch 环境变量

变量	用途
DATA	指出数据存放地方
TMP	应用文件的暂存目录
SPOOL	输出目录
PROCLIB	PROC 和 INCLUDE 文件目录
MT_ACC_FILEPATH	文件之间并发访问
MT_DB_LOGIN	数据库连接用户
MT_LOG	日志目录
MT_TMP	内部文件的暂存目录

表 21-2

1.4.2.3 创建一个脚本

1. 脚本的结构

ART for Batch 提供了一个脚本模板，这个模板中指出了每个过程的清楚步骤，用以规范使用 KSH。KSH 脚本中每一个步骤 (STEP) 都对应一个标号 (LABEL)，一个步骤结束后，JUMP_LABEL 变量的值改变，以执行下一个步骤。

2. 定义和使用 symbol

Symbol 是一个脚本文件中的变量，ART Batch 的 KSH 中，变量可以使用 \$[symbol] 来定义，可以通过 m_SymbolSet 来给变量赋值。

3. 创建执行程序步骤

一个步骤 (STEP)，是调用一系列的运行环境的功能来完成一个具体的动作。最常见的就是执行一个应用程序。一个作业中通常包含多个步骤。

1.4.2.4 创建一个过程

● 过程简介

- ART for Batch 中的过程和 Z/OS 中的过程 (Procedure) 原则上一致。

● 过程优点

- 一次编写，然后可以多次调用

- 过程可以统一修改
- 过程分类
- In-stream 过程：包含在脚本之中，但这个过程只能用在当前的脚本中
- External 过程：写在其他的文件中，与脚本文件分开，这样可以被多个脚本使用
- 创建一个 In-stream 过程
- 该过程定义在文件最后，也就是到 m_JobEnd 后。在 KSH 脚本中以 m_ProcBegin 开始，期间进行一些具体操作后，以 m_ProcEnd 结束。
- 创建一个 External 过程
- 该过程就不要使用 m_ProcBegin 和 m_ProcEnd，在过程中就是简单的代码，为了显示这个是过程代码，可以开始于：

```
JUMP_LABEL=FIRSTSTEP  
;;  
(FIRSTSTEP)
```

示例 21-4

结束于：

```
JUMP_LABEL=ENDPROC  
;;  
(ENDPROC)
```

示例 21-5

- a) 使用过程
- 在一个 KSH 脚本中使用一个过程，要用 m_ProcInclude 来调用，在 batch 执行的转换阶段，所调用的过程会被展开到主脚本中。
- b) 在执行时修改过程
- 过程中的任务在执行时可以根据运行状态改变，通过修改 symbol 或者指定 parameter 这两种方法都可以实现。

1.4.2.5 控制脚本的执行

- c) 控制执行的步骤
- 使用 m_CondIf, m_CondElse, and m_CondEndif

可以控制一个或多个步骤的执行，这个相当于 Z/OS 中的 IF, THEN, ELSE 和 ENDIF。

- 使用 m_CondExec

可以控制是否执行该步骤，m_CondExec 必须有一个或多个参数，执行的条件是该参数或者所有参数都符合时才执行。1. m_CondExec 4, LT, STEPEC01 表明是在返回结果符合要求才执行；2. m_CondExec EVEN 表明就算前一步非正常中断了也执行接下来的工作；3. m_CondExec ONLY 表明只有在前一步非正常结束时才执行接下来的工作。

- d) 改变默认的输出信息

- 如果 batch 管理员想改变默认的输出信息，可以通过 MT_DISPLAY_MESSAGE_FILE 指定的配置文件来修改。这个文件是用分号隔开字段的 CSV (comma separated values) 文件，在文件中每个记录都分成 6 个部分：
 1. 信息定义
 2. 能显示该信息的函数 (默认 '*')
 3. 显示级别
 4. 显示的目的
 5. 保留以后用
 6. 显示的信息

1.4.2.6 文件使用

e) 创建文件定义

- 使用 m_FileBuild 函数定义一个文件，支持三种文件：顺序文件，顺序行文件，索引文件。

f) 分派以及使用文件

- 在运行中，一个文件可以被一个函数使用也可以被一个程序使用，在使用前，一定要用 m_FileAssign 指定：指定打开的模式，指定访问模式，指定该文件是否为生成文件，指定访问的逻辑名所映射到的具体物理位置。

g) 使用 GDG 文件

- 使用 m_GenDefine 函数来指定，其中唯一参数-s 来指定该文件的最高版本。
- 可以使用 m_FileAssign 函数的-g 参数，指明该文件是 GDG 文件。

h) 使用 In-Stream 文件

- 文件的数据直接从 KSH 脚本中写入，使用-i 参数，以_end 结束 in-stream 流。

i) 使用连接文件集

- m_FileAssign 函数-C 参数指定使用一系列文件作为连续输入 (在 Z/OS 的 JCL 中作为 DD card, 该 DD card 只有第一个含有 label) 。

j) 使用外部的“sysin”

- 使用 m_UtilityExec 函数可以使用含有执行命令的“external sysin”文件。

k) 删除文件

- 使用 m_FileDelete 可以删除文件。

l) RDB 文件

- z/OS 文件可以通过 ART Workbench 转换为关系型数据库表，.rdb 文件描述了转换的对应关系。

m) 使用 RDBMS 连接

- 当执行程序要使用到数据库连接时，可以使用 m_ProgramExec 的-b 参数。

- 数据库的创建连接、断开连接，提交、回滚都是隐式执行的，数据库连接串通过环境变量 MT_DB_LOGIN 指定，用户名和密码必须正确。

1.4.2.7 用 EJR 提交一个 JOB

n) TuxJES 可以发起一个 job，一个 job 也可以直接通过 EJR 发起。

在执行前，确保整个环境的正确性，包括环境变量以及目录。

Beijing Landing Technologies