ALIBABA CLOUD

阿里云

函数计算

函数调用

文档版本：20220620

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，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。

3. 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。阿里云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，并在阿里云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阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

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 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 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 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 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 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.同步调用 05
2.异步调用 06
2.1. 功能概览 06
2.2. 重试策略 07
2.3. 结果回调 08
3.异步任务 16
3.1. 功能概览 16
3.2. 任务管理 17
3.3. 事件触发 20
3.4. 任务去重 20
3.5. 任务监控 21
3.6. 任务编排 22
4.调用示例 24
5.错误处理 26
6.弹性管理（含预留模式） 28
7.函数级按量实例伸缩控制 38
8.函数调用FAQ 40
8.1. 我的客户端不关心函数执行结果，我不希望我的客户端一直等函数返... 40

1.同步调用

同步调用是调用函数的一种方式，当您同步调用一个函数时，事件将直接触发函数，函数计算会运行该函数并等待响应。当函数调用完成后，函数计算会将执行结果直接返回给您，例如返回结果、执行摘要和日志输出。本文介绍同步调用的使用场景和使用限制等。

使用场景

同步调用是事件被函数处理后直接返回结果。同步调用的场景非常广泛，包括但不限于以下使用场景：

- 需及时查看执行结果。
- 设置了HTTP触发器的函数。

使用限制

资源调用限制：您的阿里云账号（主账号）在单个地域内默认的按量实例上限数为300。

> **说明** 您可以通过函数在云监控控制台中的相关指标（throttles）来观察流控行为。如果您需要提高该限制，请提交工单。

并发执行

并发执行是指在任意指定时间您的函数代码同时执行的数量。您可以用以下公式来估算并发的函数调用数：

并发调用数=请求速率×函数执行时间

- 请求速率：函数被调用的速率，即每秒请求数或每秒事件数。
- 函数执行时间：函数请求到达实例开始，到请求执行完毕的时长。单位为秒。

例如，一个处理阿里云OSS事件的函数的平均执行时间为3秒，OSS每秒发布10个事件，那么根据该公式计算可得，您的函数有30个并发执行。

> **说明** 函数并发执行数会影响您的计费。关于计费的详细信息，请参见计费概述。

2.异步调用

2.1. 功能概览

本文介绍异步调用的背景信息、应用场景以及如何实现延迟调用函数等。

背景信息

函数计算系统接收异步调用请求后，将请求持久化后会立即返回响应，而不是等待请求执行完成后再返回。函数计算保证请求至少执行一次。如果您希望获得异步调用的结果，可以通过配置异步调用目标来实现，具体信息，请参见结果回调。如果您希望获得函数异步请求各个阶段的状态，可通过开启任务模式来实现，具体信息，请参见功能概览。

应用场景

如果您的函数中存在耗时较长、资源消耗较大或容易出错的逻辑，您可以使用异步调用的方式，让您的程序响应更加迅速，更加可靠地面对突发流量。例如：

- 新用户注册系统中，新用户发送注册请求，注册成功后系统向用户发送注册成功邮件通知，发送邮件的动作可以从注册请求处理流程中剥离，异步执行。
- 上传文件时，转换格式和导入导出等动作可以从上传数据流程中剥离，异步执行。

> **说明** HTTP函数支持同步调用和异步调用（公测中）。

延迟调用

针对某些场景，您提交一次异步调用后，需要函数计算对其进行延迟触发。您可以通过调用API（SDK）实现延迟调用函数。

在代码中添加HTTP请求头 `x-fc-async-delay` ，其取值范围为(0,3600)，单位为秒。函数计算将从您触发执行开始计算，延迟 `x-fc-async-delay` 设置的时间后触发函数调用。

以Go SDK为例，代码如下所示：

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/fc-go-sdk"
)
func main() {
    fcClient, err := fc.NewClient(fmt.Sprintf("%s.cn-shanghai.fc.aliyuncs.com", os.Getenv("
ACCOUNT_ID")),
        "2016-08-15", os.Getenv("ACCESS_KEY_ID"), os.Getenv("ACCESS_KEY_ID_SECRET"))
    if err != nil {
        panic(err)
    }
    // invoke function with delay
    invokeInput := fc.NewInvokeFunctionInput({ServiceName}, {FunctionName}).WithPayload({pa
yload})
    invokeInput = invokeInput.WithAsyncInvocation().WithHeader("x-fc-async-delay", "200")
    _, err := FCClient.InvokeFunction(invokeInput)
    if err != nil {
        panic(err)
    }
}
```

注意 通过上述操作实现的延迟调用在某些场景下存在一定误差，如您需要更加精准的延迟调用函数，请使用定时触发器。具体信息，请参见配置定时触发器。

常见功能

异步调用的常见功能如下所示：

- 事件触发
- 重试策略
- 结果回调

2.2. 重试策略

当函数异步调用执行失败后，函数计算会自动进行错误重试。本文介绍重试机制以及如何在函数计算控制台配置重试策略。

重试机制

对于常见错误，系统默认的重试策略如下表所示：

状态码	执行失败原因	服务器端行为
200	错误类型为 `HandledInvocationError` 或 `UnhandledInvocationError` 。更多信息，请参见基础信息。	默认重试3次，或根据异步调用配置中设置的maxAsyncRetryAttempts重试。

状态码	执行失败原因	服务器端行为
429	函数并发执行超过上限被流控。	以二进制指数退避方式重试执行5小时。当您的函数执行失败后将在0.5秒后开始重试，后续重试执行的时间间隔将以二进制指数退避方式计算，即重试时间间隔为1秒、2秒、4秒、8秒等持续重试5小时。
500	系统错误。	
503	函数计算资源不足。	

配置重试策略

函数计算支持自定义重试次数和消息最大存活时长。

1. 登录函数计算控制台。
2. 在左侧导航栏，单击**服务及函数**。
3. 在顶部菜单栏，选择地域。
4. 在**服务列表**页面，单击目标服务。
5. 在**函数管理**页面，单击目标函数名称。
6. 在目标函数详情页面，单击**异步配置**页签，然后在**异步策略**区域，单击**编辑**。
7. 在**编辑异步策略**面板，按需修改**最大重试次数**和**消息最大存活时长**，然后单击**确定**。

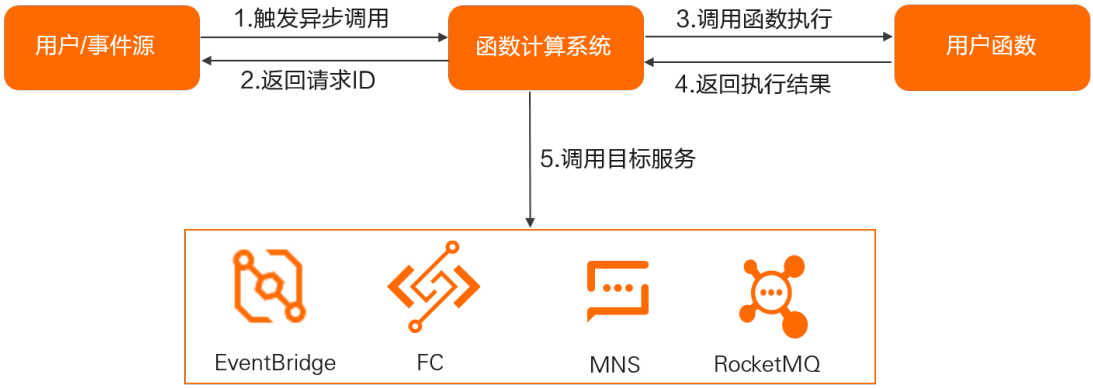
参数名称	解释说明
最大重试次数	用于配置异步调用流程中的消息最大重试次数，取值范围[0,8]。 函数计算在默认情况下，对异步触发失败的消息进行3次重试，您可以根据业务需求减少或增加对异步调用的重试。
消息最大存活时长	用于配置异步调用流程中的消息最大存活时长，取值范围[1,2592000]，单位为秒。 该时长从触发异步调用时开始计算，到该消息出队准备进行处理为止。如果超过配置的消息最大存活时长，该条消息将被丢弃。未被消费的消息将计入云监控异步调用触发事件（次）指标。指标详情，请参见监控指标。

2.3. 结果回调

函数计算异步调用支持结果自动回调功能。当任务执行完成后，函数计算将根据执行结果自动回调对应的服务。此功能需要您配置异步目标服务。本文介绍结果回调功能原理、适用场景和支持的异步目标服务，以及如何在函数计算控制台配置异步目标服务。

功能原理

结果回调流程如下图所示：



适用场景

- 保存丢弃的事件供后续使用

  当异步请求执行失败，并且按照指定的策略重试后仍然失败，函数计算将丢弃该请求。如果您配置了失败目标，函数计算将自动把失败请求的上下文信息推送到

  消息队列RocketMQ版

  等消息服务中，以便后续处理。您也可以将目标服务设置为另一个函数，函数计算将自动把失败请求的上下文信息推送到该函数，执行您自定义的错误处理逻辑。

- 自动通知下游服务执行结果

  请求执行成功后，如果您配置了成功目标，函数计算系统会自动将成功请求的上下文信息推送到下游目标服务。

支持的异步调用目标服务

当您为函数配置了异步调用目标，并且异步调用后的结果符合条件时，函数计算会将请求上下文和数据推送至对应服务。您可以针对不同函数、别名和版本配置不同的目标服务。目前支持的异步调用目标服务如下：

-
- 函数计算
- 事件总线EventBridge

  > **说明** 支持将事件总线EventBridge配置为目标服务的地域包括华东1（杭州）、华东2（上海）、华北2（北京）、中国香港、美国（硅谷）和美国（弗吉尼亚）。

- 消息队列RocketMQ版

  > **说明** 支持将
  >
  > 消息队列RocketMQ版
  >
  > 配置为目标服务的地域包括华东1（杭州）和华北1（青岛）。

异步调用目标服务的配置说明如下：

- 异步调用目标的事件内容

```
{
    "timestamp": "2020-08-20T12:00:00.000Z",
    "requestContext": {
        "requestId": "xxx",
        "functionArn": "acs:fc:::services/{serviceName}/functions/{functionName}",
        "condition": "FunctionResourceExhausted",
        "approximateInvokeCount": 3
    },
    "requestPayload": "",
    "responseContext": {
        "statusCode": 200,
        "functionError": ""
    },
    "responsePayload": ""
}
```

参数说明

参数	说明
timestamp	调用时间戳。
requestContext	请求上下文。
requestContext.requestId	异步调用的请求ID。
requestContext.functionArn	异步执行的函数ARN。
requestContext.condition	调用错误码。
requestContext.approximateInvokeCount	异步调用的执行次数。当该值大于1时，说明函数计算对您的函数进行了重试。
requestPayload	请求函数的原始负载。
responseContext	返回上下文。
responseContext.statusCode	调用函数的返回码（系统）。当该返回码不为200时，说明出现了系统错误。
responseContext.functionError	调用错误信息。
responsePayload	执行函数返回的原始负载。

注意 事件总线EventBridge作为函数异步调用目标时，事件内容会多一些参数。具体信息，请参见事件概述。具体示例如下所示。


```
{
    "datacontenttype": "application/json",
    "aliyunaccountid": "143xxxx",
    "data": {
        "requestContext": {
            "condition": "",
            "approximateInvokeCount": 1,
            "requestId": "0fcb7f0c-xxxx",
            "functionArn": "acs:fc:::services/xxxx.LATEST/functions/xxxx"
        },
        "requestPayload": "",
        "responsePayload": "",
        "responseContext": {
            "functionError": "",
            "statusCode": 200
        },
        "timestamp": 12345
    },
    "subject": "acs:fc:::services/xxxx.LATEST/functions/xxxx",
    "source": "acs:fc",
    "type": "fc:AsyncInvoke:succeeded",
    "aliyunpublishtime": "2021-01-03T09:44:31.233Asia/Shanghai",
    "specversion": "1.0",
    "aliyuneventbusname": "xxxxxxx",
    "id": "ecc4865xxxxxx",
    "time": "2021-01-03T01:44:31Z",
    "aliyunregionid": "cn-shanghai-vpc",
    "aliyunpublishaddr": "199.99.xxx.xxx"
}
```


- 负载限制

  支持的异步调用目标服务负载的最大限制如下：

  - ：64 KB
  - 函数计算：128 KB
  - 事件总线EventBridge：64 KB
  - 消息队列RocketMQ版

    ：4 MB

- 避免循环调用

  当您在配置异步执行目标时，请务必保证不要出现循环调用的情况。例如，您为函数A配置了成功调用时的异步目标为函数B，为函数B配置了成功调用时的异步目标为函数A。当您异步触发函数A并且执行成功后，则可能出现A到B，再到A的循环调用的情况。

配置异步调用目标服务

说明 配置异步调用目标前，请先创建函数并确保服务中函数所使用的角色具有对应的云服务权限。具体信息，请参见授予函数计算访问其他云服务的权限。

- ：mns:SendMessage或mns:PublishMessage。
- 函数计算：fc:InvokeFunction。
- 事件总线EventBridge：eventbridge:PutEvents。
- 消息队列RocketMQ版

  ：mq:PUB。

1. 登录函数计算控制台。
2. 在左侧导航栏，单击**服务及函数**。
3. 在顶部菜单栏，选择地域。
4. 在**服务列表**页面，单击目标服务。
5. 在**函数管理**页面，单击目标函数名称。
6. 在目标函数详情页面，单击**异步配置**页签。
7. 在**异步配置**页签，按需配置参数信息。
   - 配置成功目标
     a. 在**成功目标**区域，单击**编辑**。

b. 在**编辑成功目标**面板，单击**启用**，然后配置当函数成功执行后将执行信息发送的目标云服务。参数信息如下：

参数	说明
成功时调用其他服务	启用该功能后，您可以在函数执行成功后将执行信息发送给其他目标服务。
目标服务	函数计算。当目标服务选择的是函数计算时，需配置以下参数信息： ■ **服务名称**：指定目标服务的名称。 ■ **版本或别名**：指定服务的别名或版本。 ■ **函数名称**：指定目标函数的名称。
	。当目标服务选择的是时，需配置以下参数信息： ■ **目标类型**：按需选择目标类型，取值为： ■ **队列**： 队列模型提供高可靠、高并发的一对一消费模型，即队列中的每一条消息都只能够被某一个消费者消费。 ■ **主题**： 主题模型提供一对多的发布订阅模型，支持消息通知。 ■ **队列**：设置的队列名称。当目标类型选择的是队列时需设置此参数。 ■ **主题**：选择的主题名称。当目标类型选择的是主题时需设置此参数。
	消息队列RocketMQ版 ，当目标服务选择的是 消息队列RocketMQ版 时，需配置以下参数信息： ■ **实例**：选择目标实例。 ■ **Topic**：选择目标Topic。
	事件总线EventBridge。当目标服务选择的是事件总线EventBridge时，需指定**自定义事件总线**。

c. 单击**确定**。

- 配置失败目标

a. 在**失败目标**区域，单击**编辑**。

b. 在**编辑失败目标**面板，单击**启用**，然后配置当函数执行失败后将执行信息发送的目标云服务。

您可以参照配置成功目标的参数描述，配置失败目标的参数信息。

c. 单击**确定**。

触发函数后，函数执行成功或失败后，您将从配置的目标服务中读取以下内容：

```
{
    "timestamp": "2020-08-20T12:00:00.000Z",
    "requestContext": {
        "requestId": "xxx",
        "functionArn": "acs:fc:::services/{serviceName}/functions/{functionName}",
        "condition": "FunctionResourceExhausted",
        "approximateInvokeCount": 3
    },
    "requestPayload": "",
    "responseContext": {
        "statusCode": 200,
        "functionError": ""
    },
    "responsePayload": ""
}
```

回调失败的处理

当服务角色无目标服务访问权限或目标服务不可用时，回调目标服务可能会失败。函数计算提供了相关的指标及日志，您可以根据需要进行相应处理。常见的错误及系统行为如下所示：

错误码	错误原因	系统行为
5xx	限流或内容错误等。	函数计算系统内容按指数退避自动重试。初始重试间隔为500毫秒，最大重试时长为30分钟。
4xx	无权限、请求参数不正确（如目标服务的资源已被删除）或请求消息体超过目标服务限额等。	返回错误并记录错误信息。

结果回调指标

当回调目标服务失败后，函数计算会记录相应指标并展示到控制台。您可以登录函数计算控制台，在左侧导航栏选择**高级功能 > 监控大盘**，然后在**服务名称**列表单击目标服务名称，查看服务维度的指标情况。关于目标服务功能的指标，如下所示：

指标名称	描述
目标触发失败（FunctionDestinationErrors）	函数异步调用配置Destination时，函数执行中触发目标失败的请求数。按1分钟或1小时粒度统计求和。
目标触发成功（FunctionDestinationSucceed）	函数异步调用配置Destination时，函数执行中触发目标成功的请求数。按1分钟或1小时粒度统计求和。

更多监控指标，请参见监控指标。

更多信息

您不仅可以通过函数计算控制台配置异步调用目标服务，还可以通过以下方式配置：

- Serverless Devs

  当您使用Serverless Devs管理资源时，您可以将函数的调用方式设置为异步调用，同时配置异步调用相关参数。更多信息，请参见Serverless Devs操作命令。

- SDK

  详细信息，请参见SDK列表。

  您可以通过InvokeFunction接口调用函数。关于配置异步调用API的接口，请参见：

  - PutFunctionAsyncInvokeConfig
  - GetFunctionAsyncInvokeConfig
  - ListFunctionAsyncInvokeConfigs
  - DeleteFunctionAsyncInvokeConfig

3.异步任务

3.1. 功能概览

当您对函数发起异步调用时，相关请求会被持久化保存到函数计算内部队列中，然后被可靠地处理。如果您想追踪并保存异步调用各个阶段的状态，实现更丰富的任务控制和可观测能力，可以选择开启任务模式处理异步请求。本文介绍异步任务的背景信息、使用限制和常用功能。

背景信息

开启异步任务后，您可以实现以下功能：

- 每次函数调用将详细记录调用过程中的状态转换信息，例如调用输入、执行结果和错误信息等。
- 拥有调用级的控制能力，可以主动终止调用等。

异步任务会保存状态信息，因此，函数的调用和执行会有一定的额外延迟，该延迟不会产生额外的费用。关于函数计算计费的详细信息，请参见计费概述。

使用限制

- 场景限制

  异步任务虽然功能更丰富，但相应的系统开销更大。以下场景建议您关闭任务模式：

  - 您对请求处理链路延时非常敏感，需要平均延时在百毫秒以下。
  - 您需要每秒数千甚至更高的速率发起异步调用。

- 地域限制

  异步任务支持华东1（杭州）、华东2（上海）、华北2（北京）、华北3（张家口）、华南1（深圳）、中国香港、新加坡、英国（伦敦）、美国（硅谷）、美国（弗吉尼亚）、德国（法兰克福）、澳大利亚（悉尼）和印度（孟买）地域。

- 时效限制

  仅支持查询7天内的任务状态信息。

功能对比

如果您要自行构建异步任务处理平台，或者实现简单的定时类任务，可以使用Kubernetes的Jobs功能来实现。以下是函数计算异步任务和Kubernetes的Jobs功能对比。

对比项	函数计算异步任务	Kubernetes的Jobs功能
适用场景	适用于任务执行时长数十毫秒的实时任务和任务执行时长数十小时的离线任务。	适用于任务提交速度要求不高，任务负载比较固定，任务实时性要求不高的离线任务。
任务可观测能力	支持。提供日志、任务排队数等指标和任务链路耗时、任务状态查询等丰富的可观测能力。	需自行整合开源软件来实现。
任务实例自动扩缩容	支持。根据任务排队数和实例资源使用率自动扩缩容。	需通过任务队列自行实现扩缩容和实例负载均衡，复杂度较高。

对比项	函数计算异步任务	Kubernetes的Jobs功能
任务实例伸缩速度	毫秒级。	分钟级。
任务实例资源利用率	用户只需要选择合适的实例规格，实例自动伸缩，按实际处理任务的时长计量，资源利用率高。	需在Jobs提交时确定实例的规格和数目。实例难以自动伸缩和负载均衡，资源利用率低。
任务提交速度	单个用户支持每秒提交数万条任务。	整个集群每秒最多启动数百条Jobs。
任务定时或延时提交	支持。	支持任务定时提交，不支持任务延时提交。
任务去重	支持。	不支持。
暂停或恢复任务执行	支持。	仅Kubernetes 1.21以上版本支持。
终止指定任务	支持。	有限支持。通过终止任务实例间接实现。
任务流控	支持。可在用户或任务处理函数等不同粒度进行流控。	不支持。
任务结果自动回调	支持。	不支持。
开发运维成本	只需要实现任务的处理逻辑。	需维护K8s集群。

常见功能

异步任务的常见功能如下所示：

- 任务管理
- 任务去重
- 任务监控
- 重试策略
- 事件触发
- 结果回调
- 任务编排

3.2. 任务管理

本文介绍异步任务的状态以及如何在函数计算控制台管理异步任务，包括创建、启动、停止和查看异步任务。

通过控制台管理任务

在创建函数时创建异步任务

1. 登录函数计算控制台。
2. 在左侧导航栏，单击**任务**。
3. 在顶部菜单栏，选择地域。

4. 在**任务处理函数**页面，单击**创建函数**。
5. 在**创建函数**页面，创建函数并配置触发器，然后单击**创建**。

> 说明
> - 关于创建函数的具体步骤，请参见创建函数。
> - 您可以在**创建函数**页面创建函数的同时配置触发器，也可以待函数创建成功后，在目标函数的**触发器管理**页签，创建触发器。
> - 使用上述方法创建的函数将自动开启**异步策略**的**任务模式**，您后续对该函数发起的异步调用请求将按照任务模式处理。

创建完成后，在**任务处理函数**页面的**函数名称**列表显示您刚才创建的函数，如您想查看具体任务详情，可在右侧**操作**列单击**查看任务**。

为已有函数开启异步任务模式

前提条件：创建函数

1. 登录函数计算控制台。
2. 在左侧导航栏，单击**服务及函数**。
3. 在顶部菜单栏，选择地域。
4. 在**服务列表**页面，单击目标服务。
5. 单击目标函数，在函数详情页面，单击**异步配置**页签。
6. 在**异步配置**页签的**异步策略**区域，单击**编辑**。
7. 在**编辑异步策略**面板，**任务模式**选择**开启**。

启动或停止任务

1. 登录函数计算控制台。
2. 在左侧导航栏，单击**任务**。
3. 在顶部菜单栏，选择地域。
4. 在**任务处理函数**页面，单击目标函数。
5. 在**异步任务列表**页签，单击**提交任务**。

   您也可以单击**提交任务**右侧的图标，从下拉列表中选择**配置任务参数**，事件函数将以event的形式，HTTP函数将以HTTP参数的形式输入参数传递给函数，模拟提交任务。

   提交任务后，刷新页面，您可以看到执行中的任务。您可以按需登录实例、停止任务、重新执行任务和查看日志等。

> 说明 异步任务中，HTTP函数暂不支持重新执行任务。

调用API（SDK）管理任务

创建异步任务

调用PutFunctionAsyncInvokeConfig接口，配置异步调用模式为任务模式。将AsyncConfig配置为如下内容，完成该配置后，该函数的所有异步调用将变为任务模式。

```
{
  "statefulInvocation": true
}
```

> **说明** 您配置异步调用模式为任务模式后，您仍然可以使用同步方式调用函数，但异步任务模式只针对异步调用生效。

开启任务模式

您可以通过调用InvokeFunction接口触发一次异步调用来启动任务模式。您可以再调用时添加HTTP请求头 `X-Fc-Stateful-Async-Invocation-Id` 来为本次调用设置任务ID，更多信息，请参见任务去重。

查看任务

您提交任务后，如需查询某次执行的状态或执行记录等信息，您可以调用GetStatefulAsyncInvocation接口进行查询。

如果您需要根据关键字或条件查询符合条件的任务执行列表，您可以调用ListStatefulAsyncInvocations接口来实现。

停止任务

您提交任务后，可根据需要停止任务。您可以调用StopStatefulAsyncInvocation接口执行停止任务操作。停止任务时需要提供TaskID，此ID为您提交任务时设置的TaskID，也可以是调用ListStatefulAsyncInvocations接口查询运行中任务时，获取的ID。

异步任务的状态

针对每一次任务调用，函数计算均会记录任务的状态变更过程，并提供实时的状态查询能力。您可以通过SDK或函数计算控制台查看任务的具体状态。目前任务有如下几种状态：

执行状态	说明
已入队	异步消息已入队，等待处理。
已处理	异步消息已出队，等待触发。
执行中	调用执行中，您的实例已经开始运行任务代码。
执行成功	调用执行成功。
执行失败	调用执行失败。
已停止	您已手动停止该次任务调用。任务已成功终止。
停止中	您手动停止了该次任务，任务尝试停止中。
已过期	您给异步消息配置了存活有效期，该消息因过期已被丢弃（未触发）。

执行状态	说明
无效	您的执行因函数或服务被删除等原因处于无效状态（未触发）。
重试中	异步调用因执行错误重试中。当系统准备好重试后，您的任务将会再次变为Running状态。

您可以调用GetStatefulAsyncInvocation接口获取某次任务执行的详细信息，也可以调用ListStatefulAsyncInvocations接口过滤指定状态的任务。

3.3. 事件触发

您可以为函数配置触发器，以事件驱动的方式触发任务。所有支持异步方式触发的触发器都可以触发任务。本文介绍支持触发异步任务的触发器类型。

触发器类型

- HTTP触发器
- 定时触发器
- OSS触发器
- MNS主题触发器
- IoT触发器
- Event Bridge触发器

> **注意** 请确保HTTP和Event Bridge触发器的**调用方式**为**异步调用**。

3.4. 任务去重

函数计算支持为每次提交的任务设定全局唯一的ID。当出现不可知的结果时，例如异步调用提交任务接口超时，您可以通过提交相同ID的任务进行重试，任务去重功能可以避免任务的重复执行。本文介绍如何通过设置TaskID来实现任务去重。

功能原理

函数计算提供TaskID这一任务概念，该ID全局唯一。建议您在每次提交任务时指定该ID，并在出现超时等情况下进行重试。函数计算会对您重复提交的任务进行校验，当有相同ID进入系统时，该次请求将认为是重复提交而被拒绝，并返回错误 `400` 。

> **说明** 函数计算还提供了Request ID这一概念，如果您设置了Request ID但未设置TaskID，系统将自动设置TaskID为Request ID。使用异步任务时，建议您设置TaskID，无需设置Request ID。

设置TaskID

您可以在函数计算控制台、使用Serverless Devs或调用API提交一次任务执行。如果需要设置TaskID，请在触发函数执行时添加HTTP请求头 `X-Fc-Stateful-Async-Invocation-Id` 。

以Go SDK为例，触发任务执行时设置TaskID的代码示例如下：

```
import fc "github.com/aliyun/fc-go-sdk"
func SubmitJob() {
    invokeInput := fc.NewInvokeFunctionInput("ServiceName", "FunctionName")
    invokeInput = invokeInput.WithAsyncInvocation().WithStatefulAsyncInvocationID("TaskUUID
")
    invokeOutput, err := fcClient.InvokeFunction(invokeInput)
    ...
}
```

3.5. 任务监控

本文介绍异步任务的可观测性指标，包括任务大盘、任务执行列表和任务监控指标。

任务大盘

您可以在函数计算控制台的**任务**页面查看任务大盘。

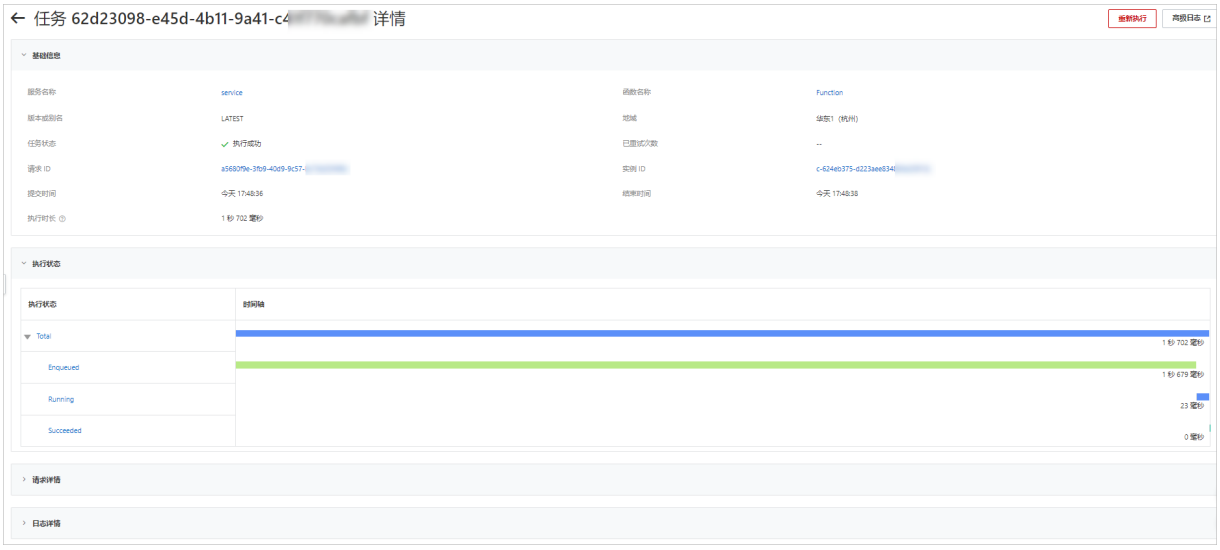
任务大盘提供以下任务监控数据：

监控项	解释说明
提交任务数	在过去1分钟内提交的任务数，包括运行中的、已完成的和未出队的任务。
完成任务数	在过去1分钟内完成的任务数，包括执行成功和执行失败的任务。
排队任务数	当前处于排队中的任务数。如果该任务数不为零，则说明出现任务积压。
运行任务数	当前处于运行中的任务数。
失败任务数	在过去1分钟内运行失败的任务数。
运行实例数	当前执行任务的实例数。

任务执行列表

您可以在函数计算控制台的**任务**页面，单击目标函数，查看指定任务处理函数的异步任务列表，如下图所示。

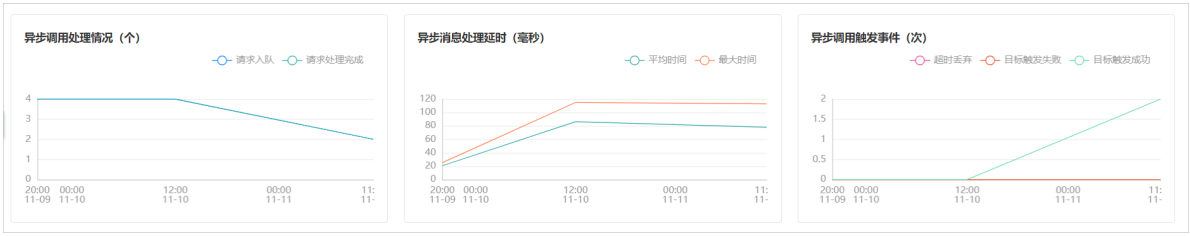
您可以单击**任务 ID**列的目标TaskID，查看指定任务的**基础信息**、**执行状态**、**请求详情**和**日志详情**，如下图所示。



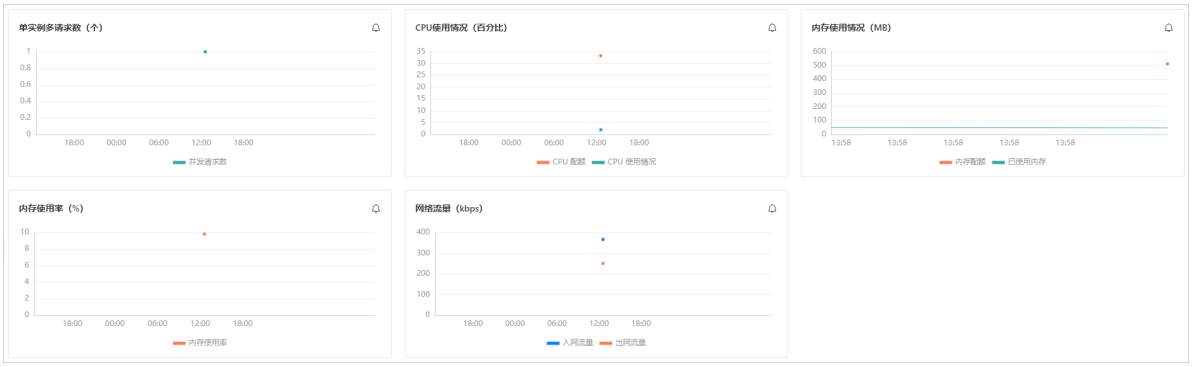
监控指标

您可以登录函数计算控制台，在目标函数的**监控指标**页签，查看异步调用相关指标和异步任务实例级别的资源使用情况。关于监控指标的详细介绍，请参见监控指标。

- 单击**函数指标**页签，查看异步调用相关指标信息，如下图所示。



- 单击**实例指标**页签，查看异步任务实例级别的资源使用情况，如下图所示。



3.6. 任务编排

结合Serverless工作流，函数的异步调用可以应用于大规模复杂场景的任务流程。您可以使用顺序、分支和并行等方式来编排函数计算任务，Serverless工作流会按照设定好的步骤可靠地协调任务执行，跟踪每个任务的状态转换，并在必要时执行您定义的重试逻辑，以确保工作流任务顺利完成。本文介绍如何配置异步任务编排。

前提条件

- 创建异步任务。具体操作，请参见任务管理。
- 授予函数计算访问Serverless工作流的权限。具体操作，请参见授予函数计算访问其他云服务的权限。

操作步骤

1.
2. 在顶部菜单栏，选择地域。
3. 在左侧导航栏，单击**流程**，然后单击**创建流程**。
4. 在**创建流程**页面的**流程定义**配置向导，选择**使用代码段创作**，填写**基本信息**和**流程定义**，然后单击**下一步**。
   - **基本信息**：填写**流程名称**和**（可选）流程备注**。
   - **流程定义**：在**流程定义**的文本框填写配置任务执行的代码段。示例如下所示。

```
version: v1
type: flow
steps:
  - type: task
    name: mytask
    resourceArn: acs:fc:{region}:{account}:services/{serviceName}.{qualifier}/functio
ns/{functionName}
    pattern: sync  # Async invocation with sync pattern
    serviceParams:
        InvocationType: Async
```

5. 在**创建流程**页面的**配置设置**配置向导，配置**流程角色**，然后单击**创建流程**。
6. 在已创建的流程页面，单击**开始执行**。

   通过以上操作，即可触发一次工作流任务。更多信息，请参见集成函数计算异步调用。

4.调用示例

函数计算的2021-04-06及以后版本的API符合阿里云OpenAPI规范，您可以在阿里云OpenAPI Explorer查看和调试API/SDK。本文介绍如何在OpenAPI Explorer调用函数计算的API和SDK。

前提条件

创建函数

调用API

1. 登录OpenAPI Explorer。
2. 在顶部菜单栏，单击**选择云产品**，在搜索框输入**函数计算**，在搜索结果中选择**函数计算**。
3. 在左侧导航栏，找到**调用函数InvokeFunction**。
4. 填写以下参数，然后在**调用结果**页签，单击**发起调用**。

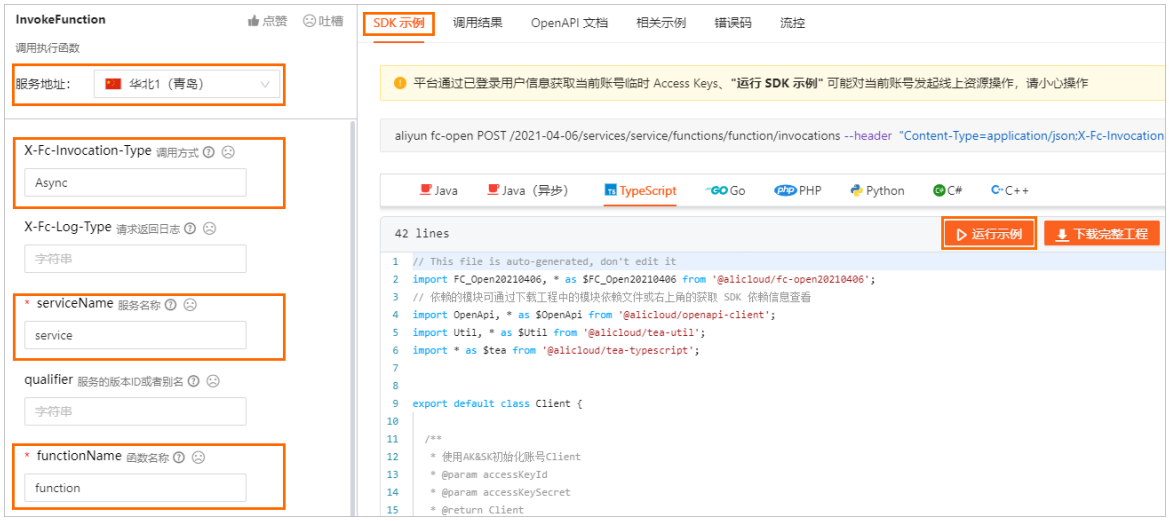
参数说明如下：

参数名称	解释说明
服务地址	选择您要执行的函数所在的地域。
X-Fc-Invocation-Type	填写函数调用类型。取值说明如下： • Sync：同步调用 • Async：异步调用

参数名称	解释说明
serviceName	填写函数所在服务名称。
functionName	填写函数名称。

调用SDK

1. 登录OpenAPI Explorer。
2. 在顶部菜单栏，单击**选择云产品**，在搜索框输入**函数计算**，在搜索结果中选择**函数计算**。
3. 在左侧导航栏，找到**调用函数InvokeFunction**。
4. 填写以下参数，然后在**SDK 示例**页签选择对应的语言，系统自动为您生成示例代码。单击**运行示例**。



参数说明如下：

参数名称	解释说明
服务地址	选择您要执行的函数所在的地域。
X-Fc-Invocation-Type	填写函数调用类型。取值说明如下： ○ Sync：同步调用 ○ Async：异步调用
serviceName	填写函数所在服务名称。
functionName	填写函数名称。

执行完成后，在下方**阿里云云命令行**区域查看执行结果。

5.错误处理

本文介绍函数在同步调用和异步调用执行失败时，如何进行重试完成函数调用。

重试机制

函数未成功执行的重试机制因调用方式而异：

- 同步调用失败

  您需要自行重试。

- 异步调用失败

  函数计算会自动重试的情况如下表所示。

执行失败原因	状态码	服务器端行为	是否计费	解决方案
函数计算的错误类型为 `HandledInvocationError` 和 `UnhandledInvocationError` 。关于函数计算错误类型的更多信息，请参见基础信息。	200	默认重试3次，或根据异步设置次数重试。	按照调用次数计费。关于计费的详细信息，请参见计费概述。	请自行排查您的代码。
函数并发执行超上限。	429	以二进制指数退避方式重试执行5小时。当您的函数执行失败后将在0.5秒后开始重试，后续重试执行的时间间隔将以二进制指数退避方式计算，即重试时间间隔为1秒、2秒、4秒、8秒等持续重试5小时。	否	由于阿里云账号（主账号）在单个地域内默认的按量实例上限数为300。如果您需要提高该限制，请提交工单。
系统内部错误。	500	以二进制指数退避方式重试执行5小时。当您的函数执行失败后将在0.5秒后开始重试，后续重试执行的时间间隔将以二进制指数退避方式计算，即重试时间间隔为1秒、2秒、4秒、8秒等持续重试5小时。	否	请提交工单。

执行失败原因	状态码	服务器端行为	是否计费	解决方案
函数计算资源不足。	503	以二进制指数退避方式重试执行5小时。当您的函数执行失败后将在0.5秒后开始重试，后续重试执行的时间间隔将以二进制指数退避方式计算，即重试时间间隔为1秒、2秒、4秒、8秒等持续重试5小时。	否	请提交工单。

若您在使用过程中遇到问题，请联系我们。

6.弹性管理（含预留模式）

函数计算为您提供了按量模式和预留模式两种实例使用模式。本文介绍两种实例使用模式的功能原理、计费方式、使用场景、实例伸缩限制以及如何在函数计算控制台配置预留模式实例和配置预留模式的弹性伸缩。

按量模式

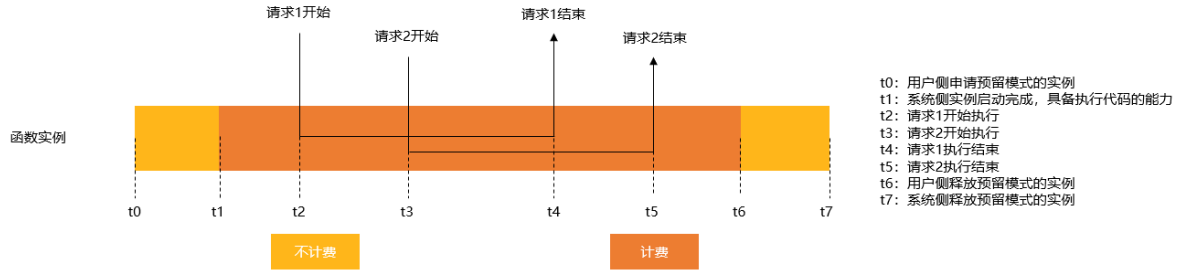
按量模式是指函数实例的分配和释放完全由函数计算系统负责。函数计算会根据函数的调用量自动进行实例扩缩容，在调用增加时创建实例，在请求减少后销毁实例。整个过程完全自动，提高了资源利用率，同时极大地降低了您管理资源的难度。您的阿里云账号（主账号）在单个地域内默认的按量实例上限数为300。如果您需要提高该限制，请提交工单申请。

计费方式：只有发生函数调用时才会产生费用，无函数调用请求就不会分配实例也不会产生费用。关于具体产品定价和计费，请参见计费概述。

预留模式

由于按量模式是通过请求自动触发实例的创建，当首次发起调用时需要等待实例冷启动，如果您希望消除冷启动延时的影响，可以通过配置预留模式来解决。预留模式是将函数实例的分配和释放交由您管理，当配置预留函数实例后，预留的函数实例将会常驻，直到您主动将其释放。函数计算会优先将函数调用请求调度至预留的函数实例，当函数请求的并发超过预留的函数实例处理能力时，超出部分的请求将会转发给按量模式的实例。

计费方式：预留模式的实例的计费从实例成功创建后开始，一直到您主动将其释放为止。由于预留实例是由您自己负责申请和释放，所以即使预留的函数实例未执行任何请求，只要没有释放函数实例，您都需要为预留的函数实例付费。关于具体产品定价和计费，请参见计费概述。



闲置计费（Beta）

默认情况下，闲置计费功能处于关闭状态，此时预留模式的实例无论是否正在处理请求，都会始终为其分配CPU，让实例始终处于活跃状态，以保证实例可以在无请求时正常运行后台任务。开启闲置计费功能后，当预留的实例无请求时，函数计算会将实例的CPU冻结，使该实例进入闲置状态，并以闲置实例资源单价计算使用成本。闲置实例资源使用单价是活跃实例资源使用单价的20%，这将帮助您节省大量的成本。更多信息，请参见计费概述。

使用场景

您可以根据不同的使用场景选择是否启用闲置计费功能。

- 使用成本

  如果您需要预留模式来消除冷启动，又担心预留成本过高，建议启用闲置计费功能。此时，您只需为闲置状态的预留实例支付较少的费用，就能实现无冷启动的响应调用需求。

- 后台任务

如果您的函数需要运行后台任务，建议关闭闲置计费功能。例如：

- 使用依赖于内置调度或后台功能的应用框架，或依赖的中间件需要定期汇报心跳。
- 使用Go语言的Goroutin轻量级线程、Node.js语言的async函数或Java语言的异步线程执行异步操作。

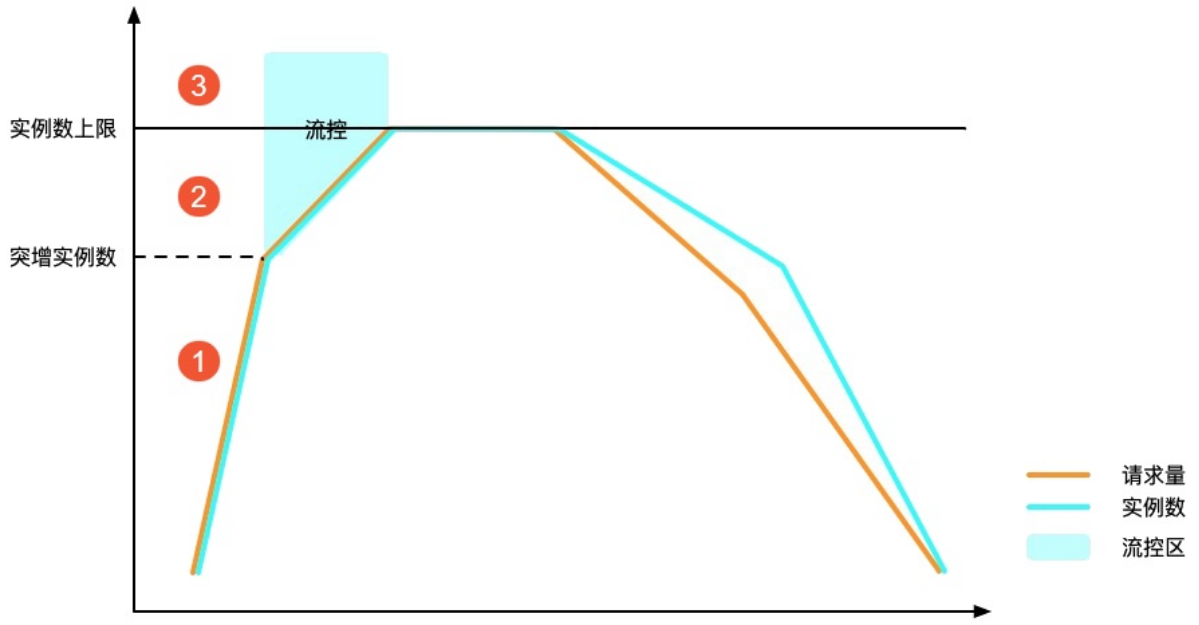
实例伸缩限制

配置按量模式实例的伸缩限制

在处理函数调用请求时，函数计算会优先使用已有的可用实例，若当前实例已经满载，函数计算会创建新的实例来处理请求。随着调用请求量的增加，函数计算会持续创建新的实例，直到有足够的实例处理请求或者达到您设置的实例数上限。在实例扩容的过程中，将受到以下限制：

- 处于执行状态的实例总数，默认限制为每个地域300个。
- 处于执行状态的实例数的扩容速度，受突增实例数和实例增长速度的限制。不同地域的限制条件，请参见各地域扩容速度限制。
  - 突增实例数：可立即创建的实例数，默认限制为100~300个。
  - 实例增长速度：超过突增实例数后每分钟可增加的实例数，默认限制为100~300个。

当实例总数或者实例扩容速度超过限制后，函数计算将返回流控错误（`HTTP Status`为`429`）。下图展示了在一个调用量快速增长的场景下函数计算的流控行为：



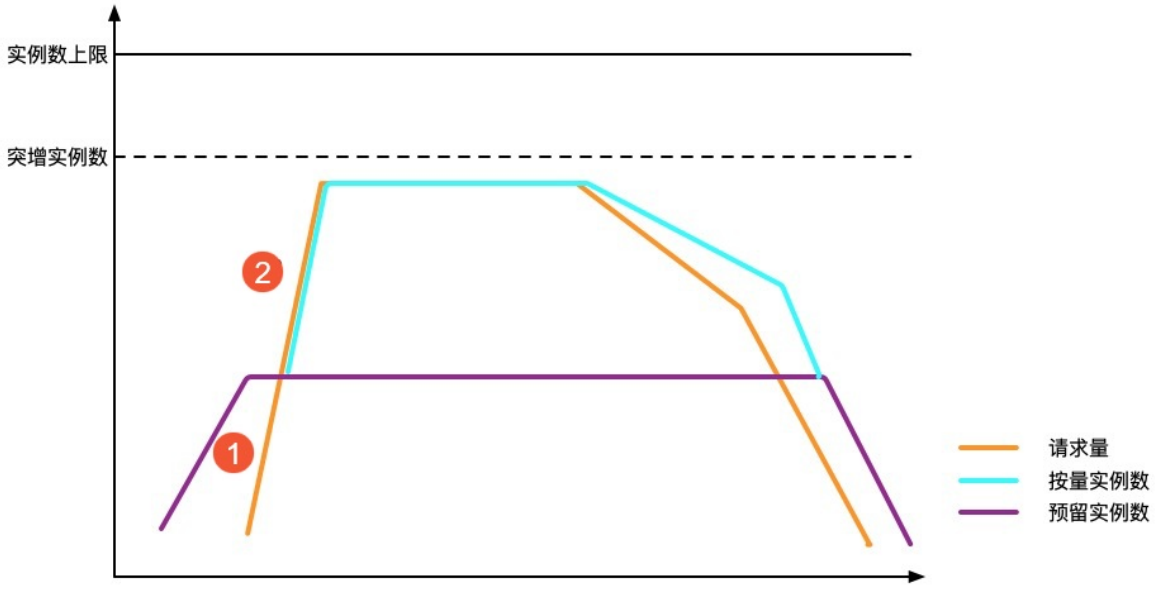
- 在达到突增实例数前，函数计算立即创建实例，这个过程中有冷启动，但没有流控错误（图示中①）。
- 达到突增实例数后，实例数的增长受速度限制，部分请求会收到流控错误（图示中②）。
- 实例数超过限制后，部分请求收到流控错误（图示中③）。

默认情况下，一个阿里云账号在同一个地域下的所有函数共享上述伸缩限制。当需要限制某个函数的实例数时，您可以配置函数级按量实例伸缩控制。配置后，当此函数处于执行状态的函数实例总数超过限制后，函数计算将返回流控错误。

配置预留模式实例的伸缩限制

当突发的调用量较大时，大量的实例创建会受到流控限制导致请求失败，实例的冷启动也会增加请求延时。为避免这些问题，您可以使用函数计算的预留实例，即提前准备好函数实例。预留实例的实例数上限和扩容速度有单独的限制，不受上述实例伸缩限制的影响。

- 预留实例总数：默认每个地域300个。
- 预留实例扩容速度：默认每分钟100~300个实例（不同地域的限制不同）。下图展示了和上面相同的负载场景下，使用了预留实例后函数计算的流控行为：



  - 在预留实例被用满之前，请求立即被执行，这个过程既没有冷启动，也没有流控错误（图示中①）。
  - 在预留实例被用满后，按量实例达到突增实例数之前，函数计算立即创建实例，这个过程中有冷启动，但没有流控错误（图示中②）。

各地域扩容速度限制

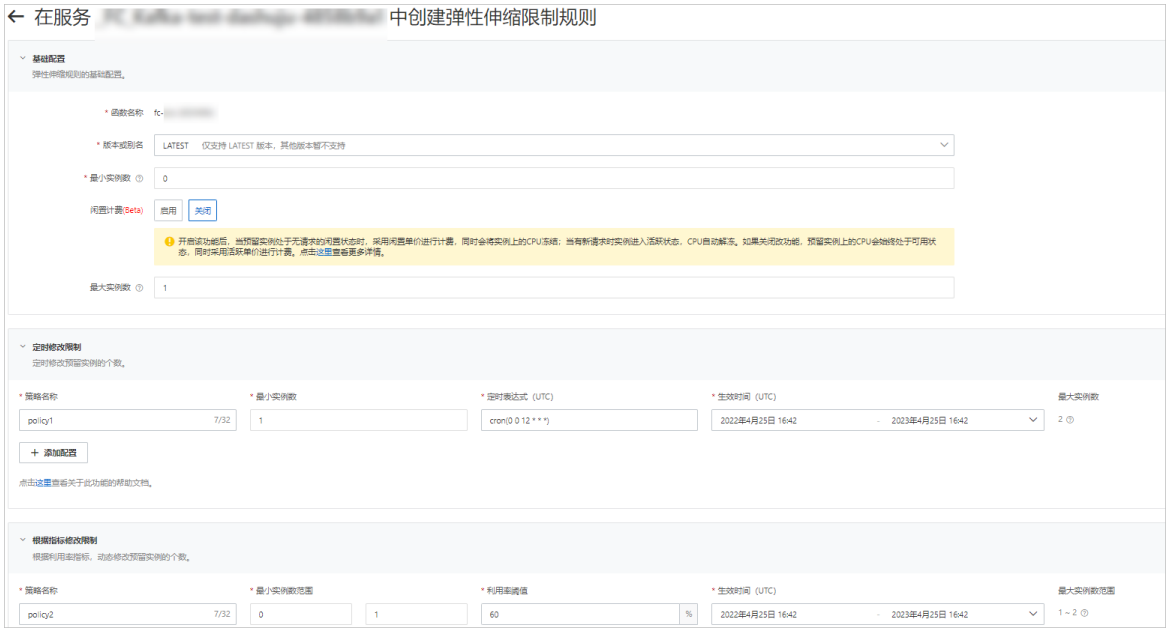
地域	实例扩容速度限制-突增实例数	实例扩容速度限制-实例增长速度
华东1（杭州）、华东2（上海）、华北2（北京）、华北3（张家口）、华南1（深圳）	300	300/分钟
其他	100	100/分钟

> 说明
>
> - 相同地域下，预留模式和按量模式的扩容速度限制一致。
> - 如果您对弹性速度有更高的需求，请提交工单申请。
> - 性能实例和GPU实例的扩容速度小于弹性实例，建议配合预留模式使用。

配置预留实例

1. 登录函数计算控制台。
2. 在左侧导航栏，单击**服务及函数**。

3. 在顶部菜单栏，选择地域。
4. 在**服务列表**页面，单击目标服务**操作**列的**函数管理**。
5. 在**函数管理**页面，单击目标函数名称。
6. 在**函数详情**页面，选择**弹性管理**页签，然后单击**创建规则**。
7. 在创建弹性伸缩限制规则页面，配置相关参数，然后单击**创建**。



参数	说明
基础配置	
版本或别名	在列表中选择需要创建预留模式实例的版本或别名。 **说明** 仅支持在LATEST版本创建预留模式实例，其他版本暂不支持。
函数名称	在列表中选择需要在预留模式的实例上执行的目标函数。
最小实例数	在文本框中填写预留模式的实例的个数。最小实例数=预留实例个数。 **说明** 通过限制函数级别最小实例数来快速响应函数调用请求，降低冷启动的发生次数，为时延敏感的在线业务提供更好的服务响应。

参数	说明
最大实例数	在文本框中填写最大实例数。最大实例数=预留实例个数+按量实例的最大个数。 **说明** ○ 通过限制函数级别最大实例数来防止单个函数过度调用导致的实例占用，保护后端资源，避免预期外的费用开销。 ○ 如果此参数留空，最大实例数限制将遵循您的账号和目前所在地域的最大实例数限制。
闲置计费（Beta）	选择启用或关闭闲置计费功能，默认关闭。参数说明如下： ○ 启用该功能后，仅在处理请求期间为预留模式的实例分配CPU，其余时间实例的CPU将被冻结。 ○ 关闭该功能后，预留模式的实例无论是否正在处理请求都会分配CPU。
（可选）定时修改限制：通过设置定时伸缩您可以更加灵活地配置预留的函数实例，在指定时间将预留的函数实例量设定成需要的值，使函数实例量更好地贴合业务的并发量。	
策略名称	在文本框中填写自定义的策略名称。
最小实例数	在文本框中按需设置预留数量。
定时表达式（UTC）	定时信息，本文示例为cron(0 0 20 * * *)。详细信息，请参见参数说明。
生效时间（UTC）	在文本框中设置定时弹性伸缩的开始生效及结束生效时间。
（可选）根据指标修改限制：根据函数实例并发利用率的情况每分钟对预留资源进行一次伸缩。	
策略名称	在文本框中填写自定义的策略名称。
最小实例数范围	在文本框中按需设置最小实例数的最小值和最大值。
利用率阈值	设置伸缩范围，当利用率低于此参数时则进行缩容，当利用率高于此参数时，则进行扩容。
生效时间（UTC）	在文本框中设置指标弹性伸缩的开始生效及结束生效时间。

创建完成后，在规则列表，您可以看到目标函数下配置的预留模式的实例情况。

更新预留实例

1. 登录函数计算控制台。
2. 在左侧导航栏，单击**服务及函数**。
3. 在顶部菜单栏，选择地域。
4. 在**服务列表**页面，单击目标服务**操作**列的**函数管理**。
5. 在**函数管理**页面，单击目标函数名称。
6. 在**函数详情**页面，选择**弹性管理**页签，找到目标规则，然后在其**操作**列单击**编辑**。

> **说明** 如果需要删除预留模式的实例，您只需将**最小实例数**设置为0即可。

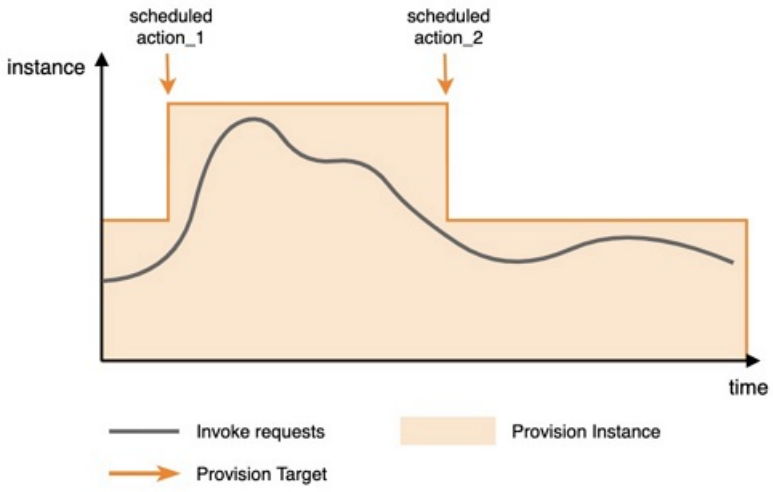
7. 在编辑服务中的限制规则页面，修改基础配置和策略信息，然后单击**保存**。

配置预留模式的弹性伸缩

由于预留模式配置的固定预留值会导致预留函数实例利用不充分，您可以通过定时弹性伸缩和指标追踪弹性伸缩两种模式解决该问题。

定时弹性伸缩

- 定义：通过定时弹性伸缩您可以更加灵活地配置预留的函数实例，在指定时间将预留的函数实例量设定成需要的值，使函数实例量更好地贴合业务的并发量。
- 适用场景：如果您的函数有明显的周期性规律或可预知的流量高峰，可以使用定时预留功能来提前预留函数实例。当函数调用并发大于定时预留值时，超出的部分会分配至按量模式的函数实例。
- 配置示例：如下图配置了两个定时操作，在函数调用流量到来前，通过第一个定时配置将预留函数实例扩容至较大的值，当流量减小后，通过第二个定时配置将预留函数实例缩容到较小的值。



参数示例：

- 为service_1的function_1函数配置定时伸缩，配置的生效区间为：2020-11-01 10:00:00至2020-11-30 10:00:00，在每天20:00将预留函数实例扩容至50，在每天22:00再将预留函数实例收缩至10。

```
{
  "ServiceName": "service_1",
  "FunctionName": "function_1",
  "Qualifier": "alias_1",
  "ScheduledActions": [
    {
      "Name": "action_1",
      "StartTime": "2020-11-01T10:00:00Z",
      "EndTime": "2020-11-30T10:00:00Z",
      "TargetValue": 50,
      "ScheduleExpression": "cron(0 0 20 * * *)"
    },
    {
      "Name": "action_2",
      "StartTime": "2020-11-01T10:00:00Z",
      "EndTime": "2020-11-30T10:00:00Z",
      "TargetValue": 10,
      "ScheduleExpression": "cron(0 0 22 * * *)"
    }
  ]
}
```

- 参数说明如下：

参数	说明
Name	配置的定时任务名称。
StartTime	配置开始生效的时间，UTC格式。
EndTime	配置结束生效的时间，UTC格式。
TargetValue	目标值。
ScheduleExpression	定时信息，支持两种格式： ○ At expressions - "at(yyyy-mm-ddThh:mm:ss)"：只调度一次，使用UTC格式。如：北京时间04月01日20:00开始调度，转换为UTC时间就是04月01日12:00开始调度，则可以使用 `at(2021-04-01T12:00:00)` 。 ○ Cron expressions - "cron(0 0 4 * * *)"：调度多次，使用标准crontab格式，默认以UTC时间运行，即北京时间减去8个小时。如：北京时间每天20:00点进行调度，转化为UTC时间就是每天12:00进行调度，则可以使用 `cron(0 0 12 * * *)` 。

Cron表达式（Seconds Minutes Hours Day-of-month Month Day-of-week）的字段说明如下：

字段说明

字段名	取值范围	允许的特殊字符
Seconds	0～59	无

字段名	取值范围	允许的特殊字符
Minutes	0 ~ 59	, - * /
Hours	0 ~ 23	, - * /
Day-of-month	1 ~ 31	, - * ? /
Month	1 ~ 12或JAN ~ DEC	, - * /
Day-of-week	1 ~ 7或MON ~ SUN	, - * ?

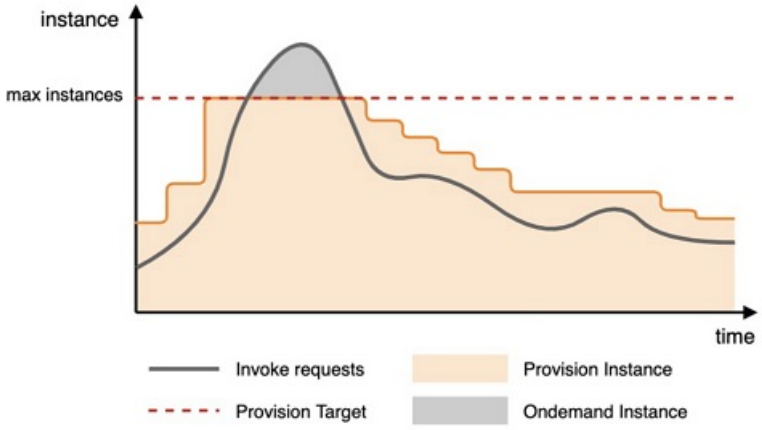
特殊字符说明

字符名	定义	示例
*	表示任一，每一。	在 `Minutes` 字段中：0表示每分钟的0秒都执行。
,	表示列表值。	在 `Day-of-week` 字段中：MON，WED，FRI表示星期一，星期三和星期五。
-	表示一个范围。	在 `Hours` 字段中：10-12表示UTC时间从10点到12点。
?	表示不确定的值。	与其他指定值一起使用。例如，如果指定了一个特定的日期，但您不在乎它是星期几，那么在 `Day-of-week` 字段中就可以使用。
/	表示一个值的增加幅度，n/m表示从n开始，每次增加m。	在 `minute` 字段中：3/5表示从3分开始，每隔5分钟执行。

指标追踪弹性伸缩

- 定义：通过追踪监控指标实现对预留模式的函数实例进行动态伸缩。
- 适用场景：函数计算系统周期性采集预留的函数实例并发利用率指标，使用该指标并结合您配置的扩容触发值、缩容触发值来控制预留模式函数实例的伸缩，使预留的函数实例量更好的贴合资源的真实使用量。
- 实现原理：指标追踪弹性伸缩根据指标情况每分钟对预留资源进行一次伸缩。
  - 当指标超过扩容阈值时，开始以积极的策略扩容预留模式的函数实例量，最快速度将函数实例量扩容至目标值。
  - 当指标低于缩容阈值时，开始以保守的策略缩容预留模式的函数实例量，小幅度向缩容目标值贴近。

  如果在系统中设置了伸缩最大值和最小值，此时预留的函数实例量会在最大值与最小值之间进行伸缩，超出最大值时将停止扩容，低于最小值时将停止缩容。
- 配置示例：



- 当流量不断增加时，触发扩容阈值80%，预留模式的函数实例开始扩容，当达到最大值100时停止扩容，超出部分的请求分配至按量模式函数实例。
- 当流量不断减小时，触发缩容阈值60%，预留模式的函数实例开始缩容。

预留模式函数实例的并发利用率只统计预留模式的并发情况，不包含按量模式的数据。指标口径：预留模式函数实例正在响应的请求并发值与所有预留函数实例最大可响应并发值的占比，数值范围为[0,1]。对于不同的实例并发数，预留模式的函数实例最大可响应并发值的计算逻辑如下所示。关于实例并发数的具体信息，请参见设置实例并发数。

- 单实例单并发：最大可响应并发值=函数实例数量
- 单实例多并发：最大可响应并发值=函数实例数量×单实例并发度

扩缩容目标值：

- 根据当前指标值、指标追踪值、当前预留模式的函数实例数、缩容系数共同决定。
- 扩缩容计算原理：缩容时会通过缩容系数来实现相对保守的缩容过程，缩容系数取值范围为(0,1]。缩容系数为系统参数，用于减缓缩容速度，防止缩容过快，您无需设置。扩缩容目标值对计算结果向上取整得到最终结果，计算逻辑如下：
  - 扩容目标值=当前预留模式的函数实例数×（当前指标值/指标追踪值）
  - 缩容目标值=当前预留模式的函数实例数×缩容系数×（1-当前指标值/指标追踪值）
- 扩容目标值计算示例：当前指标值为80%，指标追踪值为40%，当前预留模式的函数实例数为100，经过计算100×（80%/40%）=200。预留模式的函数实例数会扩容到200，以保证扩容后指标追踪值维持在40%附近。

参数示例：

- 为service_1的function_1函数配置指标追踪弹性伸缩，配置的生效区间为：2020-11-01 10:00:00至2020-11-30 10:00:00，追踪预留模式函数实例并发利用率ProvisionedConcurrencyUtilization指标，并发利用率追踪值为60%，超过60%时开始扩容，扩容上限为100；并发利用率低于60%时开始缩容，缩容下限为10。

```
{
  "ServiceName": "service_1",
  "FunctionName": "function_1",
  "Qualifier": "alias_1",
  "TargetTrackingPolicies": [
    {
      "Name": "action_1",
      "StartTime": "2020-11-01T10:00:00Z",
      "EndTime": "2020-11-30T10:00:00Z",
      "MetricType": "ProvisionedConcurrencyUtilization",
      "MetricTarget": 0.6,
      "MinCapacity": 10,
      "MaxCapacity": 100,
    }
  ]
}
```

- 参数说明如下：

参数	说明
Name	配置的定时任务名称。
StartTime	配置开始生效的时间，UTC格式。
EndTime	配置结束生效的时间，UTC格式。
MetricType	追踪的指标：ProvisionedConcurrencyUtilization。
MetricTarget	指标的追踪值。
MinCapacity	扩容的最大值。
MaxCapacity	缩容的最小值。

7.函数级按量实例伸缩控制

本文介绍函数级按量实例伸缩控制的背景信息、应用场景、使用限制、使用说明以及TPS计算公式。

背景信息

为了防止过度调用函数导致费用失控，每个账号在当前地域中按量实例数存在限制，该限制为账号级别限制，所有函数共享按量实例数的最大限制值。例如，账号164901546557****在某地域的按量实例数上限为300，该账号下有3个函数function-a、function-b、function-c。在某一时刻所有正在处理调用的函数按量实例数之和最大为300。

除了账号级别的实例数限制，函数计算为函数的调用提供了更细粒度的按量调用实例数限制，您可以通过控制台或API设置函数级别实例限制数来防止单个函数过度调用导致的实例占用，保护后端资源，避免预期外的费用开销。例如，账号164901546557****下有函数function-a、function-b、function-c。您可以为函数function-a设置按量实例数上限10，调用函数function-a时最多只能占用10个实例。

应用场景

- 保护函数的正常并发度。

  例如，有function-a、function-b两个函数共享账号级别实例限制数，其中function-a是需要保护的重点业务函数，而function-b有可能被过度调用而影响function-a的正常请求。此时，可以单独为function-b设置实例限制防止function-b抢占大量的按量实例数，使function-a分配不到足够的实例。

- 保护下游服务。

  例如，在函数计算中需要大量访问RDS数据库，由于数据库处理能力有限，您需要保护RDS不被打垮，您可以为访问RDS的函数设置实例限制。

- 禁止异常函数调用。

  例如，如果发现某个函数调用异常，可以设置最大函数实例数为0，禁止其调用。

- 防止过度调用函数。

  例如，浏览器端或客户端用户的操作行为不受控制，设置函数级实例数限制可以防止调用失控而产生意外费用。

- 配合预留模式使用。

  通过设置函数级按量模式实例数限制，配合预留模式实例数，您可以只使用预留模式实例、只使用按量模式实例或混合使用以上两种模式的实例。

设置函数级按量实例限制后的调用行为

调用类型	调用行为
同步调用	函数调用所需要占用的实例数超过所设置的值后，超出的请求会被拒绝，并收到 `ResourceExhausted` 流控错误。
异步调用	函数调用所需要占用的实例数超过所设置的值后，请求不会被拒绝，请求会在队列里以所有实例满负荷执行的速度逐渐被消费。

更多信息，请参见函数调用。

函数级按量实例与预留模式实例的配合使用

如果您给指定的函数分配了预留模式实例资源，则优先使用预留资源，在预留资源用满的情况下，再使用按量实例资源。按量实例资源与预留资源配合使用示例如下。

按量模式实例限制	预留模式实例限制	资源使用情况
0	10	不使用按量模式实例，只使用预留模式实例，最多可用10个预留模式实例。 当前预留模式实例不足以支撑并发请求时，新请求会被流控，收到429错误。
20	0	不使用预留模式实例，只使用按量模式实例，最多可用20个按量模式实例。
50	30	优先使用30个预留模式实例，用满后再使用按量模式实例，最多使用50个按量模式实例，总共最多使用80个实例资源。

使用限制

- 每个账号在当前地域下最多设置100条函数级按量实例数限制规则，每条限制规则的实例限制值不得超过账号级别实例限制值300。
- 函数级按量实例数限制规则必须设置在指定别名或LATEST版本之上，可以针对函数的多个不同别名设置不同的实例限制。

TPS计算公式

TPS指一秒钟内一个函数所能处理的请求数目。您可以结合TPS和您的业务需求，设置函数按量实例数。

TPS的计算公式为：TPS = 1 / DurationInSecond × InstanceConcurrency × MaxInstances

假设一个函数执行平均时长（DurationInSecond）为0.1s，该函数的按量实例数上限（MaxInstances）为5。如果单个实例并发度（InstanceConcurrency）为2，那么这5个函数实例每秒能处理100个（1/0.1× 2 × 5）这样的请求，即TPS为100。

更多信息

关于如何创建函数级按量实例数，请参见弹性管理（含预留模式）。

8.函数调用FAQ

8.1. 我的客户端不关心函数执行结果，我不希望我的客户端一直等函数返回怎么办？

您可以使用函数计算的异步调用，异步调用会将您的请求加入到后端队列，客户端会立即返回。函数计算后端会将队列中的请求做并发调用。各SDK的async Invoke请参见：

- Python SDK
- Node.js SDK
- Java SDK
- Golang SDK