

# SOFAStack

## 全链路压测 运维指南

产品版本：AntStack Plus 1.11.0

文档版本：20221008



# 法律声明

蚂蚁集团版权所有©2022，并保留一切权利。

未经蚂蚁集团事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

## 商标声明

 蚂蚁集团 ANT GROUP 及其他蚂蚁集团相关的商标均为蚂蚁集团所有。本文档涉及的第三方的注册商标，依法由权利人所有。

## 免责声明

由于产品版本升级、调整或其他原因，本文档内容有可能变更。蚂蚁集团保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在蚂蚁集团授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过蚂蚁集团授权渠道下载、获取最新版的用户文档。如因文档使用不当造成的直接或间接损失，本公司不承担任何责任。

# 通用约定

| 格式   | 说明                                 | 样例  |
|--|------------------------------------|---|
|  危险   | 该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。   |  危险<br>重置操作将丢失用户配置数据。          |
|  警告   | 该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。 |  警告<br>重启操作将导致业务中断，恢复业务时间约十分钟。 |
|  注意   | 用于警示信息、补充说明等，是用户必须了解的内容。           |  注意<br>权重设置为0，该服务器不会再接受新请求。    |
|  说明 | 用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。       |  说明<br>您也可以通过按Ctrl+A选中全部文件。  |
| >  | 多级菜单递进。                            | 单击设置> 网络> 设置网络类型。   |
| 粗体   | 表示按键、菜单、页面名称等UI元素。                 | 在结果确认页面，单击确定。   |
| Courier字体  | 命令或代码。                             | 执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。  |
| 斜体   | 表示参数、变量。                           | <code>bae log list --instanceid</code><br><code>Instance_ID</code>  |
| [ ] 或者 [a b]   | 表示可选项，至多选择一个。                      | <code>ipconfig [-all -t]</code>   |
| { } 或者 {a b}   | 表示必选项，至多选择一个。                      | <code>switch {active stand}</code>  |

# 目录

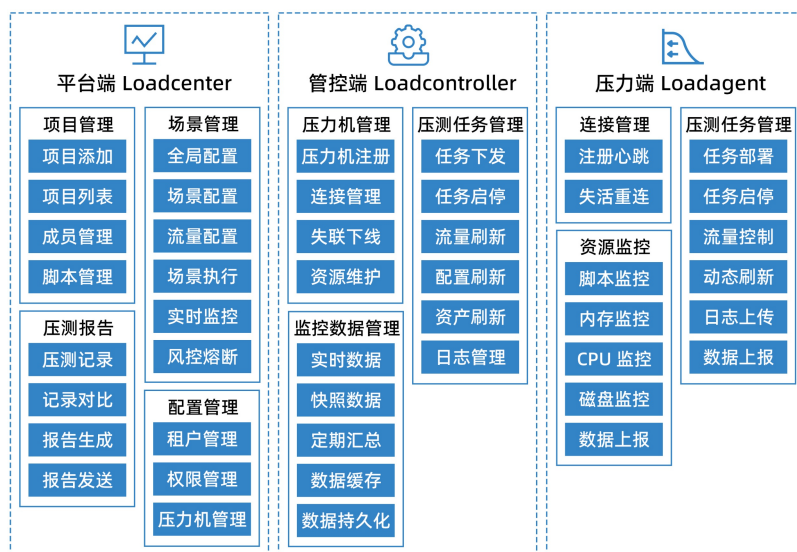
|           |    |
|-----------|----|
| 1. 运维概述   | 05 |
| 1.1. 功能架构 | 05 |
| 1.2. 系统架构 | 05 |
| 2. 运维准备   | 08 |
| 3. 应急运维   | 09 |
| 4. 日常运维   | 12 |
| 5. 常见故障处理 | 14 |
| 6. 参考日志   | 16 |
| 7. 基础术语   | 19 |

# 1. 运维概述

## 1.1. 功能架构

全链路压测平台（Loadcenter）是 SOFAShark 下的压测管理和执行平台，提供涵盖性能压测、压测报告输出以及压测风险防控的一站式压测服务。作为一个企业级产品，全链路压测平台帮助发现业务系统的性能问题，为企业进行系统性能评估、容量规划、性能基线跟踪、风险发现提供易用、稳定、安全的平台支撑。

全链路压测平台历经几代产品打磨并结合蚂蚁集团多年的线上全链路压测经验，具有复杂场景建模、报告分析、多协议支持、风险防控等功能。



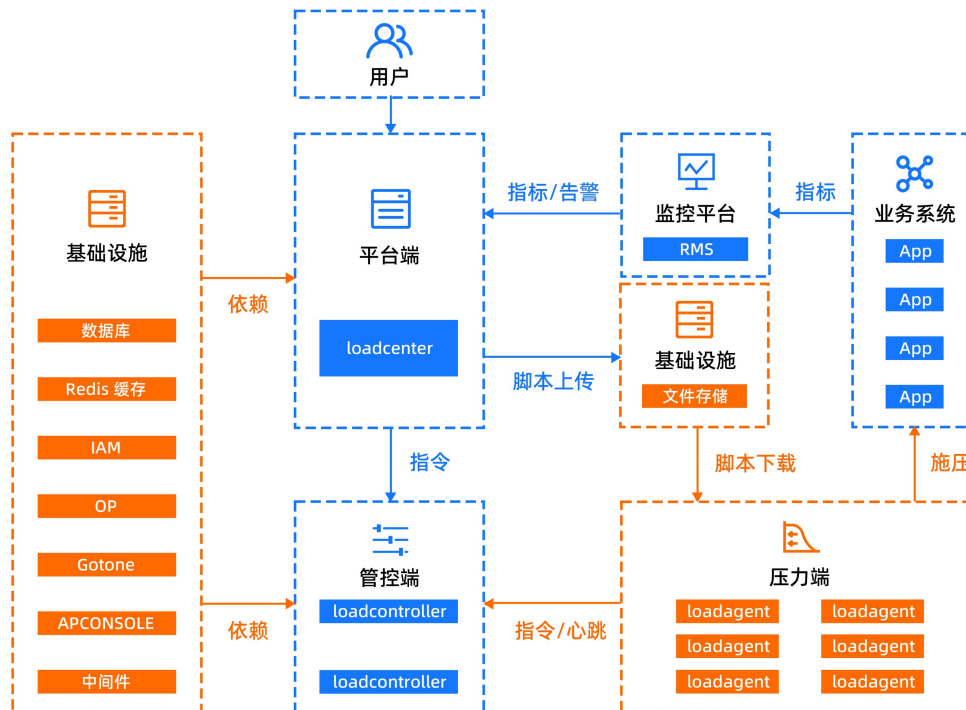
全链路压测平台有三个组件：

- 平台端（Loadcenter）：压测平台产品层，使用操作的入口，含项目管理、场景管理、压测报告、配置管理等模块。
- 管控端（Loadcontroller）：管控端管理并控制压力机，含压力机管理、压测任务管理、监控数据管理等模块。
- 压力端（Loadagent）：即压力机，负责向应用施压。

## 1.2. 系统架构

### 外部依赖

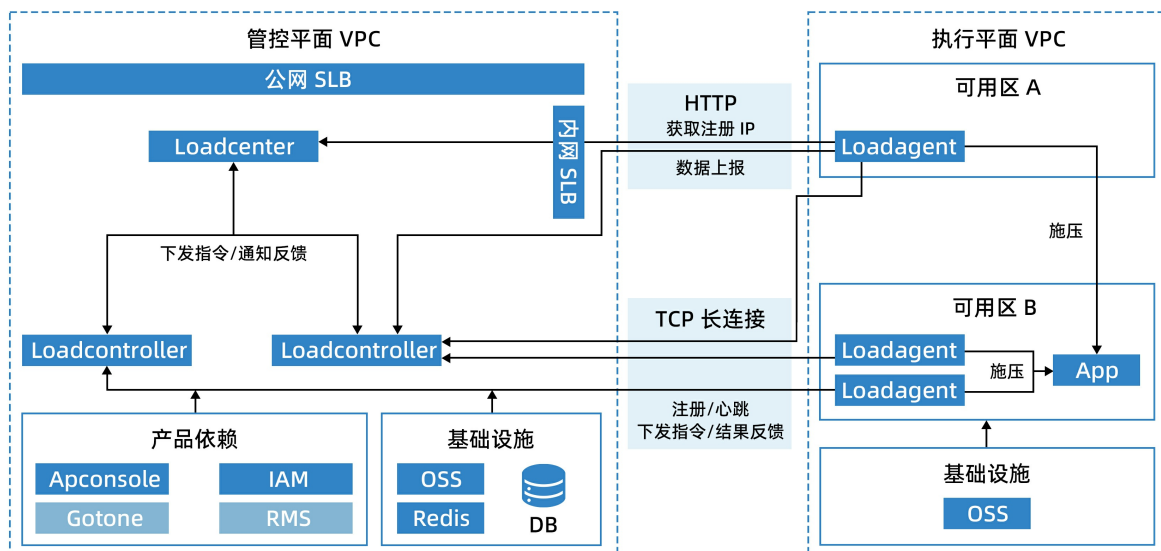
全链路压测平台整体架构如上图所示，涉及的组件可分为以下几部分：



- **基础设施**
  - **数据库**：平台端和管控端强依赖数据库。强依赖。
  - **Redis 缓存**：高可用 Redis 实例，用于缓存数据库热点数据，并在压测数据上报汇总时提供分布式缓存队列。强依赖。
  - **IAM**：身份鉴权，对接用户体系。强依赖。
  - **OP**：网关，对接租户、操作员、工作空间等基础信息。强依赖。
  - **Apconsole**：环境管理，您可以在平台端导航栏环境切换。强依赖。
  - **中间件**：内部平台端与管控端 RPC 通信。强依赖。
  - **Gotone**：压测报告邮件发送、压测执行、停止、熔断的钉钉通知。弱依赖。
  - **文件存储**：支持 OSS、AFS、MINIO 三种，用于压测脚本和压测结果数据的存储。强依赖。
- **监控平台 (RMS)**：应用、自定义指标实时监控，应用、自定义告警规则触发后进行压测熔断。若未对接 RMS，则产品监控方案和风控配置不可用。弱依赖。
- **业务系统**：被压测的业务应用，通常是一个集群。

## 内部通信机制

全链路压测平台按部署域可分为管控平面和执行平面。管控平面包括平台端和管控端，部署在中枢环境 VPC。执行平面包括压力端，部署在租户环境 VPC。



## 平台端和管控端

- 平台端探活管控端。
  - 管控端定期更新数据库记录作为心跳信息。
  - 平台端定期扫描数据库心跳时间以判断管控端是否存活。
- 平台端向管控端发送指令。平台端通过 TR (Taobao Remoting) 协议向管控端发送用户操作的指令，这些指令包括但不限于：
  - 压测任务部署。
  - 压测任务执行、停止。
  - 流量、配置、数据资产刷新。
  - 执行日志上报。
- 管控端通过 TR (Taobao Remoting) 协议向平台端压反馈测执行的结果。

## 管控端和压力端

- 压力端向管控端注册及发送心跳。

压力端启动后会通过 TCP 长连接注册到管控端，并定期向管控端发送心跳保活。
- 管控端向压力端发送指令。

管控端通过已建立的 TCP 长连接向压力端定向发送指令。
- 压力端向管控端上报数据。

压力端使用 HTTP 协议向管控端上报压测执行过程中产生的成功数、失败数、响应时间等数据。

## 压力端和平台端

压力端通过平台端内网 SLB 向平台端获取管控端地址。

## 2. 运维准备

产品运维过程中需要准备以下登录入口、账号、权限及工具。

| 名称   | 登录入口  | 权限    |
|------|---|-------|
| 云游   | <code>yunyou.example.com</code> ，根域名需要根据实际环境确认。     | 产品管理员 |
| 压测平台 | <code>loadcenter.example.com</code> ，根域名需要根据实际环境确认。 | 管理员   |



## 3. 应急运维

### 应急维护的定义和流程

#### 应急维护的定义

在出现重大故障时，最短时间内进行业务恢复的处理定义为应急处理操作。

所谓重大故障，是指发生突然、影响面广、涉及范围大、并可对网络的安全运行与服务质量造成严重后果的故障，如可用性故障。

#### 应急维护的流程和原则

如遇到下文覆盖的故障，建议按照文中指导进行故障排查和修复；若故障排查和修复遇到困难，可联系技术支持寻求帮助。如遇到下文未覆盖的故障，请直接联系技术支持。

### 应用宕机

#### acloadcenter 宕机

【现象】压测平台页面无法访问，报网络连接失败或者浏览器页面报 503 错误。

##### 【可能原因】

- Java 内存溢出导致应用宕机。
- 组件的 bug。
- 底层 PaaS 异常调度。

##### 【解决方法】

登录云游页面，找到 acloadcenter 应用的容器，检查容器状态是否健康。

- 若容器状态异常，可直接通过云游发起容器重启操作。
- 若容器可以登录，通过 `ps -ef | grep java | grep -v grep`，检查容器 Java 进程是否存在。
  - 若 Java 进程不存在，查看 `/home/admin/logs` 下是否有 `.Hprof` 结尾的文件，若有此文件，则保存此文件，然后通过云游发起容器重启操作。
  - 若 Java 进程存在，可以执行 `jmap -heap <pid>` 命令，查看各内存区域的设置；执行 `jmap-histo:live <pid>` 命令，查看每个 class 的实例数目、内存使用率和类全名；保存收集到的 JVM 相关数据、信息，联系技术支持人员排查，然后通过云游发起容器重启操作。

【验证方法】页面是否可以正常访问。

#### acloadcontrol 宕机

【现象】平台 配置/压力机管理 中所有压力机均显示离线状态。

##### 【可能原因】

- Java 内存溢出导致 acloadcontrol 应用宕机。
- 组件的 bug。
- 底层 PaaS 异常调度。

##### 【解决方法】

登录云游页面，找到 acloadcontrol 应用的容器，检查容器状态是否健康。

- 若容器状态异常，可直接通过云游发起容器重启操作。
- 若容器可以登录，通过 `ps -ef | grep java | grep -v grep`，检查容器 Java 进程是否存在。
  - 若 Java 进程不存在，查看 `/home/admin/logs` 下是否有 `.Hprof` 结尾的文件，若有此文件，则保存此文件，然后通过云游发起容器重启操作。
  - 若 Java 进程存在，可以执行 `jmap -heap <pid>` 命令，查看各内存区域的设置；执行 `jmap-histo:live <pid>` 命令，查看每个 class 的实例数目、内存使用率和类全名；保存收集到的 JVM 相关数据、信息，联系技术支持人员排查，然后通过云游发起容器重启操作。

【验证方法】压测平台 配置/压力机管理 中所有压力机显示在线状态。

## acloadagent 宕机

【现象】平台 配置/压力机管理 中某台压力机显示状态为离线且更新时间字段超过 10 分钟。

【可能原因】

- 压测脚本内存溢出导致 acloadagent 宕机。
- 组件的 bug。
- 底层 PaaS 异常调度

【解决方法】

登录云游页面，查看是否有对应 acloadagent 的容器。

- 若无容器则表明该压力机已下线，属于正常现象。
  - 若有容器则检查容器状态是否健康。
    - 若容器状态异常，可直接通过云游发起容器重启操作。
    - 若容器可以登录，通过执行 `ps -ef | grep java | grep -v grep | grep -v com.alipay.antcloud.loadagent.Starter` 命名检查容器内是否有脚本进程存在。
- 若脚本进程存在，在 配置/压力机管理 中找到该压力机正在执行的场景，进而找到对应脚本进行排查，然后通过云游发起容器重启操作。

【验证方法】压测平台 配置/压力机管理 中所有压力机显示在线状态。

## 断电恢复

压测平台所涉及的组件都是通过容器方式部署，断电重启之后，容器会自动恢复。如果服务没有恢复，可以到云游控制台上逐个重启压测平台相关应用，不涉及启动顺序。

## 网络连接失败

网络连接失败，可能导致 压测平台 相关服务不可用，需要对底层 PaaS 进行排查。

## 数据丢失

压测平台的数据存储包括以下三部分：

- 关系型数据库（RDS 或者 OceanBase）

- RDS、OceanBase 存储压测平台自身的业务数据，其可用性由 RDS\OceanBase 自身的高可用能力保证。
- 若出现数据丢失则需要压测平台进行重新配置。

- **Redis 缓存数据**

Redis 缓存可用性由 Redis 实例自身的高可用能力保证，存储了以下两部分数据：

- 数据库热点数据。
- 压测数据上报汇总的实时数据。

若在压测执行时 Redis 数据丢失，则会导致正在执行的压测结果异常，需要重新执行；若在非压测执行时 Redis 数据丢失，不会造成业务影响。

- **文件存储（OSS 或 AFS 或 MINIO）**

文件存储数据可用性依托于文件存储自身的数据备份与冗余机制，主要用于存储以下内容：

- 平台模板脚本和租户上传的自定义脚本，若出现数据丢失需要重新上传这些脚本。
- 压测执行产生的流量数据及系统监控数据，若出现数据丢失需要重新压测。
- 已发送的压测报告邮件中的流量、指标曲线图片，若出现数据丢失、邮件图片异常，需要重新生成。

## 4. 日常运维

Loadcenter 的日常运维主要包含监控和巡检两部分内容。

### 监控

登录 RMS 3.0 页面，查找对应的应用信息，基于监控项查看应用监控信息。

| 场景   | 监控应用          | 告警项        | 告警阈值   |
|------|---------------|------------|--------|
| 组件监控 | acloadcenter  | CPU 使用率    | 大于 90% |
|      |               | 内存使用率      | 大于 90% |
|      |               | 磁盘使用率      | 大于 90% |
|      |               | Load1      | 2      |
|      | acloadcontrol | CPU 使用率    | 大于 90% |
|      |               | 内存使用率      | 大于 90% |
|      |               | 磁盘使用率      | 大于 90% |
|      |               | 端口监控（5061） | 不通     |
|      | acloadagent   | CPU 使用率    | 大于 90% |
|      |               | 内存使用率      | 大于 90% |
|      |               | 磁盘使用率      | 大于 90% |
|      |               | Load1      | 2      |

### 巡检

您可以通过人工拨测 URL 或者端口来判断产品是否正常运转，同时也可以通过守夜人平台执行巡检脚本进行判断。巡检通过守夜人平台进行，不区分业务巡检和组件巡检。

| 应用            | 分类      | 说明                   |
|---------------|---------|----------------------|
| acloadcenter  | 基础巡检—端口 | 巡检 80 和 7001 端口是否正常。 |
| acloadcontrol | 基础巡检—端口 | 巡检 80 和 7001 端口是否正常。 |

## 5. 常见故障处理

### 压力机显示离线状态

**场景描述：**在 **配置/压力机管理** 中某台压力机显示 **离线** 状态，且该压力机的 **更新时间** 已较长时间未更新。

**产生原因：**

- 压力机已经下线（通过云游下线或缩容）。
- 底层环境异常（PaaS 平台无法调度、宿主机环境异常等）。
- 管控端未更新压力机心跳（程序 bug 导致未发送心跳或收到心跳后处理异常、网络问题导致未收到心跳）。

**排查步骤：**

1. 排查压力机是否已缩容或下线。

在云游上查看对应容器状态，如果压力机 IP 对应的容器是下线状态或找不到该容器，说明压力机已经下线。根据实际情况决定是否需要上线该容器。

2. 排查该容器状态是否正常。

查看云游上对应容器状态是否正常，是否是由于环境问题导致 Pod 没有启动。

3. 排查压力机心跳情况。

```
tail -200f /home/admin/logs/loadagent.log | grep "Heartbeat"
```

- i. 观察 begin Heartbeat 和 Heartbeat success 是否每 10 秒成对出现一次。

- 无 Heartbeat 日志：说明压力机未发送心跳包，需联系技术支持协助排查压力机。
- 有 Heartbeat 日志，说明压力机没问题，接着通过以下 SQL 排查对应管控端状态：

```
select gmt_modified, status
from load_controller_server
where identifier =(
select register_identifier from load_agents where ip = "对应压力机IP地址")
```

- ii. 根据查出来的管控端 status 状态分两种情况：

- status 为 disabled：管控端异常，可重启管控端暂时解决该问题。如经常出现该问题，需联系技术支持进一步排查。
- status 为 enabled：管控端正常，程序处理心跳包异常，需重启该压力机或联系技术支持协助排查。

### 场景配置中无法添加应用监控

**场景描述：**压测场景/监控方案 的应用监控项下单击 **添加** 按钮后未出现应用筛选框。

**产生原因：**压测平台无法从 RMS 处获取应用列表。

**排查步骤：**

1. 排查该环境是否已部署 RMS，若未部署则需要先部署 RMS 才能使用。
2. 若已部署则可联系技术支持协助排查。

## 实时监控/流量监控中无数据曲线

**场景描述：**压测场景能正常执行，但是执行后 实时监控/流量监控 中无数据。

**产生原因：**

- 压力机上报数据异常。
- Redis 连接异常导致无法处理上报数据。
- 文件存储连接异常导致无法生成最终数据。

**排查步骤：**

1. 排查压力机上报数据是否异常。

```
grep -rnI "回调 loadcontroller 异常" /home/admin/logs/loadagent.log
```

若grep出异常信息，则需要排查网络问题。

2. 检查 Redis 配置，telnet redis 端口是否联通，进一步也可以查看 acloadcontrol 应用的日志是否有 Redis 报错。

```
grep -rnI "Redis" /home/admin/logs/loadcontrollercloud/common-error.log
```

3. 检查文件存储是否异常。

```
grep -rnI "OSS" /home/admin/logs/loadcontrollercloud/common-error.log
```

## 压测部署所属环境选不到内容

**场景描述：**在 压测场景/压测部署 中选择 所属环境 时无法选择所属环境。

**产生原因：**

- 环境选择不正确，需要在导航栏中选择正确的环境。
- 压力机配置的环境与导航栏选择的环境不一致。
- 该环境已无可用压力机，或压力机全部离线。

**排查步骤：**

1. 确认选择的环境是正确的。
2. 检查压力机配置的 `workspace_name` 是否和当前选择的环境一致。
3. 检查当前可用压力机数，若全部离线请参考 [压力机显示离线状态](#) 处理。

## 6. 参考日志

在运维过程中，您可以通过日志了解 Loadcenter 各组件的状态或定位故障。

Loadcenter 包括如下组件日志：

- Acloadcenter
- Acloadcontrol
- Acloadagent

Loadcenter 会根据既定策略将这些日志进行归档和清理。

### Acloadcenter 日志说明

| 文件名                   | 日志文件路径                           | 说明       | 归档策略   |
|-----------------------|----------------------------------|----------|--|
| app-default.log       | /home/admin/logs/loadcentercloud | 业务日志     | <ul style="list-style-type: none"><li>• 每日一个新文件。</li><li>• 旧文件的名称中会加日期后缀：<br/>名称.log.{yyyy-MM-dd}</li><li>• 最多保留 7 天的日志文件。</li></ul> |
| common-error.log      |                                  | 业务异常日志   |  |
| sofa-boot.log         | /home/admin/logs/sofa            | 应用启动日志   |  |
| rpc-client-digest.log | /home/admin/logs/tracelog        | RPC 调用日志 |  |
| rpc-server-digest.log |                                  | RPC 响应日志 |  |

### Acloadcontrol 日志说明

| 文件名                   | 日志文件路径                               | 说明       | 归档策略   |
|-----------------------|--------------------------------------|----------|--|
| app-default.log       | /home/admin/logs/loadcontrollercloud | 业务日志     | <ul style="list-style-type: none"><li>• 每日一个新文件。</li><li>• 旧文件的名称中会加日期后缀：<br/>名称.log.{yyyy-MM-dd}</li><li>• 最多保留 7 天的日志文件。</li></ul> |
| common-error.log      |                                      | 业务异常日志   |  |
| sofa-boot.log         | /home/admin/logs/sofa                | 应用启动日志   |  |
| rpc-client-digest.log | /home/admin/logs/tracelog            | RPC 调用日志 |  |
| rpc-server-digest.log |                                      | RPC 响应日志 |  |



## Acloadagent 日志说明

| 文件名                    | 日志文件路径                      | 说明         | 归档策略   |
|------------------------|-----------------------------|------------|--|
| loadagent.log          | /home/admin/logs            | agent 运维日志 | <ul style="list-style-type: none"><li>文件超过10 MB 后会加后缀索引生成旧文件：<br/>如 <code>.log.1</code>，最多保留 1 个历史文件。</li><li>每次压测执行都会生成一份对应本次压测的日志<br/><code>.log.<br/>{recordId}</code><br/>，如<br/><code>.log.1616056786<br/>321</code><br/>。</li><li>agent 会自动检测磁盘清理历史压测日志文件。</li></ul> |
| loadengine-default.log | /home/admin/logs/loadengine | 压测脚本日志     | <ul style="list-style-type: none"><li>每次压测执行生成一个日志文件格式为：<br/><code>.log.<br/>{recordId}</code><br/>。</li><li>agent 会自动检测磁盘清理历史日志文件。</li></ul>  |
| loadengine-digest.log  |                             | 压测脚本接口调用日志 | <ul style="list-style-type: none"><li>每日一个新文件。</li><li>旧文件的名称中会加日期后缀：<br/>名称 <code>.log.{yyyy-MM-dd}</code><br/>。</li><li>最多保留 7 天的日志文件，agent 会自动检测磁盘清理历史压测日志文件。</li></ul>   |
|                        |                             |            |  |

| 文件名                   | 日志文件路径 | 说明         | 归档策略  |
|-----------------------|--------|------------|---|
| loadengine.stderr.log |        | 压测脚本标准错误日志 | <ul style="list-style-type: none"> <li>每次压测执行生成一个日志文件格式为：                             <pre>.log. {recordId}</pre> </li> <li>agent 会自动检测磁盘清理历史日志文件。</li> </ul> |
| loadengine.stdout.log |        | 压测脚本标准输出日志 |   |

## 7. 基础术语

### 测试数据 (Test data)

保存参数化内容的文件为 `.csv` 格式，其他扩展文件一般为脚本自行读取，默认存放在脚本根目录下。如果选择拆分测试数据文件，则除表头行以外的文件拆分行数不得小于压测时分配的压力机数，否则会导致压测运行失败。

### 拆分文件 (Split files)

当使用多台压力机进行压测时，为避免数据产生并发冲突，按压力机数进行数据文件切割的一种方式。

### 断言 (Assert)

编程术语，用布尔值表示，主要用于代码开发和测试阶段，用于对某些关键数据的判断，如果这些关键数据不是程序所预期的数据，程序就提出警告或退出。

### 脚本配置 (Script configuration)

整个场景中，脚本可读取的测试数据。脚本配置的用法与测试数据类似，定义为形式。脚本配置与测试数据的区别在于，脚本配置修改后无需重启场景即可立即生效，并且脚本配置读取到内存中，对于单一配置，其性能比使用 `.csv` 文件更高。

### 每秒事务数 (Transaction Per Second, 简称 TPS)

压测场景中的 TPS，取决于用户定义的度量维度，压测平台支持业务单元和业务流两种维度。

### 全链路压测 (Loadcenter)

全链路压测基于真实的生产业务场景、真实或高仿真的系统环境，模拟海量用户高并发的请求，对整个业务链路进行压力测试，发现整个链路中的性能问题和瓶颈点，也可以理解为端到端压力测试。

### 循环读取 (Read circularly)

针对测试数据文件，采用从头到尾的顺序依次循环读取。

### 压测报告 (Loadtest report)

压测报告基于压测记录生成。通常有效压测记录会生成压测报告，压测报告支持对目标、结论以及通过状态进行补充和修改。

### 压测场景 (Loadtest scenario)

压测场景定义一个压测模型集合，用来统一调度压测资源，执行压测，生成压测结果和报告。

### 压测记录 (Loadtest record)

压测记录记录一次压测中产生的数据，如响应时间、TPS、成功/失败数以及应用监控等，每次压测对应一条压测记录。

### 压测项目 (Loadtest project)

压测项目定义一个场景集合的维度，同一项目下的场景间可以共享压测脚本，项目成员拥有该项目下所有场景的查看和操作权限。

### 压力机 (Load Agent)

压力机又称负载机，一般为集群部署，可模拟用户并发场景下产生的负载，对目标服务器进行压力测试。在全链路压测产品中，压力机特指装有 LoadAgent 应用的服务器。

### 业务单元 (Test unit)

业务单元是压测的最小粒度，TPS、响应时间等压测时最小的度量单位，压测平台会统计包含在业务单元中的所有耗时、成功/失败数。

#### 业务监控 (Businessmonitor)

业务监控是具体的业务运行数据，例如 xx 交易、xx 支付交易量、交易耗时。

#### 业务流 (Testworkflow)

业务流由业务单元组装而成。业务流中的业务单元串行执行，并且同一业务流中的业务单元间可通过脚本引擎上下文进行数据传递，即业务流可以看作一个线程组，业务流之间并行执行。

业务流也是度量维度，响应时间为业务流内所有业务单元响应时间之和。若业务流的其中一个业务单元出现异常，系统将不会继续执行该业务流，而是继续下一个循环，即进行该业务流的下一次调用。

#### 影子全链路压测 (ShadowFull chain loadtest)

影子全链路压测基于 Tracer 工具传递全链路压测标识，后续各个中间件或业务基于压测标来进行不同的分支处理流程，这部分业务流量也称为影子流量（系统基于压测标来区分影子流量和真实流量）。所有的表需要建立对应的影子表，在影子表落地影子流量产生的脏数据，例如影子账户在压测业务进行中产生的数据。

#### 应用监控 (Applicationmonitor)

应用监控指系统硬件指标监控（如 CPU、负载、硬盘、内存等）及应用业务指标监控（如错误量（Error）、页面访问量（PV）、被调用的应用服务量（Service）等）。