

Table of Contents

1. [基调网络听云Ruby探针说明](#) 1.1
2. [准备阶段](#) 1.2
 1. [兼容性](#) 1.2.1
 1. [操作系统](#) 1.2.1.1
 2. [Ruby 版本](#) 1.2.1.2
 3. [Web servers](#) 1.2.1.3
 4. [Web frameworks](#) 1.2.1.4
 5. [Databases](#) 1.2.1.5
 6. [Background jobs](#) 1.2.1.6
 7. [HTTP clients](#) 1.2.1.7
 8. [Other](#) 1.2.1.8
 2. [探针架构](#) 1.2.2
 3. [常见问题](#) 1.2.3
 4. [Release Notes](#) 1.2.4
3. [安装和配置](#) 1.3
 1. [安装探针须知](#) 1.3.1
 2. [在线安装探针](#) 1.3.2
 3. [本地安装探针](#) 1.3.3
 4. [更新探针](#) 1.3.4
 5. [卸载探针](#) 1.3.5
 6. [安装配置文件](#) 1.3.6
 7. [配置项](#) 1.3.7
 1. [授权序列号](#) 1.3.7.1
 2. [应用名](#) 1.3.7.2
 3. [日志管理](#) 1.3.7.3
 4. [审计模式](#) 1.3.7.4
 5. [DC连接选项](#) 1.3.7.5
 6. [Rake](#) 1.3.7.6
 7. [采集数据项](#) 1.3.7.7
4. [Ruby agent API](#) 1.4
 1. [record_metric](#) 1.4.1
 2. [notice_error](#) 1.4.2
 3. [manual_start](#) 1.4.3
 4. [shutdown](#) 1.4.4
5. [故障排查](#) 1.5
 1. [查看日志](#) 1.5.1

基调网络听云Ruby探针说明



准备阶段

Ruby探针运行环境

兼容性

操作系统

Ruby Agent 支持 UNIX-like 操作系统。eg: Linux, Solaris, and Mac OS X.

Ruby 版本

名字	支持	不支持	实验阶段
JRuby	-	-	-
MRI	1.9.2, 1.9.3, 2.0.0, ~> 2.1.0, ~> 2.2.0	-	-
Rubinius	-	-	-

Web servers

名字	支持	不支持	试验阶段
passenger	~>2.2, ~>3.0, ~>4.0	-	-
puma	~>2.0	~>1.0	-
rainbows!	-	-	4.5.0
thin	~>1.0	-	-
unicorn	~>4.0	~>1.0, ~>2.0, ~> 3.0	-

Web frameworks

名字	支持	不支持	实验阶段
grape	-	-	-
padrino	-	-	-
rack	>= 1.1.0	~>1.0.0	-
rails	~3.0.0, ~>3.1.0, ~>3.2.0, ~>4.0.0, ~>4.1.0, ~>4.2.0	~>2.0.0	-
sinatra	-	-	-

Databases

名字	支持	不支持	实验阶段
activerecord	~>3.0.0, ~>3.1.0, ~>3.2.0, ~>4.0.0, ~>4.1.0, ~>4.2.0	~>2.0.0	-
datamapper	-	-	-
mongo	~>1.8, ~>2.1	-	-
redis	~> 3.0	-	-
sequel	-	-	-

Background jobs

名字	支持	不支持	实验阶段
delayed_job	-	-	-
rake	~> 10.0	-	-
resque	-	-	-
sidekiq	-	-	-

HTTP clients

名字	支持	不支持	实验阶段
Net::HTTP	Supported for all agent-supported versions of Ruby	-	-
curb	-	-	-
excon	-	-	-
httpclient	-	-	-
typhoeus	-	-	-

Other

名字	支持	不支持	实验阶段
activemerchant	-	-	-
acts_as_solr	-	-	-
authlogic	-	-	-
dalli	-	-	-
memcache-client	-	-	-
sunspot	-	-	-

探针架构

听云Ruby探针是以gem的方式运行。听云Ruby探针采集Ruby应用的运行时性能数据和一些trace信息，经过合并整理，发送到听云服务器

常见问题

1. 听云Ruby探针都能采集哪些数据?

目前，听云Ruby探针可以采集Rails3+ 应用运行时的性能数据和错误信息。

性能数据包括：HTTP请求的总响应时间、前端阻塞时间、数据库耗时、NOSQL耗时、外部服务耗时等性能；详细到数据库表及操作的性能统计；超过阈值的性能瓶颈函数；慢sql语句及调用堆栈；

错误信息包括：数据库连接错误、sql语法错误、Ruby运行时异常等；

2. 我在多台服务器上部署了应用探针，报表查看时，如何将不同的应用区分开？

把配置文件内的 `app_name`（应用名称） 改为不同名称就可以了。

Release Notes

探针版本 1.0.0

更新日期：2016-04-28

新增

Rails 3+ Ruby 1.9+ mysql 5+

mongo 1.0+

redis 1.0+

支持Rails中间件的跟踪

支持性能数据的分析

支持慢过程应用追踪

支持慢SQL的追踪

支持多进程启动探针的数据合并

探针版本 1.1.0

更新日期：2016-07-05

新增

基于HTTPVnet 的跨应用追踪

跨应用追踪的错误细分

错误率相关功能的代码优化重构

支持Thrift的Ruby Gem客户端版本 9.3

探针版本 1.1.2

更新日期：2016-08-04

修改bug

非正常thrift过程，会导致性能数据在某些情况下会获取不到

探针与DC数据交互的优化

探针版本 1.1.3

更新日期：**2016-08-23**

新增

支持rake的监测

修改**bug**

修复结点超过`transaction_tracer.limit_segments`数量的bug

修复缓存中间件监测过程中会照成数据丢失的bug

安装和配置

安装探针须知

1. 如果您还没有听云server帐户，请[注册听云server帐户](#)
2. 从听云server探针安装页面复制您的授权序号
3. 安装Ruby探针
4. 配置Ruby探针的授权序号
5. 重启Ruby应用
6. 通过管理页面察看数据，有问题请参考安装和排除故障章节

在线安装探针

Install the gem

The TingYun Ruby agent's gem is available from rubygems.org as `tingyun_rpm`. For applications using Bundler, add this gem to the Gemfile:

```
gem 'tingyun_rpm'
```

定位至「应用程序根目录」，运行以下安装指令

```
bundle install
```

若出现以下提示，则代表安装成功

```
Your bundle is complete!
```

```
Use 'bundle show [gemname]' to see where a bundled gem is installed.
```

本地安装探针

Install the gem

如果你的环境无法正常访问Internet. Awsome!
你需要使用另外一台正常访问听云官网，下载最新的Ruby agent 包。
解压并获取其中 `tingyun_rpm.gem` 文件。

在项目的Gemfile文件中，添加：

```
gem 'tingyun_rpm'
```

定位至「应用程序根目录」，运行以下安装指令

```
gem install 【path】/tingyun_rpm.gem
```

若出现以下提示，则代表安装成功

```
Parsing documentation for tingyun_rpm  
Installing ri documentation for tingyun_rpm  
Done installing documentation for tingyun_rpm after 6 seconds  
1 gem installed
```

更新探针

If you're using Bundler, run `bundle update tingyun_rpm`

Otherwise, run `gem update tingyun_rpm`

卸载探针

如果采用**Bundler**安装的，删除 Gemfile 中的 `gem 'tingyun_rpm'`

如果不是采用**Bundler**安装的，删除 environment.rb 中所有与 `tingyun_rpm` 有关的内容

安装配置文件

Ruby agent 的 `tingyun.yml` 是标准的 **yaml** 配置文件. 包含了默认的一些配置项, 并且增加了不同环境的区域设置。 **for example, Development, Testing, and Production.**

可以从听云官网的**最新Ruby agent**包中, 获取**最新**的`tingyun.yml`. 将`tingyun.yml`复制到 **Application** 的 **config** 目录下。

配置项

授权序列号

licenseKey

Type String

Default ""

NOTE 本条配置项是您在安装时输入的授权序号。

应用名

app_name

Type

String

Default (Dynamic)

NOTE 应用识别名称,此名称将体现在报表页面

审计模式

nbs.audit_mode

Type

Boolean

Default false

NOTE 本选项设定是否在日志文件中写入更详尽的信息，包括所有的向听云后台上传和下载的数据。

DC连接选项

host

Type **String**

Default redirect.networkbench.com

NOTE 本选项指定报表服务器的地址

port

Type **Fixnum**

Default 80/443

NOTE 本选项指定报表服务器的端口。开启ssl是443，否则是80

ssl

Type **Boolean**

Default true

NOTE 本选项指定向服务器发送数据是否弃用安全连接(https)。若设定为true，则向服务器发送数据时采用https方式。否则，使用普通http方式

proxy_host

Type **String**

Default 空

NOTE 本选项指定代理服务器的地址。若选项不为空，并且未启用安全连接，则本选项值为http代理服务器的ip地址

proxy_port

Type **Fixnum**

Default 8080

NOTE 本选项指定代理服务器的端口。

proxy_user

Type **String**

Default 空

NOTE 若代理服务器需要用户名密码，本选项指定代理服务器的登陆名

password

Type **String**

Default 空

NOTE

NOTE

keep_retrying

Type **Boolean**

Default true

NOTE 容许探针与报表服务器连接失败以后，可以继续尝试连接

keep_alive_timeout

Type **Fixnum**

Default 60 sec

NOTE agent与报表服务器的TCP连接有效期

Rake

disable_rake

Type **Boolean**

Default true

NOTE 默认不采集Rake相关数据

rake.tasks

Type **Array**

Default []

NOTE 如果 disable_rake为false，则采集列表中指定的Rake任务

采集数据项

data_report_period

Type **Fixnum**

Default 60 sec

NOTE 采集数据周期

post_size_limit

Type **Fixnum**

Default 2 * 1024 * 1024

NOTE 单次采集数据大小限制,默认2M

disable_middleware_instrumentation

Type **Boolean**

Default false

NOTE 采集数据是否忽略中间件的信息

disable_view_instrumentation

Type **Boolean**

Default false

NOTE 采集数据是否忽略页面的信息

transaction_tracer.limit_segments

Type **Fixnum**

Default 2000

NOTE 一次慢过程追踪的数据汇总数，如果太多，不仅影响性能，也没有太大意义

Ruby agent API

Ruby agent 不仅容许你个性化配置，还可以使用探针提供的**API**，监测指定的内容
收集性能数据的 **public API** 定义在 **moudle TingYun::Agent** 中

record_metric

采集指定的数据变化

```
TingYun::Agent.record_metric(metric_name, value)
```

This method is safe to use from any thread.

metric_name 以 "Custom/" 开头，指定性能片段的名称
value Numeric or a Hash containing :count :total, :min, :max, and :sum_of_squares keys

```
class Article
  def checkout
    amount = compute_articles_total
    ::TingYun::Agent.record_metric('Custom/article/charge_amount', amount)
  end
end
```

notice_error

采集指定的异常

```
TingYun::Agent.notice_error(exception, options = { })
```

记录一个异常，或者异常信息。options 是可选的

options = { }	Comments
:uri	The request path, minus any request parameters or query string
:referer	The URI of the referring resource
:metric	The metric name associated with the transaction
:request_params	Request parameters, already filtered if necessary
:custom_params	Custom parameters

```
class Article
  def checkout
    begin
      amount = compute_articles_total
    rescue => e
      ::TingYun::Agent.notice_error(e)
    end
  end
end
```

manual_start

手动启动agent

通常情况，agent自动随application启动。但是，在某些情况下，可能需要手动启动探针，比如执行Rake任务。

```
::TingYun::Agent.manual_start(options={})
```

options 是一个可选Hash参数,会在tingyun.yml的配置生效前，预先生效。

下面是在Rake任务中，手动启动agent的例子:

```
::TingYun::Agent.manual_start
```

```
task :manual do
  p "manula start agent"
end
```

shutdown

关闭agent

```
::TingYun::Agent.shutdown
```

该方法会在程序exit前调用。主要提供agent内部方法使用，暂不需要用户直接调用。

故障排查

查看日志

如果系统工作不正常或者没有数据，通常可以从日志里获取更多的信息来定位问题。

```
grep -n -E 'FATAL|ERROR' tingyun_agent.log
```

如果有ERROR信息，请阅读日志文件对应行数，通过error描述定位问题。

关于日志的输出控制，请参考配置章节。

默认情况，日志级别为info, 审计模式是关闭的，这种情况得到的信息量比较小。通过修改日志级别为debug，开启审计模式，让听云Ruby输出更多日志信息，通过日志得到的信息定位问题。