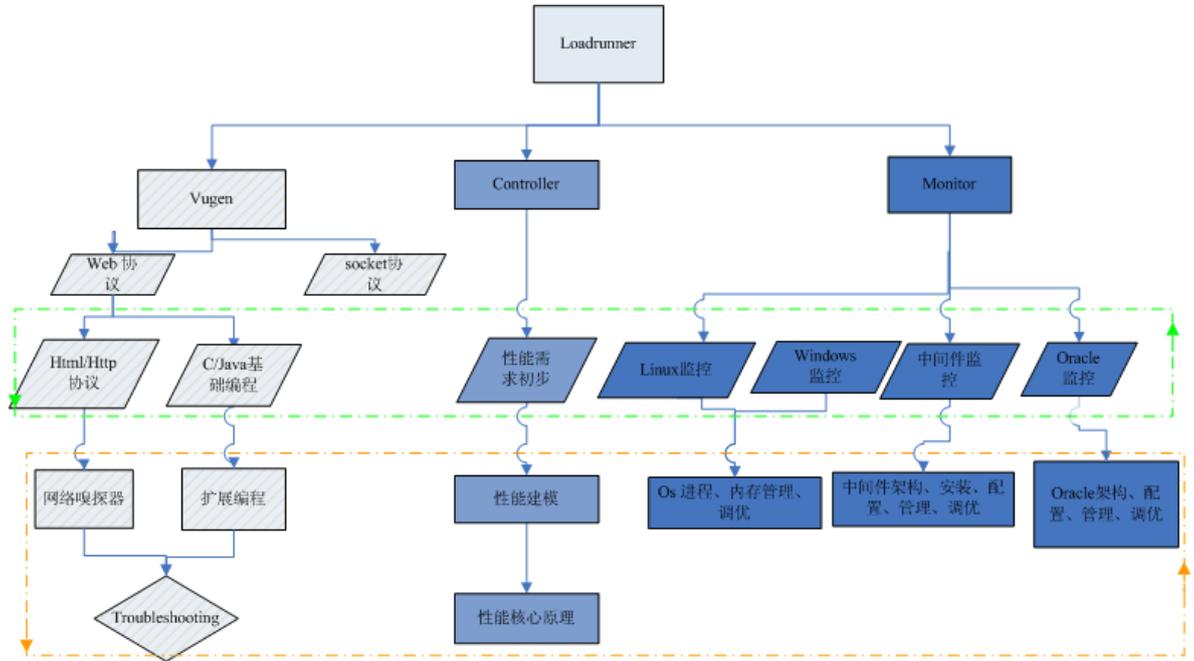


性能测试学习线路图(建议)

1 概览



纵向划分 3 颗子树:vugen,controller,monitor。优先学习 vugen 脚本开发以及调试。
横向划分为 2 层：基础知识以及高级应用。

2 基础知识

2.1 Loadrunner 工具使用

2.1.1 建议学习路径

Vugen 开发脚本（函数使用）->controller 场景设置->monitor 增加计数器

<http://www.cnblogs.com/jackei/archive/2006/10/20/534684.html>

2.1.2 Loadrunner 认证



更多见

http://www.51testing.net/BWF_DIY/mercury/mercury_051107_1.htm

http://www.51testing.net/BWF_DIY/mercury/mercury_060104_11.htm

2.1.3 Vugen 常用增强函数

增加事务: `lr_start_transaction/``lr_end_transaction`

检查点: `web_reg_find`

关联: [web_reg_save_param](#) , [web_set_max_html_param_len](#)

日志: [lr_error_message](#), [lr_log_message](#), [lr_output_message](#) ...

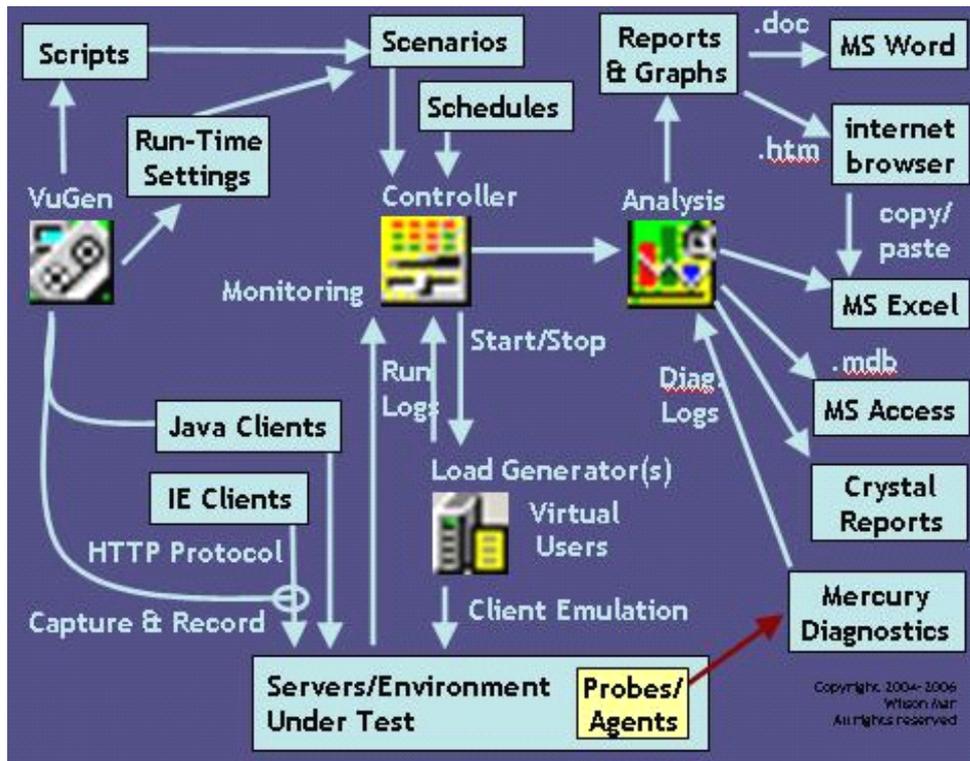
选项设置: [web_set_timeout](#)

http header: [web_add_auto_header](#)

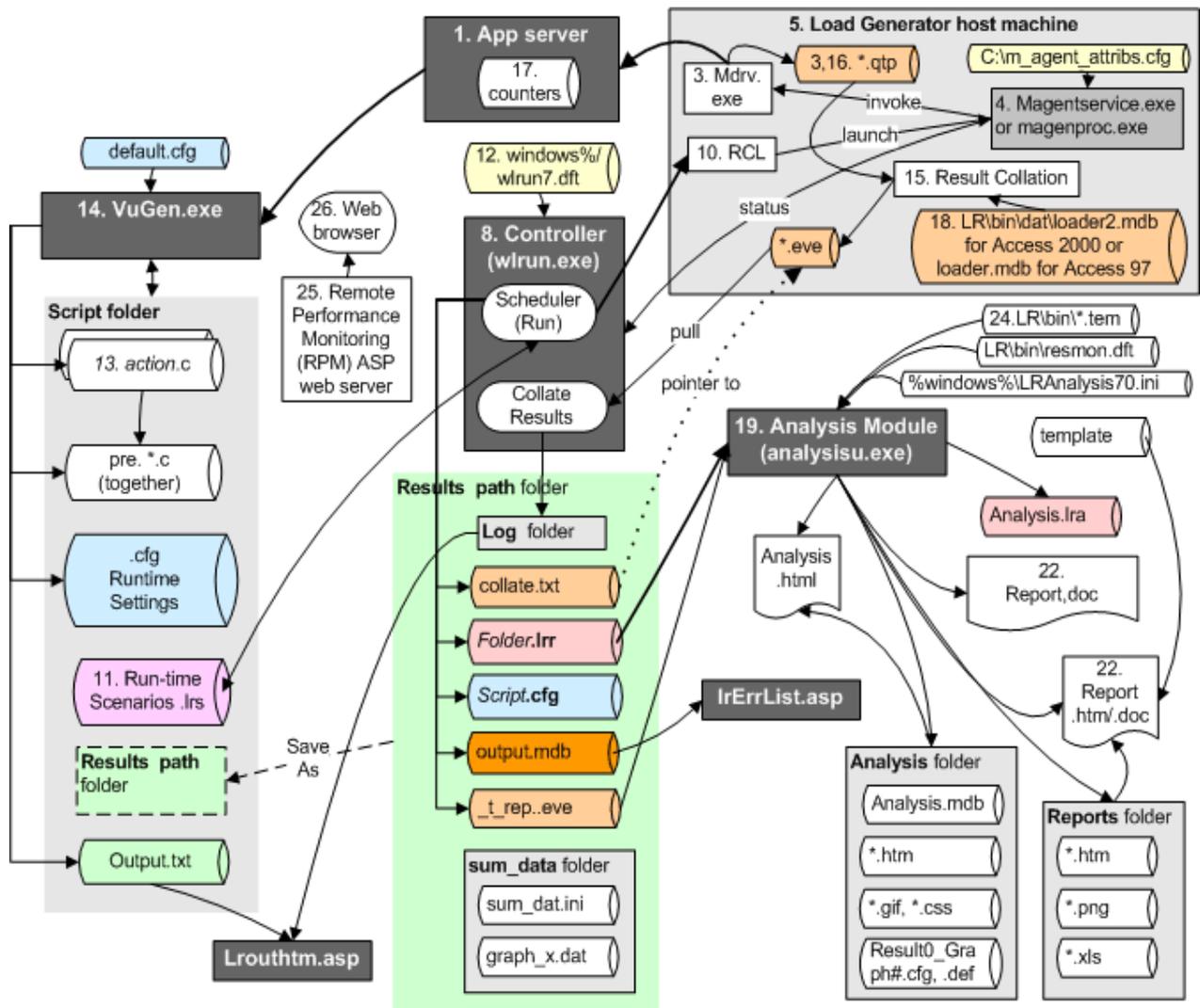
以及更改 runtime setting。

2.1.4 Loadrunner 架构图

概览图



详细架构图:



2.2 Html/http 协议

HTML 4.01 Specification:

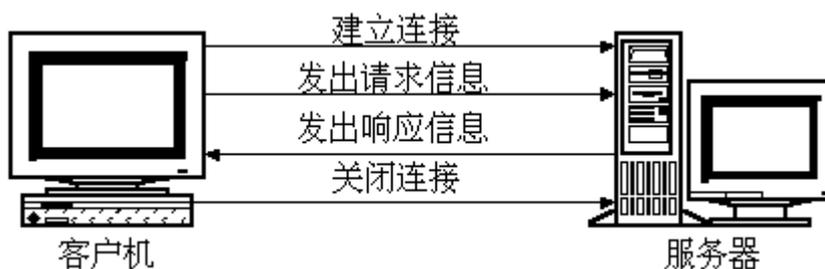
<http://www.w3.org/TR/html4/>

http://www.eygle.com/digest/2006/12/html_xml_and_internet.html

对性能测试而言，META http-equiv 控制 http 头、浏览器行为，故最为重要。

http 协议:

<http://www.faqs.org/rfcs/rfc2616.html>



客户端发送请求:

HTTP/1.0 请求消息的格式为:

请求消息=请求行(通用信息|请求头|实体头)CRLF[实体内容]

请求行=方法 请求URL HTTP版本号 CRLF

方 法=GET|HEAD|POST|扩展方法

U R L=协议名称+宿主名+目录与文件名

请求行中的方法描述指定资源中应该执行的动作，常用的方法有GET、HEAD和POST。不同的请求对象对应GET的结果是不同的，对应关系如下:

对象	GET的结果
文件	文件的内容
程序	该程序的执行结果
数据库查询	查询结果

HEAD——要求服务器查找某对象的元信息，而不是对象本身。

POST——从客户机向服务器传送数据，在要求服务器和CGI做进一步处理时会用到POST方法。POST主要用于发送HTML文本中FORM的内容，让CGI程序处理。

一个请求的例子为:

GEThttp://networking.zju.edu.cn/zju/index.htmHTTP/1.0

头信息又称为元信息，即信息的信息，利用元信息可以实现有条件的请求或应答。

请求头——告诉服务器怎样解释本次请求，主要包括用户可以接受的数据类型、压缩方法和语言等。

实体头——实体信息类型、长度、压缩方法、最后一次修改时间、数据有效期等。

实体——请求或应答对象本身。

服务器响应:

HTTP/1.0的响应消息格式如下：

响应消息=状态行(通用信息头|响应头|实体头) CRLF (实体内容)

状态行=HTTP版本号 状态码 原因叙述

状态码表示响应类型

1×× 保留

2×× 表示请求成功地接收

3×× 为完成请求客户需进一步细化请求

4×× 客户错误

5×× 服务器错误

响应头的信息包括：服务程序名，通知客户请求的URL需要认证，请求的资源何时能使用。

2.3 C 语言基础编程

Loadrunner 常用的 c 函数主要集中在内存分配/释放、字符串操作、文件读写。

如: malloc/free

Sprintf/stremp/strlen/

Fopen/fread/fwrite

2.4 Linux 性能分析初步

核心指令：

Top

Sar

Vmstat

Iostat

Sar -n DEV

/proc 文件系统

分析经验性步骤：

首先查看 CPU 使用情况，按照诊断 CPU、内存或磁盘瓶颈的指导进行操作。对于下面的每个步骤，查找一端时间内的趋势，从中收集系统运行性能较差时的数据。另外，只有将这些数据与系统正常运行时收集的数据进行比较时才能进行准确的诊断。

步骤 1

```
# sar -u [interval] [iterations]
```

(示例: sar -u 5 30)

%idle 是否很低? 这是 CPU 未在运行任何进程的时间百分比。在一端时间内 %idle 为零可能是 CPU 瓶颈的第一个指示。

不是 -> 系统未发生 CPU 瓶颈。转至步骤 3。

是 -> 系统可能发生了 CPU、内存或 I/O 瓶颈。转至步骤 2。

步骤 2

%usr 是否较高? 很多系统正常情况下花费 80% 的 CPU 时间用于用户, 20% 用于系统。其他系统通常会使用 80% 左右的用户时间。

不是 -> 系统可能遇到 CPU、内存或 I/O 瓶颈。转至步骤 3。

是 -> 系统可能由于用户进程遇到 CPU 瓶颈。转至部分 3, 部分 A, 调整系统的 CPU 瓶颈。

步骤 3

%wio 的值是否大于 15? (不同 os 有不同的阈值)

是 -> 以后记住这个值。它可能表示磁盘或磁带瓶颈。转至步骤 4。

不是 -> 转至步骤 4。

步骤 4

```
# sar -d [interval] [iterations]
```

用于任何磁盘的 %busy 是否都大于 50? (请记住, 50% 指示一个大概的指南, 它可能远远高于您系统的正常值。在某些系统上, 甚至 %busy 值为 20 可能就表示发生了磁盘瓶颈, 而其他系统正常情况下可能就为 50% busy。)

对于同一个磁盘上, avwait 是否大于 avserv?

不是 -> 很可能不是磁盘瓶颈, 转至步骤 6。

是 -> 此设备上好像发生了 IO 瓶颈。

转至步骤 5。

步骤 5

系统上存在磁盘瓶颈, 发生瓶颈的磁盘上有哪些内容?

原始分区,

文件系统 -> 转至部分 3, 部分 B, 调整发生磁盘 IO 瓶颈的系统。

Swap -> 可能是由于内存瓶颈导致的。

转至步骤 6。

步骤 6

```
# vmstat [interval] [iterations]
```

在很长的一端时间内, po 是否总是大于 0?

对于一个 s800 系统 (free * 4k) 是否小于 2 MB,

(对于 s700 系统 free * 4k 是否小于 1 MB)?

(值 2 MB 和 1 MB 指示大概的指南, 真正的 LOTSFREE 值, 即系统开始发生 paging 的值是在系统引导时计算的, 它是基于系统内存的大小的。)

不是 -> 如果步骤 1 中的 %idle 较低, 系统则很可能发生了 CPU 瓶颈。
转至部分 3, 部分 A, 调整发生了 CPU 瓶颈的系统。

如果 %idle 不是很低, 则可能不是 CPU、磁盘 IO 或者内存瓶颈。
请转至部分 4, 其他瓶颈。

是 -> 系统上存在内存瓶颈, 转至部分 3 部分 C, 调整发生内存瓶颈的系统。

2.5 Windows 性能分析初步

同 windows perfmon。

同样集中在 cpu,内存, io,网络上。

一般经验值:

网络

网络利用率阈值没有统一。 <30% or 80%?

冲突率: <1%

Packets Received Errors < 1%

I/O:

Disk Time % <90%

Avg. Disk Bytes/Read + Avg. Disk Bytes/Write <20K

Avg. Disk sec/Transfer <0.3 sec

队列长度: Queue Length <2

Avg. Disk sec/Transfer <18 milliseconds

内存

Available Mbytes >25%

Page in+out <20 次

内存泄露以及错误:

Pool Nonpaged Bytes : an increase of 10 percent or more from its value at system startup.

Server -> Pool Nonpaged Failures shows the number of times allocations from nonpaged pool have failed - indicates that the computer's physical memory is too small. 应为 0

Server -> Pool Paged Failures indicate that either physical memory or a paging file is near capacity. 应为 0

Server -> Pool Nonpaged Peak shows the maximum number of bytes in nonpaged pool the server has had in use at any one point. Indicates how much physical memory the computer should have.

处理器

利用率 <85%

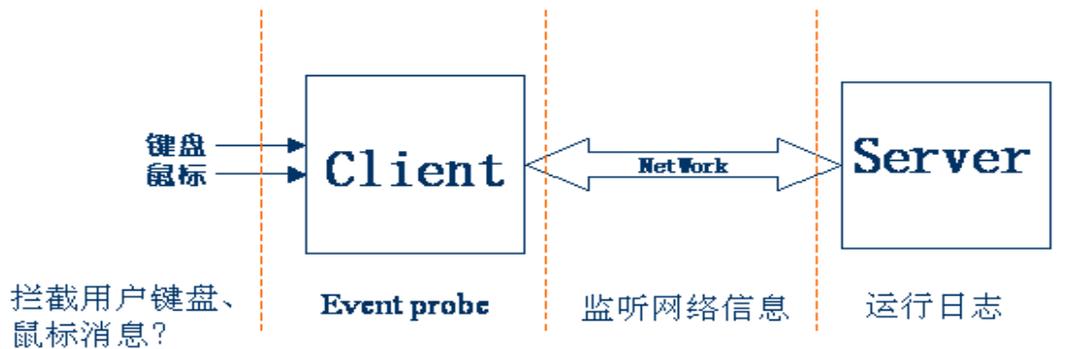
每个 CPU 队列长度 <2

Context Switches/sec <5000 次 或者 <5% of total threads

3 高级应用

3.1 性能建模

从 business layer、function layer、session layer、customer layer 出发，借助日志分析工具挖掘系统负载模型、用户行为模型。



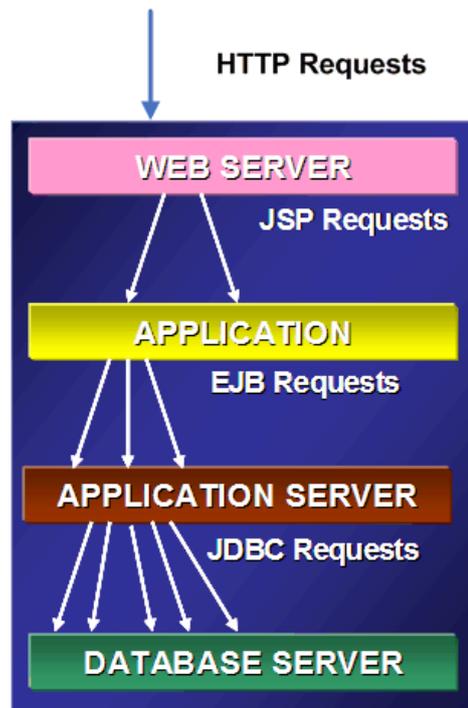
Mercury End User Management 正是一个从 client 视觉出发的业务监控解决方案。

3.2 网络嗅探器

如 ethereal, fiddler, 以及 http debugger 工具，诊断网络问题。

3.3 Loadrunner troubleshooting

各个环节都可能有问题。（client, 通信，服务器）



经验性步骤:

诊断 vugen 脚本、loadrunner 场景设置、客户端硬件资源是否充足

判断服务器端各层是否达到资源上限

借助 ethereal 等网络嗅探器判断 c/s 通信问题

3.4 Linux 性能分析与调优

深入了解:

进程/多线程编程

同步机制

Gdb 调试 core

内核参数

3.5 Windows 性能分析与调优

Windows 其他计数器

3.6 C 语言调用 dll 扩展测试脚本

Dll 编写格式:

如

```
int __stdcall lr_getLinuxIOwait_end()
{
    clnt_destroy(clnt);
}
```

.def 文件

```
EXPORTS
```

```
lr_getLinuxIOwait_end @3
```

loadrunner 调用 dll 函数:

```
lr_load_dll("my.dll");
```

3.7 Apache/jboss/weblogic/oracle 性能优化

参见各个产品 tuning 文档

<http://httpd.apache.org/docs/2.0/misc/perf-tuning.html>

<http://httpd.apache.org/docs/1.3/misc/perf-tuning.html>

<http://edocs.bea.com/wls/docs81/perform/>

<http://wiki.jboss.org/wiki/Wiki.jsp?page=JBossASTuningSliming>

<http://www.jboss.org/index.html?module=bb&op=viewforum&f=121>

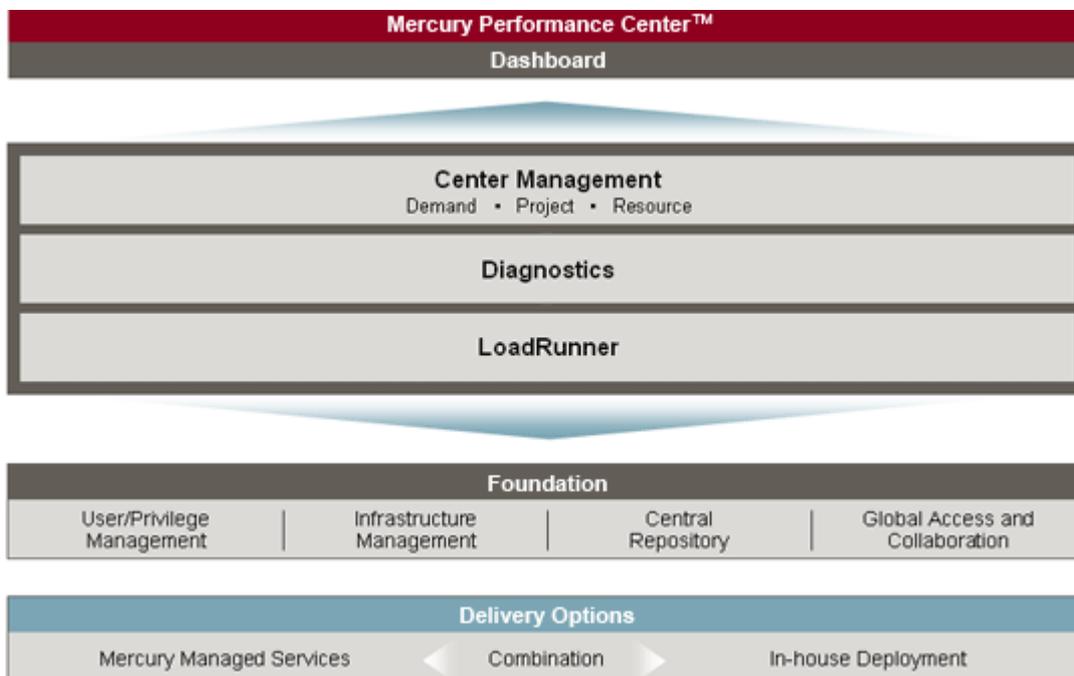
3.8 计算机性能评测研究方向

- 相关理论研究（泊松分布、排队论、MaKov 模型、Monte carlo 模拟、自相似理论）
- 负载特性 Workload Characteristic 的研究（Commercial Workload/ Technical Workload）
- 基准程序 Benchmark 的研究
- 性能指标的研究（全生命周期/高效能/TPO Total Productivity Ownership/**SLA Service Level Agreement**/QoS）
- 模拟器的研究（SimpleScalar/SimOS/SandOS）
- 测试系统的研究（Benchmark Factory/ServerScope/Benchmark Studio/LoadRunner/Forecast toolset）
- 监控系统 Performance Evaluation（Intel Vtune/ EMon/ TeamQuest Lite/ ServerScope-Monitor/ Grid-View）
- 性能评测与体系结构的结合（Computer architecture analysis using workloads）

参见 <http://www.cnblogs.com/jackei/archive/2006/11/20/565527.html>

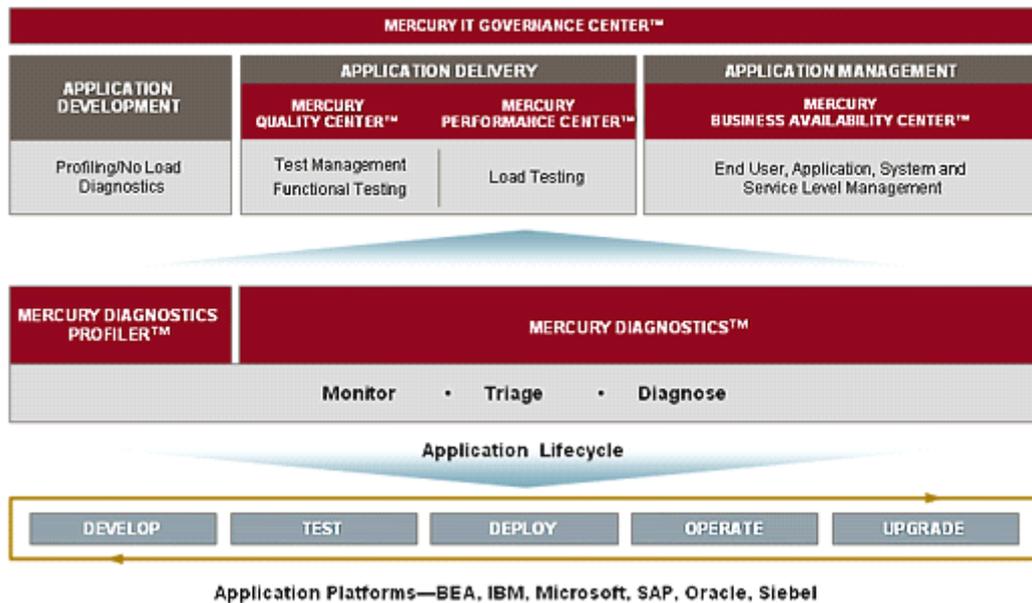
3.9 Mercury performance center

<http://www.mercury.com/us/products/performance-center/>



Mercury Diagnostics

For J2EE, .NET, ERP/CRM



4 性能测试书籍

- 《Microsoft .NET Web 应用程序性能测试》
- 《软件性能测试过程详解与案例剖析》
- 《WEB 性能测试实战》
- 《J2EE 性能测试》
- 《LoadRunner 性能测试实战》

5 测试网站

<http://www.51testing.com/>

<http://www.17testing.com/>

<http://www.wilsonmar.com/1loadrun.htm>

http://www.avoka.com/resources/white_papers.shtml

<http://www.qaforums.com>

<http://support.mercury.com>