

HP Service Health Reporter

软件版本：9.30

Performance Insight 数据迁移指南



目录

简介.....	2
为什么要进行迁移?	2
迁移目标.....	2
迁移的好处	2
迁移范围.....	3
一个好的数据迁移方案需满足的条件	4
术语定义.....	4
数据迁移 - 架构.....	5
假设	5
从 Performance Insight 进行的数据迁移 - OM 部署方案	6
从 Performance Insight 进行的数据迁移 - BSM 部署方案	7
数据迁移方法	7
已知问题和限制.....	8
数据迁移步骤.....	10
先决条件.....	10
PI 迁移内容包.....	10
在从 PI 迁移数据时执行的步骤.....	10
典型用例/方案	12
常见问题 [FAQ].....	13
参考	15
PI 的 System Resource 报告包与 SHR 中的 System Performance 内容之间的数据映射	15

简介

HP Performance Insight 是一款性能管理及报告应用程序。我们计划将现有 Performance Insight 客户迁移到 BSM 产品组合中的其他报告解决方案，以此作为一种阶段性方法。

为什么要进行迁移？

此迁移程序可帮助将当前 Performance Insight 客户（基于有效的支持合同）转移到其他 BSM 报告工具中。

HP PI 客户将迁移到一个或多个产品（基于用例）

- Service Health Reporter
- NNMi + Performance iSPI

迁移目标

- 在 HP 执行迁移以整合 BSM 报告工具之后，使 HP Performance Insight 客户能够体验几乎完全相同的功能。
- HP PI 客户可以选择迁移到 Service Health Reporter，从而利用交叉域分析及报告等增强功能。

本文档描述了关于在将数据从 HP PI System Resource 报告包迁移到 SHR 时，要执行的操作过程的详细信息。迁移到 SHR 之后，即可利用 SHR System Performance 内容。

迁移的好处

以下是迁移到 Service Health Reporter 之后可获得的一些好处（第 1 阶段）

- 升级到下一代跨 BSM 的报告解决方案
 - SHR 提供模型驱动的报告功能 => 将应用程序性能与底层基础架构关联
 - 提供有关您的环境的整体视图
 - 自动跟踪拓扑更新
 - 单个透明窗格用于查看应用程序、基础架构、数据库和服务网络的性能
- 将业务智能引入数据中心管理
 - 将 SAP Business Objects Enterprise 作为报告框架
 - 可更容易地进行自定义
- 其他功能
 - 在报告中提供停机因素/自定义分组/可自定义的移位，等
 - 提供平台以供将来分析，如虚拟容量规划等
- 可扩展性
 - 一个 SHR 实例可直接从 5000 个系统节点收集数据

本文档旨在描述

- 用于将数据从 Performance Insight 迁移到 SHR 的方法
- 在执行 Performance Insight 数据迁移以利用 System Performance 内容时需遵循的步骤
- 假设
- 已知限制
- 可扩展性和性能

迁移范围

从 Performance Insight 进行数据迁移时涉及的内容范围（第 1 阶段）限于 System Resource 报告包及以下子包：

- SystemResourceCPU
- SystemResource_Disk
- SystemResource_NetInterface

SHR 支持	SHR 不支持
<p>支持持续的服务改进和历史报告用例</p> <ul style="list-style-type: none"> - 整合的报告 - BSM 和非 BSM 方案 <p>从中间数据存储进行收集</p> <ul style="list-style-type: none"> - 除 Performance Agent 数据收集以外 	<p>SHR 中提供的是高级别的历史报告，而非近实时的报告，不可用于进行故障排除。</p> <ul style="list-style-type: none"> - 不提供用于诊断或故障排除的预置报告
<p>从 PI 进行一次性数据迁移</p> <ul style="list-style-type: none"> - 在迁移数据之后，会在 SHR 中配置从源（用于 System Performance 内容的代理和用于拓扑的 HP OM/RiSM）进行的实时收集 	<p>一次性数据迁移不包括：</p> <ul style="list-style-type: none"> - HP OM SPI（包括虚拟化 SPI） - InterfaceReporting/DeviceResource、通信量、MPLS、QoS、IPTelephony、服务保障和其他网络性能 SPI 指标
<p>仅会从 PI 迁移聚合数据（每小时和每日粒度），而不会迁移比率数据将不会迁移配置</p>	<p>将不会使用 PI 作为持续收集数据的源</p>
<p>支持通过内容开发环境 (CDE) 对内容进行自定义和扩展</p> <ul style="list-style-type: none"> - PI 中的各个报告本身将不会迁移到预置的 SHR 内容；但是，可以创建自定义报告来处理其他用例（如果有）。 - 也可以自定义预置内容 	<p>将不会收集低延迟数据</p> <ul style="list-style-type: none"> - 报告中不提供实时数据
<p>将支持创建用于报告的自定义组</p>	<p>在 Linux/Unix 或 Oracle/Sybase ASE 上不受支持 - Sybase IQ 是唯一受支持的数据库</p>

一个好的数据迁移方案需满足的条件

以下条件将帮助确定优秀的的数据迁移方案：

- PI 的主要使用对象：
 - 系统（OM 消息/数据库 SPI/虚拟化和 Sitescope）
 - 网络数据（由 Perf/iSPI 处理的数据）
- 环境范围
 - 针对 SHR 约为 5000 个节点
- 具有自定义基于 SAP Business Objects 的报告的专业技能
- 深入了解底层域（以创建自定义报告）
- 当前的 OM 或 BSM 用户
- 在报告解决方案中寻找比 PI 所提供功能更新的功能
- 移位、停机、自定义分组等

术语定义

术语	定义
CP	内容包
RP	报告包
DWH	数据仓库
HP PI	HP Performance Insight
BSM	Business Service Management
RtSM	运行时服务模型
OM	Operations Manager
CDE	内容开发环境
SPI	Smart Plug-In
SHR	Service Health Reporter
ETL	提取、转换及加载
ABC	审核、平衡及控制

数据迁移 - 架构

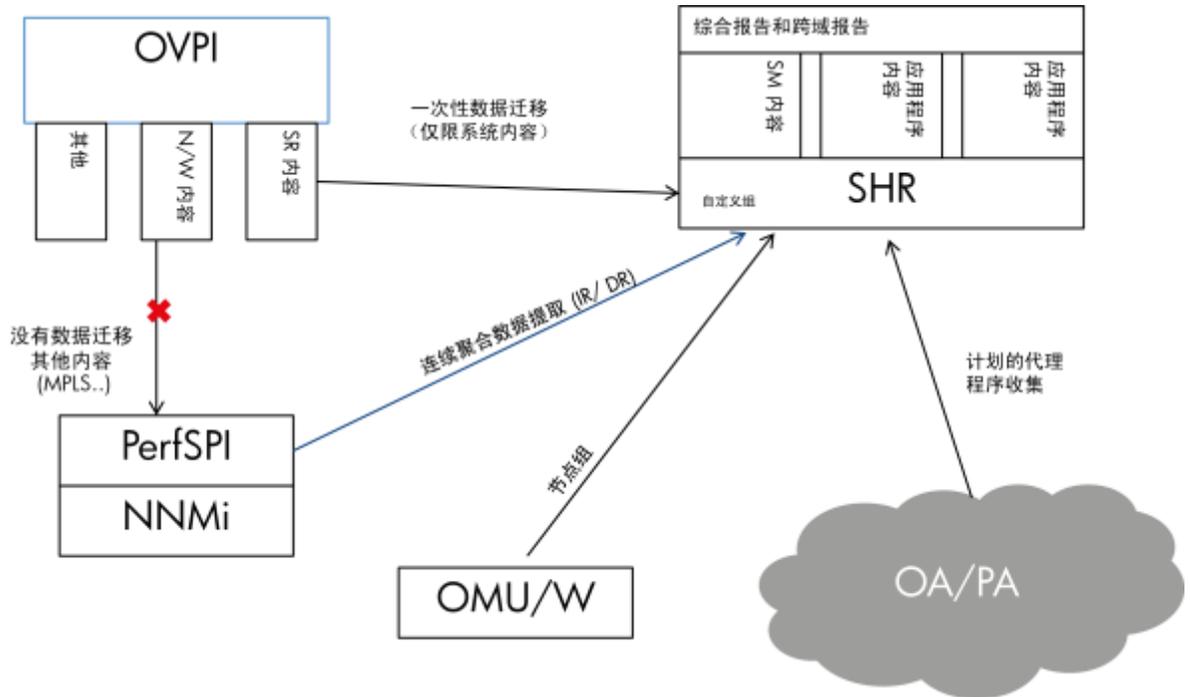
在第 1 阶段中，将 PI 客户迁移到一个或多个 BSM 产品（基于用例）。本节将详细说明用于迁移数据的结构化方法、提出的假设、包括的方案以及已知的限制。

假设

- PI 不是用于进行持续数据收集的数据源（一次性数据迁移方法）
- PI 客户是当前的 OM 或 BSM 用户
- 报告解决方案的大小必须小于 SHR 9.30 版所支持的范围
- SHR 将安装在单独的服务器上（不会在 PI 服务器上共存）
- 仅支持 HP PI 版本 5.3 和 5.4x，不支持将以前的 PI 版本（以及不受支持的 HP PI 配置）作为用于迁移过程的数据源。
- SHR 中的拓扑源配置是一次性配置，需要在迁移前后保持一致。
- 数据迁移包引入了 PI 的维度和事实数据。用户需安装 SHR ETL 包（如 SysPerf_ETL_PerformanceAgent）来实现以下目标
 - 引入拓扑数据（从各拓扑源 - 如，在 RfSM 中基于 OM 和 Business Service 的拓扑中的节点组）
 - 从代理进行持续的实时收集（由于这是一次性数据迁移）

下面几节描述了迁移方法（第 1 阶段）。

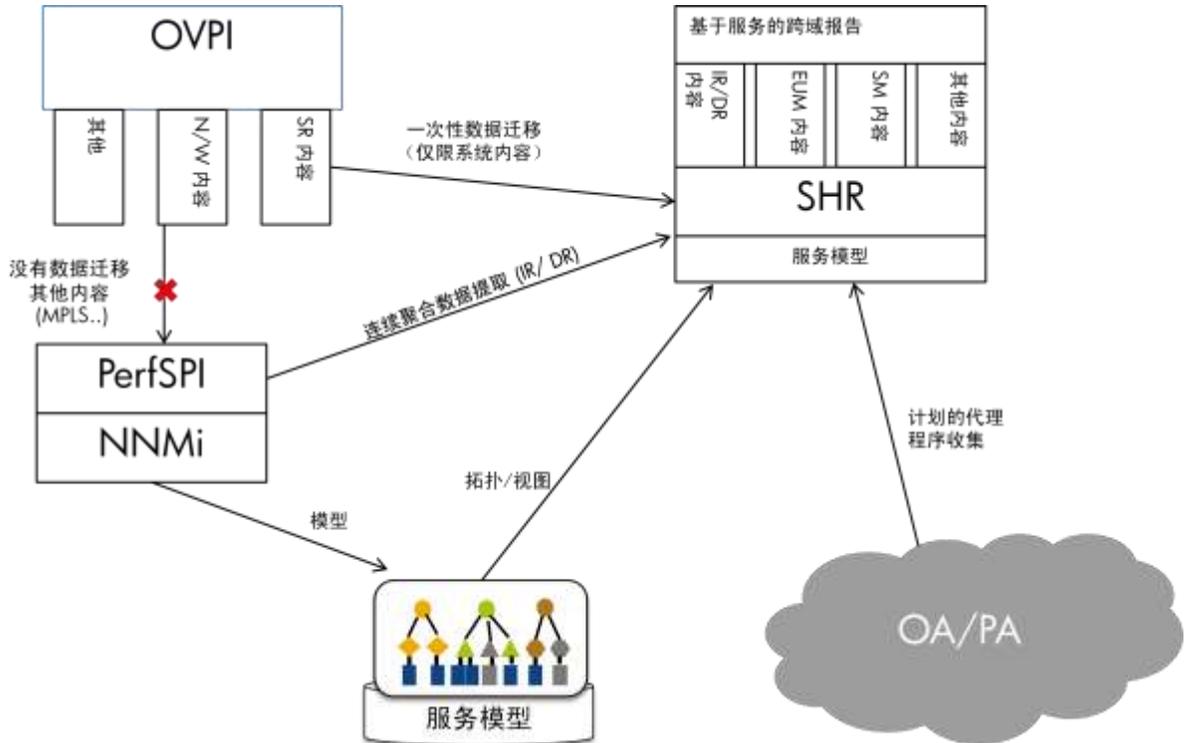
从 Performance Insight 进行的数据迁移 - OM 部署方案



在 OM 部署方案中，数据迁移包从 SHR 中被配置为进行拓扑收集的 OM (Unix/Linux/Windows) 源中获取拓扑信息。对于被配置为拓扑源且受 OM 监控的节点，会将数据从 PI System Resource 内容的每小时和每日表中迁移到 SHR 中。

完成数据迁移之后，需要安装 SysPerf_ETL_PerformanceAgent 以将拓扑/节点组信息和实时比率数据从代理引入到 SHR 中。

从 Performance Insight 进行的数据迁移 - BSM 部署方案



在 BSM 部署方案中，数据迁移包从 SHR 中被配置为进行拓扑收集的 RiSM 实例中获取拓扑信息。对于显示在 RiSM 的 SM_PA 视图中的节点，会将数据从 PI System Resource 内容的每小时和每日表中迁移到 SHR 中。

完成数据迁移之后，需要安装 SysPerf_ETL_PerformanceAgent 以将基于服务的拓扑信息（来自 SHR 视图，如在 RiSM 中部署的 SM_PA）和实时比率数据从代理引入到 SHR 中。

数据迁移方法

数据迁移方法利用了 SHR 的 ETL 和业务流程（审核、平衡及控制 (ABC)）框架。以下内容概述了在每个 ABC 流中发生的数据流：

- 收集拓扑数据
- 构建协调注册表
- 使用 db 收集器从 PI (Oracle 和 Sybase ASE) 提取数据（维度/每小时及每日度量数据）（本文档的“参考”一节中详细说明了包含的 PI 表）。
- 根据拓扑源协调来自 PI 的数据
- 使用加载程序将数据加载到 SHR 聚合表

收集拓扑数据

从 SHR 中已配置的拓扑源（RtSM 和 OM）收集数据。必须为拓扑收集过程构建协调注册表，以用于协调来自 PI 的度量数据。

备注：迁移包中将不包含详细的维度和拓扑收集。在启动实时数据流之后，会将此收集引入迁移后过程。

从 PI 提取数据

支持从 HP PI 所使用的 Oracle/Sybase ASE 数据库提取数据。迁移包将利用 SHR 数据库收集机制来实现此操作。

迁移内容包只需从 PI 数据库的每小时表开始提取数据。由于 SHR 仅提供了 System Performance 的每小时和每日表，因此数据迁移将从其在 PI 的 System Resource 内容包（及其子包）中的对应表开始。PI 中的每年和每月聚合数据及比率数据将不会迁移到 SHR。

在迁移数据之后，可使用足够的初始历史记录在 SHR 中启动比率数据收集，以引入 HP Performance Agent 中的原始度量数据。

处理来自 HP PI 的数据

SHR DB 收集器会转储在 %PMDB_HOME%\collect 文件夹中收集的数据，并需要使用“收集”步骤将其复制到暂存文件夹以供进一步处理。从 PI 收集的数据将不包含拓扑信息（也不包含 CI_UID）。因此，需要将已收集的数据与从拓扑源收集的数据进行协调。

将 HP PI 数据加载到 SHR

SHR 加载程序模块用于加载从 PI 收集的每小时和每日聚合度量数据，以及维度和位置/客户桥接数据。

已知问题和限制

- 可能不会迁移的项包括：
 - 从 PI 引入的数据范围（度量和维度）仅包括可映射到 SHR 中数据模型的数据（例如，不会迁移文件系统数据/预测度量）。本文档的“参考”一节中提供了 PI 和 SHR 架构之间的详细映射列表。
 - 配置详细信息（收集配置、PI 中默认的硬编码移位、报告配置（如计划）等）将不会迁移到 SHR
 - PI 中不支持计划停机功能，因此迁移的数据集中将不包含相应数据。
 - SHR 9.30 不包含客户创建的预置报告。但是，客户维度数据及其与节点维度的关联将会迁移到 SHR（可以使用相同方法创建自定义 Universe/报告）。
- 由于 PI 不支持一致的时区，因此迁移的数据集将在源的时区内可用。在 SHR 9.30 中，用户可以选择将时区设置为 GMT 或本地时区。但是，数据迁移不会在 PI 和 SHR 之间的时区不同时执行时区转换操作。迁移的数据将在源时区中可用，而新收集的数据（在安装 SysPerf_ETL_PerformanceAgent 内容之后）将遵循 SHR 中所配置的时区。

- 可能会使用 SHR 实时收集所引入的维度属性来更新从 PI 引入的维度属性。
- 将收集 PI 中所有可用的每小时和每日数据。当前不能在迁移内容中自定义间隔时间或筛选此类数据。
- SHR 中的历史报告仅会显示 PI 中所配置的保留期内的 PI 迁移数据（每小时/每日）。例如，如果 PI 中每小时数据的保留期仅为 7 天，则从 7 天之前的每日数据向下搜索可能不会获得每小时粒度的数据。
- 数据迁移包仅会引入 PI 中某些节点的度量数据，这些节点是已配置的 OM/RtSM 拓扑源的一部分。
- 可能已根据 PI 中配置的阈值计算从 PI 引入到 SHR 中的异常和服务等级度量。SHR 中实时收集引入的数据将利用在 SHR 中配置为用于计算这些度量的阈值，而在 PI 和 SHR 中配置的阈值可能不一致。
- 仅会将默认移位应用于迁移的数据 - 不应在数据迁移之前在 SHR 中配置移位（移位扩展可能会影响正在迁移的数据的健全性）。在迁移后即可配置移位，这样会为之后收集的数据应用相关的移位。
- 管理 UI 中的通用数据库配置当前存在一个已知问题。如果 Perf iSPI 数据库（或任何其他外部数据库）已配置为通用数据库数据源，并且用于迁移的 PI 也如此，则可针对已部署的收集策略从两个源进行数据收集尝试。在这种情况下，将从正确的源进行收集；但是另外一方面，将在日志文件中记录错误，并且 ABC 流状态将显示为“警告”。

数据迁移步骤

先决条件

在启动从 PI 到 SHR 的 System Performance 内容的数据迁移之前，必须确保以下条件：

- 已停止 PI Timer 服务（并在迁移过程中保持停止状态）。
- 已禁用 SHR Collection 和 Timer 服务
- 未在 SHR 管理 GUI 中配置任何移位（仅存在默认移位）。这一点非常重要，因为 SHR 中配置的移位可能会影响从 PI 加载的聚合数据的健全性。

PI 迁移内容包

可以在介质和 Live Network 上找到 PI 数据迁移包 (HPSHRSmPIMgr.msi)。此包可同时满足 RiSM 和 OM 部署方案的要求，且将依赖于 System Management 的域内容 (SysPerf_Domain)。

在从 PI 迁移数据时执行的步骤

安装迁移包

- 安装 SHR DVD 上的 PI 迁移包（适用于 Windows 系统的 HPSHRSmPIMgr.msi 和适用于 Linux 的 HPSHRSmPIMgr-9.30.000-Linux2.6_64.rpm）。
导航至 DVD 中的包文件夹并执行以下命令，在 Linux 中安装以下 rpm 文件：

```
rpm -ivh HPSHRSmPIMgr-9.30.000-Linux2.6_64.rpm
```

这样会将数据迁移包 (Migration_SM_PI\ Migration_SM_PI.ap) 提取到 %PMDB_HOME%/packages（在 Windows 上）/\$PMDB_HOME/packages（在 Linux 上）文件夹中。
- 使用部署管理器部署包。（在继续进行数据迁移之前，请确保 Collection and Timer 服务没有运行）。

配置数据源

下一步将配置拓扑和 PI 数据库源以用于数据收集。

- 应使用 SHR 管理 UI 中的 RiSM/OM 拓扑源配置页面来配置需在拓扑收集使用的拓扑源详细信息。
- 应在 SHR 管理 UI 中将 HP PI 数据库详细信息配置为“通用数据库”。



- 在此配置中，必须选择域名“**Migration_SM_PI**”，以确保从 PI 数据库收集数据。
 - 收集站选项必须设置为本地。
- (备注：有关在此类情况下的其他配置，请参考本文档的用例部分中的“PI 的分布式部署”)

执行数据迁移脚本

要启动数据迁移，请通过以下命令行执行数据迁移脚本：

在 Windows 上：

```
trend_proc -f %PMDB_HOME%/packages/Migration_SM_PI/Migration_SM_PI.ap/migrate_pi_shr.pro
```

在 Linux 上：

```
trend_proc -f $PMDB_HOME/packages/Migration_SM_PI/Migration_SM_PI.ap/migrate_pi_shr.pro
```

由于 SHR 的数据流业务流程 (ABC) 框架中提供了对数据迁移过程的辅助，因此可以在管理 UI 中查看迁移中各步骤的状态。

相关日志文件

以下日志文件将包含在数据迁移包的安装/执行期间记录的详细信息。这些日志文件位于 %PMDB_HOME%/log 文件夹 (Windows 系统) 和 \$PMDB_HOME/log (Linux 系统) 中。

- pagemanager.log – 使用部署管理器执行操作的迁移包 (及其依赖关系 SysPerf_Domain 包) 的详细部署信息。
- Topologycollector.log – OM/RtSM 拓扑收集详细信息
- reconcileStep.log – 根据已配置的拓扑源中的维度数据进行的协调的详细信息
- dbcollector.log – 关于对 HP PI 数据库的查询执行情况的详细信息，以及关于在生成初始 .csv 文件前的提取阶段中涉及的联接查询的详细信息
- dwablauncher.log – ABC 流执行情况详细信息
- loader.log – 用于将每小时/每日聚合数据加载到 SHR 表的最后步骤的详细信息。

卸载迁移包

在成功迁移数据之后，需卸载迁移包。

可使用部署管理器卸载数据迁移包。

此外，应卸载 PI 迁移包（适用于 Windows 系统的 HPSHRSmPIMgr.msi 和适用于 Linux 的 HPSHRSmPIMgr-9.30.000-Linux2.6_64.rpm）。

执行以下命令可在 Linux 上卸载以下 rpm 文件：

```
rpm -ev HPSHRSmPIMgr-9.30.000-1
```

这是为了确保在完成数据迁移过程之后，迁移包不再作为部署管理器中的可选项。

应删除在 SHR 管理 UI 中配置为“通用数据库”的 HP PI 数据库详细信息。

重新启动 SHR 服务以及安装 System Performance ETL 包

现在可以启用和启动包括 Collection 和 Timer 服务在内的 SHR 服务。

对于 System Performance 内容，请安装 ETL 包 (SysPerf_ETL_PerformanceAgent)，以便从 Performance Agent 收集所需的拓扑数据以及用于生成报告的实时数据。

备注：PI 迁移内容包仅会从 PI 迁移度量和维度数据。只有在已安装相关 ETL 包并运行拓扑收集时，才会从配置的拓扑源收集拓扑/维度。因此，必须首先启动定时器和收集服务，然后才能在报告中查看数据。

典型用例/方案

分布式环境 - 卫星服务器和中央服务器

在此情形下，用户必须在 SHR 管理 UI 中配置 PI 卫星/中央服务器详细信息之后，多次执行迁移流。因此，如果存在 2 个卫星服务器和 1 个中央服务器，则用户必须首先在管理 UI 中配置第 1 个卫星服务器的详细信息，并执行 migrate_pi_shr.pro 文件。在完成流的执行之后，用户必须配置第二个卫星服务器的详细信息并执行此流。用户对每个卫星服务器和中央服务器执行相同的步骤。（对于较小规模的部署，可首先在管理 UI 中配置所有源（中央和卫星），然后执行 migrate_pi_shr.pro 文件）。

备注：如果中央服务器本身可提供每小时数据（取决于客户在 HP PI 中定义的复制策略），则无需从各卫星服务器请求数据。

常见问题 [FAQ]

本节将对数据迁移过程中的一些常见问题进行解答。

问题 1.是否可以在安装并运行 SHR 数月之后再从 PI 迁移数据？如果部署了其他报告包，是否需要延迟迁移活动直到所有必需迁移包都已准备就绪？

回答 1.可以在安装且运行 SHR 一段时间之后开始从 PI 迁移数据。需要注意的是，将从 PI 迁移尚未在 SHR 中收集对应数据的时间段内的数据。例如，如果 SHR 已收集和聚合了过去一周的数据，则仅会将过去一周之前的数据从 PI 加载到 SHR 中。

将以阶段性方式正确地迁移报告包，因此没有必要等待所有数据迁移包准备就绪。

问题 2.是否可以在 SHR 中创建用于报告的自定义节点组？

回答 2.可以在 SHR 中使用 xml 格式定义自定义组，如下所示：

```
<groups>
  <group name="System_customgroup" type="CUSTOMGROUP">
    <instances type="K_CI_System">
      <instance>
        <attribute name="Node_Name" value="node1.ind.hp.com" operator="EQUALS"
          relation="OR" />
        <attribute name="Node_Name" value=" node2.ind.hp.com " operator="EQUALS"
          relation="OR" />
        <attribute name="Node_Name" value=" node3.ind.hp.com " />
      </instance>
    </instances>
  </group>
</groups>
```

- *组名* - 自定义组的名称
 - *类型* - 首选 “CUSTOMGROUP” 或其他不同类型（用以与 OM 中的其他组类型（如 VIEWS 和 NODEGROUPS）进行区分）
 - *实例类型* - 维度表的名称，要进行分组的维度实例。
 - *属性名* - 为其提供实例值的维度表的列名称。
 - *值* - 要为其创建组的列的实际数据值。
- xml 文件名的格式应为 *customgroup.xml，并且应置于 %PMDB_HOME%/config（在 Windows 上）/\$PMDB_HOME/config（在 Linux 上）文件夹中。
 - 请执行以下命令处理 .xml（生成 .csv 文件并将其暂存）：
 - `abcloadNrun -loadBatch -streamId CustomGroup@Platform`
 - `abcloadNrun -runStream -streamId CustomGroup@Platform`
 - 使用以下查询验证是否已将数据加载到暂存表中：
 - `select * from K_CI_Group_Bridge_`
 - 在通过定时器调用时，核心 CP 中的维度流将执行加载程序步骤，并且将相同的内容加载到数据仓库表 K_CI_Group_Bridge 中
 - 请确保 SHR 报告的组选择提示中显示自定义组值。

问题 3.是否可以针对选定的节点/节点组的集合配置数据迁移?

回答 3.当前只能在 SHR 的 OM 拓扑源方案中执行此操作。默认情况下将关闭节点筛选。

- 要启用节点筛选，您需要在位于 %PMDB_HOME%\config（在 Windows 上）或 \$PMDB_HOME/config（在 Linux 上）的 collection.properties 文件中设置以下属性：
`om.filtering.enabled=true`
- 创建一个文件，其中包含需要选择性地导入到 SHR 的节点的列表，即 %PMDB_HOME%\config\filterednodes.conf（在 Windows 上） / \$PMDB_HOME/config/filterednodes.conf（在 Linux 上）文件。此列表应包含在 OM 控制台中显示的节点名称/FQDN。
- 创建一个文件，其中包含 OM 中需要选择性地导入到 SHR 的组的列表（即 %PMDB_HOME%\config\filteredgroups.conf（在 Windows 上） / \$PMDB_HOME%/config/filteredgroups.conf（在 Linux 上）文件）。
- 如果两个文件同时存在并且都有有效值，则属于 filteredgroups.conf 下每个组的所有节点以及 filterednodes.conf 文件中指定的节点都将被导入 SHR 中。

参考

PI 的 System Resource 报告包与 SHR 中的 System Performance 内容之间的数据映射

本节将提供 PI 和 SHR 中的数据库表/列之间的映射。在从 PI 迁移数据之后，未在 SHR 报告中使用的列显示为深红色。

PI: K_Node	SHR: K_CI_System
NODE_NAME	DNS_Name Host_Name Name Display_Name
OPERATING_SYS	OS
MODEL	Model
MAKE	Vendor
MAKE	Manufacture
NODE_TYPE	HyperVisor_Type
SERIAL_NUM	Serial_Number
NODE_ID	
IP_ADDRESS	
DEPARTMENT	
SYSOBJECTID	
IP_STATE	
SYSOBJECTID	
	Host_Key
	OS_Version
	Internal_Name
	Server_Type
	isVirtual (设置为“FALSE”)
	Processor_Architecture
	Phys_Mem_GB
	CPU_Num
	CPU_Num_Core
	CPU_Speed_MHz
	Disk_Num
	Network_Num
	Node_Type
	CPUUtil_Threshold
	SwapUtil_Threshold
	MemUtil_Threshold
	RunQ_Threshold
	PageOut_Threshold
	StaticThresholdFlag
	CPUFamily
	CI_UID (从拓扑源检索 - PI 中的

PI: K_Node	SHR: K_CI_System
	NODE_NAME 用于协调)
	Creation_time
	Created_by
	Update_time
	Updated_by
	Description
	User_Key
	CPUUtil_STH1
	CPUUtil_STH2
	CPUUtil_STH3
	SwapUtil_STH1
	SwapUtil_STH2
	SwapUtil_STH3
	MemoryUtil_STH1
	MemoryUtil_STH2
	MemoryUtil_STH3
	RunQ_STH1
	RunQ_STH2
	RunQ_STH3
	CIT_Key
	Managed_BY
	Disk_Capacity_GB
	Total_Network_Speed
	State
	CPU_Unreserved
	Role
	HyperVisor_Type
	UUID
	ProcessorModel
	CPUCapacityGHZ
	Datacenter
	Standalone_View (设置为“1”)
	OSPatch
	Memory_Unreserved
	Cluster_Name
	Hypervisor_View

PI: SH_SR_SysXcep	SHR: SH_SM_Node_Res
	dsi_key_id (自动生成)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef (自动生成)
AVGrunq	avgRunQ
P95runq	P95RunQ
TOTrunq_grade	totRunQ_Grade
AVGrunq_grade	avgRunQ_Grade
AVGcputil	avgCPUUtil

PI: SH_SR_SysXcep	SHR: SH_SM_Node_Res
TCTcpuutil	tctCPUUtil
P95cpuutil	P95CPUUtil
TOTcpuutil_grade	totCPUUtil_Grade
AVGcpuutil_grade	avgCPUUtil_Grade
AVGmemutil	avgMemUtil
TCTmemutil	tctMemUtil
P95memutil	P95MemUtil
TOTmemutil_grade	totMemUtil_Grade
AVGmemutil_grade	avgMemUtil_Grade
AVGswapUtil	avgSwapUtil
TCTswapUtil	tctSwapUtil
P95swapUtil	P95SwapUtil
TOTswapUtil_grade	totSwapUtil_Grade
AVGswapUtil_grade	avgSwapUtil_Grade
AVGmemPageOutRate	avgMemPageOutRate
TCTmemPageOutRate	tctMemPageOutRate
P95memPageOutRate	P95MemPageOutRate
TOTavgNumProcs	totNumProcs
AvgNumProcs	avgNumProcs
PI: SH_SR_SysVolXcep	SHR: SH_SM_Node_Res
TOTInPackets	totInPackets
TOTOutPackets	totOutPackets
TOTvolume	totVolume
	maxVolume
WAVvolume	avgVolume
	avgPacketRate
	maxPacketRate
AVGCollisionRate	avgCollisionRate
	maxCollisionRate
AVGErrorRate	avgErrorRate
	maxMemUtil
	P90MemUtil
	avgFreeMemGB
	maxRunQ
	P90RunQ
	maxCPUUtil
	P90CPUUtil
	maxSwapUtil
	P90SwapUtil
	P90MemPageOutRate
	avgDiskPhysIORate
	maxDiskPhysIORate
	P90DiskPhysIORate
	avgNetIORate
	maxNetIORate
	P90NetIORate
	totNumStartedProcs
	totNumActiveProcs

PI: SH_SR_SysXcep	SHR: SH_SM_Node_Res
	maxErrorRate
	avgSysModeUtil
	maxSysModeUtil
	avgUsrModeUtil
	maxUsrModeUtil
	avgCSwitchRate
	maxCSwitchRate
	avgInterruptRate
	maxInterruptRate
	avgByteRate
	ubsavgMemoryUtil
	lbsavgMemoryUtil
	ubsavgRunQ
	lbsavgRunQ
	ubsavgCPUUtil
	lbsavgCPUUtil
	ubsavgSwapUtil
	lbsavgSwapUtil
	ubsavgDiskPhysIORate
	lbsavgDiskPhysIORate
	ubsByteRate
	lbsByteRate
	avgNetworkUtilMbps
	NetInByteRate
	NetOutByteRate
	avgDiskPhysreadbyteRate
	avgDiskPhyswritebyteRate
	avgReadlatency
	avgWritelatency
	VMNum
	DiskSpaceUtil

PI: SD_SR_SysXcep	SHR: SD_SM_Node_Res
	dsi_key_id (自动生成)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef (自动生成)
AVGrunq	avgRunQ
P95runq	P95RunQ
TOTrunq_grade	totRunQ_Grade
AVGrunq_grade	avgRunQ_Grade
AVGcpuutil	avgCPUUtil
TCTcpuutil	tctCPUUtil
P95cpuutil	P95CPUUtil
TOTcpuutil_grade	totCPUUtil_Grade
AVGcpuutil_grade	avgCPUUtil_Grade
AVGmemutil	avgMemUtil

PI: SD_SR_SysXcep	SHR: SD_SM_Node_Res
TCTmemutil	tctMemUtil
P95memutil	P95MemUtil
TOTmemutil_grade	totMemUtil_Grade
AVGmemutil_grade	avgMemUtil_Grade
AVGswapUtil	avgSwapUtil
TCTswapUtil	tctSwapUtil
P95swapUtil	P95SwapUtil
TOTswapUtil_grade	totSwapUtil_Grade
AVGswapUtil_grade	avgSwapUtil_Grade
AVGmemPageOutRate	avgMemPageOutRate
TCTmemPageOutRate	tctMemPageOutRate
P95memPageOutRate	P95MemPageOutRate
TOTavgNumProcs	totNumProcs
AvgNumProcs	avgNumProcs
PI: SD_SR_SysVolXcep	SHR: SD_SM_Node_Res
TOTInPackets	totInPackets
TOTOutPackets	totOutPackets
TOTvolume	totVolume
	maxVolume
WAVvolume	avgVolume
	avgPacketRate
	maxPacketRate
AVGCollisionRate	avgCollisionRate
	maxCollisionRate
AVGErrorRate	avgErrorRate
	maxMemUtil
	P90MemUtil
	avgFreeMemGB
	maxRunQ
	P90RunQ
	maxCPUUtil
	P90CPUUtil
	maxSwapUtil
	P90SwapUtil
	P90MemPageOutRate
	avgDiskPhysIORate
	maxDiskPhysIORate
	P90DiskPhysIORate
	avgNetIORate
	maxNetIORate
	P90NetIORate
	totNumStartedProcs
	totNumActiveProcs
	maxErrorRate
	avgSysModeUtil
	maxSysModeUtil
	avgUsrModeUtil
	maxUsrModeUtil

PI: SD_SR_SysXcep	SHR: SD_SM_Node_Res
	avgCSwitchRate
	maxCSwitchRate
	avgInterruptRate
	maxInterruptRate
	avgByteRate
	ubsavgMemoryUtil
	lbsavgMemoryUtil
	ubsavgRunQ
	lbsavgRunQ
	ubsavgCPUUtil
	lbsavgCPUUtil
	ubsavgSwapUtil
	lbsavgSwapUtil
	ubsavgDiskPhysIORate
	lbsavgDiskPhysIORate
	ubsByteRate
	lbsByteRate
	avgNetworkUtilMbps
	NetInByteRate
	NetOutByteRate
	avgDiskPhysreadbyteRate
	avgDiskPhyswritebyteRate
	avgReadlatency
	avgWritelatency
	VMNum

PI: SH_SR_SysUp	SHR: SH_SM_Node_Avail
	dsi_key_id (自动生成)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef (自动生成)
TOTUPTIME	totuptime
TOTDOWNTIME	totdowntime
	Totplandtime
	Totexcdtime
	Totunknowntime
	Totavailability

PI: SD_SR_SysUp	SHR: SD_SM_Node_Avail
	dsi_key_id (自动生成)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef (自动生成)
TOTUPTIME	totuptime
TOTDOWNTIME	totdowntime
	Totplandtime

PI: SD_SR_SysUp	SHR: SD_SM_Node_Avail
	Totexcdtime
	Totunknowntime
	Totavailability

PI: K_System_CPU	SHR: K_SM_CPU
	dsi_key_id (自动生成)
CPUID	cpu_id
(K_Node.node_name(K_System_CPU.node_fk))	node_name
	Systemref (自动生成 (使用 ciid - 使用 node_name 协调))
	cpu_vendor
	cpu_speed

PI: SH_SR_CPU	SHR: SH_SM_CPU
	dsi_key_id (自动生成)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef (自动生成)
AVGCPUUTIL	avgTotUtil
AVGCPUSYSTEMMODE	avgSysModeUtil
AVGCPUUSERMODE	avgUsrModeUtil
AVGINTRATE	avgInterruptRate
AVGCSRATE	avgCntxtSwitchRate
	maxTotUtil
	maxSysModeUtil
	maxUsrModeUtil
	maxInterruptRate
	maxCntxtSwitchRate

PI: SD_SR_CPU	SHR: SD_SM_CPU
	dsi_key_id (自动生成)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef (自动生成)
AVGCPUUTIL	avgTotUtil
AVGCPUSYSTEMMODE	avgSysModeUtil
AVGCPUUSERMODE	avgUsrModeUtil
AVGINTRATE	avgInterruptRate
AVGCSRATE	avgCntxtSwitchRate
	maxTotUtil
	maxSysModeUtil
	maxUsrModeUtil
	maxInterruptRate

PI: SD_SR_CPU	SHR: SD_SM_CPU
	maxCntxtSwitchRate

PI: K_Disk_Dsk	SHR: K_SM_PhysicalDISK
	dsi_key_id (自动生成)
prop_disk_name	disk_name
(K_Node.node_name(K_System_CPU.node_fk)	node_name
	Systemref (自动生成 (使用 ciid - 使用 node_name 协调))
	Dir_name

PI: SH_SR_Disk	SHR: SH_SM_Disk
	dsi_key_id (自动生成)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef (自动生成)
AVGDISKUTIL	avgPctUtil
AVGPHYSICALIORATE	avgPhyIORate
AVGPHYSREADRATE	avgPhyReadRate
AVGPHYSWRITERATE	avgPhyWriteRate
AVGSYSTEMIORATE	avgSysIORate
AVGVMIORATE	avgVMIORate
	avgPhyByteRate
	avgPhyReadByteRate
	avgPhyWriteByteRate
	avgRawReadRate
	avgRawWriteRate
	maxPctUtil
	maxPhyIORate
	maxPhyReadRate
	maxPhyWriteRate
	maxSysIORate
	maxVMIORate
	maxPhyByteRate
	maxPhyReadByteRate
	maxPhyWriteByteRate
	maxRawReadRate
	maxRawWriteRate

PI: SD_SR_Disk	SHR: SD_SM_Disk
	dsi_key_id (自动生成)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef (自动生成)
AVGDISKUTIL	avgPctUtil
AVGPHYSICALIORATE	avgPhyIORate

PI: SD_SR_Disk	SHR: SD_SM_Disk
AVGPHYSREADRATE	avgPhyReadRate
AVGPHYSWRITERATE	avgPhyWriteRate
AVGSYSTEMIORATE	avgSysIORate
AVGVMIORATE	avgVMIORate
	avgPhyByteRate
	avgPhyReadByteRate
	avgPhyWriteByteRate
	avgRawReadRate
	avgRawWriteRate
	maxPctUtil
	maxPhyIORate
	maxPhyReadRate
	maxPhyWriteRate
	maxSysIORate
	maxVMIORate
	maxPhyByteRate
	maxPhyReadByteRate
	maxPhyWriteByteRate
	maxRawReadRate
	maxRawWriteRate

PI: K_NetInterface_NetIf	SHR: K_SM_NetInterface
	dsi_key_id (自动生成)
prop_netif_name	Interface_name
(K_Node.node_name(K_System_CPU.node_fk)	Node_Name
	Systemref (自动生成 (使用 ciid - 使用 node_name 协调))
	Network_speed

PI: SH_SR_NetInterface	SHR: SH_SM_NetInterface
	dsi_key_id (自动生成)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef (自动生成)
	avgPktRate
AVGINPACKETS	avgInPktRate
AVGOUTPACKETS	avgOutPktRate
	avgByteRate
AVGINBYTES	avgInByteRate
AVGOUTBYTES	avgOutByteRate
AVGCOLLISIONRATE	avgCollisionRate
AVGERRORRATE	avgErrorRate
	NetworkUtil
	maxPktRate
	maxInPktRate
	maxOutPktRate

PI: SH_SR_NetInterface	SHR: SH_SM_NetInterface
	maxByteRate
	maxInByteRate
	maxOutByteRate
	maxCollisionRate
	maxErrorRate

PI: SD_SR_NetInterface	SHR: SD_SM_NetInterface
	dsi_key_id (自动生成)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef (自动生成)
	avgPktRate
AVGINPACKETS	avgInPktRate
AVGOUTPACKETS	avgOutPktRate
	avgByteRate
AVGINBYTES	avgInByteRate
AVGOUTBYTES	avgOutByteRate
AVGCOLLISIONRATE	avgCollisionRate
AVGERRORRATE	avgErrorRate
	NetworkUtil
	maxPktRate
	maxInPktRate
	maxOutPktRate
	maxByteRate
	maxInByteRate
	maxOutByteRate
	maxCollisionRate
	maxErrorRate

PI: K_Customer	SHR: K_Customer
	dsi_key_id (自动生成)
cust_id	Customer_ID
cust_name	Name Display_Name
	User_Key
	Address1
	Address2
	City
	State
	ZIP_Code
	Phone_Number
	Description

PI: K_SR_System, K_Node, K_Customer	SHR: K_CI_Cust_Bridge
	dsi_key_id (自动生成)
K_SR_System.node_fk (协调以获取 CI_UID)	CI_Key
K_Customer.cust_name	Cust_Key
	Valid_Period_Start
	Valid_Period_End

PI: K_Location	SHR: K_Location
	dsi_key_id (自动生成)
	Country (默认值)
	State (默认值)
	City (默认值)
location_name	Region
	Address (默认值)
	Building (默认值)
	Floor (默认值)
location_name	Name

PI: K_SR_System, K_Node, K_Location	SHR: K_CI_Loc_Bridge
	dsi_key_id (自动生成)
K_SR_System.node_fk (协调以获取 CI_UID)	CI_Key
K_Location.location_name	Loc_Key
	Valid_Period_Start
	Valid_Period_End

© 版权所有 2013 Hewlett-Packard Development Company, L.P. 本文包含的信息如有更改，恕不另行通知。HP 产品和服务随附的明示性保证声明中列出了适用于此类产品和服务的专用担保条款。本文的任何内容均不构成额外担保。HP 不会为此处出现的技术或编辑错误或遗漏承担任何责任。

