

批量计算

最佳实践

最佳实践

GATK软件分析流程由阿里云和Broad Institute合作提供。Broad Institute提供的GATK流程最佳实践用 workflow 定义语言 (WDL) 编写, 通过批量计算集成的Cromwell workflow引擎解析执行。用户将为作业运行时实际消耗的计算和存储资源付费, 不需要支付资源之外的附加费用。

Broad Institute GATK网站和论坛为GATK工具和WDL提供了更完整的背景信息, 文档和支持。

如果需要执行用WDL编写的通用工作流程, 请参考使用App中—3. 对 cromwell workflow引擎和 WDL 的支持。

* 目前GATK和WDL支持功能在邀测中, 如果需要测试, 请提工单咨询。

1. 准备

(1) 使用 OSS 存储

要在批量计算上运行GATK, 输入输出文件都需要保存在OSS上。所以您需要先开通OSS并且创建好Bucket。注意: 创建的Bucket所在区域, 需要和打算批量计算运行GATK的区域一致。

(2) 安装 batchcompute-cli 命令行工具

```
pip install batchcompute-cli
```

安装完成后, 还需要配置。

具体配置参考[这里](#)

2. GATK 示例

运行这个命令, 生成示例代码:

```
bcs gen ./demo -t gatk
```

这个命令将生成以下目录结构:

```
demo
|__main.sh
|__Readme.md
|__src
```

```
||__PublicPairedSingleSampleWf.inputs.json  
||__PublicPairedSingleSampleWf.md  
||__PublicPairedSingleSampleWf.options.json  
||__PublicPairedSingleSampleWf.wdl
```

运行GATK示例GATK示例使用人类参考基因组的构建38来处理全基因组测序数据。输入文件是未比对的BAM格式。对于这个例子，我们将使用NA12878中的公开数据，这些数据的存储由阿里云免费提供。

现在从您的终端运行示例：

```
bcs asub cromwell gatk-job\  
--input_from_file_WDL src/PublicPairedSingleSampleWf.wdl\  
--input_from_file_WORKFLOW_INPUTS src/PublicPairedSingleSampleWf.inputs.json\  
--input_from_file_WORKFLOW_OPTIONS src/PublicPairedSingleSampleWf.options.json\  
--input_WORKING_DIR oss://luogc-shenzhen/gatkdemo/worker_dir\  
--output_OUTPUTS_DIR oss://luogc-shenzhen/gatkdemo/output
```

这个命令已经在main.sh中写好，您也可以只运行：

```
sh main.sh
```

如果看到以下信息，说明提交成功

```
Job created: job-0000000059DC658400006822000001E3
```

job-0000000059DC658400006822000001E3 即是当次提交作业的ID。

查看作业状态：

```
bcs j # 获取作业列表  
bcs j job-0000000059DC658400006822000001E3 # 查看作业详情
```

查看作业日志：

```
bcs log job-0000000059DC658400006822000001E3
```

验证结果：

要查看工作空间中的过程数据信息：

```
bcs o ls oss://my_bucket/my_key/worker_dir/
```

查看所有输出文件：

```
bcs o ls oss://my_bucket/my_key/outputs/
```

您现在已经在批量计算上运行Broad Institute GATK。

云渲染管理系统

云渲染管理系统（Render Manager简称渲管）是一个开源的web应用，可以帮助用户轻松搭建阿里云上的私有渲染系统，直接调用海量计算资源，一键管控集群规模，在加速渲染任务的同时省去自建集群的烦恼。

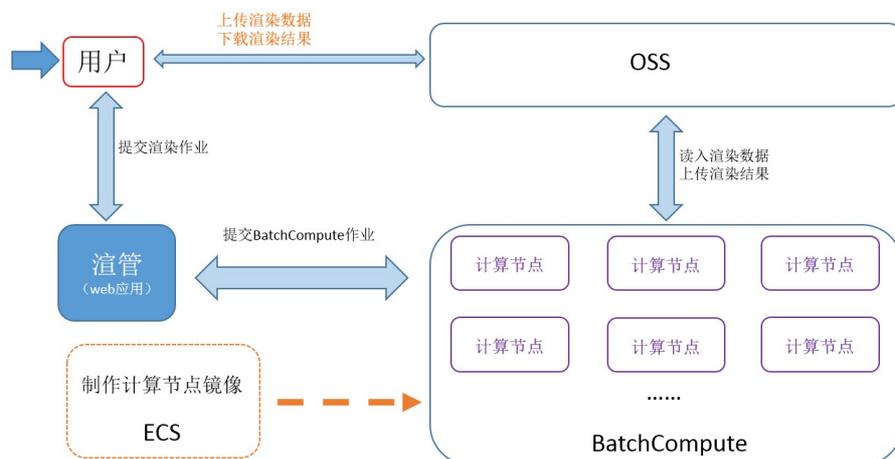


渲管建立在阿里云

BatchCompute、OSS和ECS的三个云产品基础之上的。关于这三个云产品的详细介绍可移步官网介绍，在使用渲管前也请确保已开通上述三个产品。

- BatchCompute是阿里云上的批量计算服务，可以帮助用户进行大规模并行计算
- OSS是阿里云上的对象存储服务，可以存储海量数据
- ECS是阿里云上的云服务器，极易运维和操作,可以方便的制作系统镜像

渲管与这三个云产品的关系如下图



具体使用流程如下：

1. 制作计算节点镜像

根据所要使用的区域创建ECS按量云服务器（也称ECS实例），在云服务器中安装所需的渲染软件后，保存为自定义镜像并将镜像共享给账号1190847048572539，详见计算节点镜像制作章节

2. 上传数据到OSS

将渲染所需要的数据上传到对应区域的OSS并保持上传前的数据目录结构

3. 启动渲管

在ECS控制台创建实例（短期使用选择按量即可），镜像选择镜像市场中的rendermanager（也可以使用渲管安装包进行部署，详见操作手册部署章节）

4. 配置渲管

登录渲管页面<https://ip/rm/login>，配置完基本信息后（AccessKeys和OSS bucket），在镜像管理页中添加上面制作的计算节点镜像ID并对该计算节点镜像配置渲染命令行；

5. 创建项目

在渲管的项目管理页面创建项目，需要指定OSS的数据映射规则（也称OSS挂载，在计算节点启动的时候，OSS上的数据会被挂载到节点的本地路径），选择计算节点镜像ID，OSS的输出路径（用于保存渲染结果），计算节点中的零时输出路径

6. 集群的创建和管理

在集群管理页面可以按需创建集群，需要指定计算节点使用的镜像ID，节点类型和节点数量等信息。

7. 提交渲染作业

在项目页里提交渲染作业，提交时可以指定目的集群（使用AutoCluster时，BatchCompute将按作业的规模自动生成集群，使用AutoCluster需要指定计算节点类型等配置）、渲染的帧范围以及节点数量等信息。提交完作业可以实时查看渲染日志以及节点CPU使用率等信息。

这是一个通过云渲染管理系统进行blender场景渲染的例子。BatchCompute提供了一个公共测试用的计算节点镜像（windows server 2008，ID：m-wz9du0xaa1pag4ylwzsu），里面已安装了blender渲染软件，所以在本次测试中用户无需制作计算节点镜像（实际生产需要用户自己根据实际需求制作合适的计算节点镜像）。使用blender制作的一个小场景已经被上传到OSS（可公共读取，测试时需要下载并上传到自己的OSS bucket里）。演示视频。

1. 准备工作

1. 注册阿里云账号并开通OSS、ECS和BatchCompute服务。
2. 创建AccessKey。账号信息->AccessKeys->创建Access Key，记录Access Key信息（AK为重要信息，小心泄露，正式生产请一定使用限权的子账号AK）。



2. 一个使用blender进行渲染的例子

2.1 创建OSS bucket

阿里云官网->管理控制台->对象存储OSS->创建bucket，按规范起一个bucket名字（假设创建的bucket名字

为renderbucket)，地域选择深圳（华南1），读写权限为私有。



资源包 刷新 新建Bucket

新建Bucket

BucketName :

Bucket命名规范：

- » 只能包含小写字母，数字和短横线
- » 必须以小写字母和数字开头和结尾
- » bucketName的长度限制在3-63之间

所属地域： 华南 1

相同地域内的产品内网可以互通；订购后不支持更换地域，请谨慎选择

读写权限： 私有

- » 私有：对Object的所有访问操作需要进行身份验证
- » 公共读：对Object写操作需要进行身份验证；可以对Object进行匿名读
- » 公共读写：所有人都可以对Object进行读写操作

提交 取消

3. 获取blender场景并上传到自己的OSS bucket中

在浏览器中输入<http://openrm.oss-cn-qingdao.aliyuncs.com/blender/monkey/cube.blend>, 下载场景文件（BatchCompute提供的测试场景）。在OSS控制台创建目录结构blender/monkey，然后在该目录下上传文件，文件路径为oss://renderbucket/blender/monkey/cube.blend。



4. 启动rendermanager

阿里云官网->管理控制台->云服务器ECS->创建实例



选择按量付费，在镜像市场应用开发分类中搜索rendermanager镜像，使用rendermanager镜像并按下图配置购买，可适当提高带宽。（使用按量付费要求用户账户至少有100块金额）。对于地域没有要求，看ECS实际售卖库存情况而定。



购买后，点击进入管理控制

台，在实例列表即可看到刚才启动的云主机（创建会有延迟，请刷新几次）。





5. 登入渲管页面

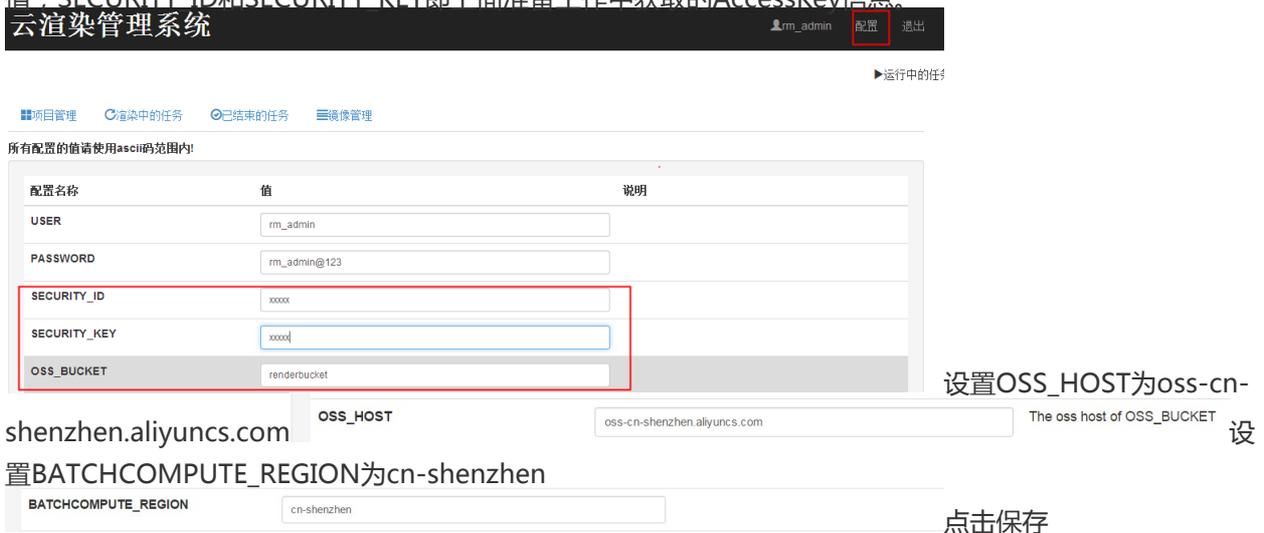
打开浏览器输入 `https://ecs_instance_ip/rm/login`，`ecs_instance_ip`为ECS实例的公网IP（由于使用了https，请在浏览器页面授权信任）。初始账号密码为：

- rm_admin
- rm_admin@123

PS：实际生产请一定更改账号和密码。

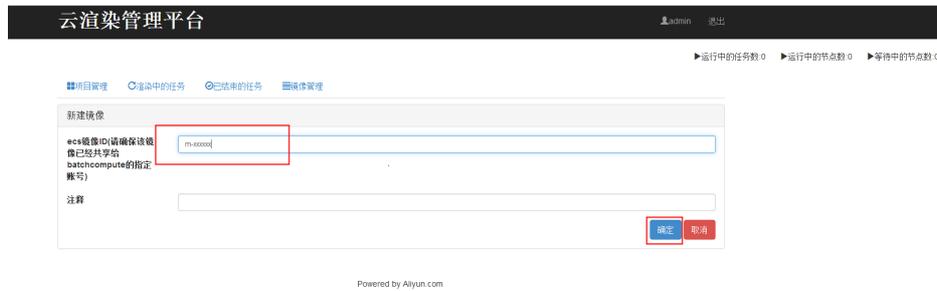
6. 配置渲管

登录后，点击右上角的配置可进入配置页面，填入SECURITY_ID, SECURITY_KEY, OSS_BUCKET三个字段的值，SECURITY_ID和SECURITY_KEY即上面准备工作中获取的AccessKey信息。
云渲染管理系统



7. 添加计算节点镜像

镜像管理->添加计算节点镜像，ECS镜像ID：m-wz9du0xaa1pag4ylwzsu (BatchCompute提供的公用计算节点镜像，实际生产，需要用户自己制作所需要的计算节点镜像，具体制作流程请看后面的操作手册)。

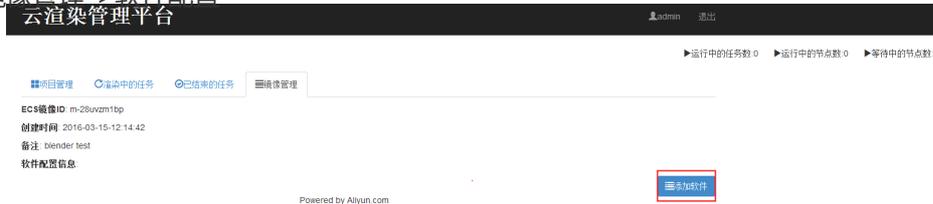


8. 配置渲染软件信息



镜像管理->软件配置

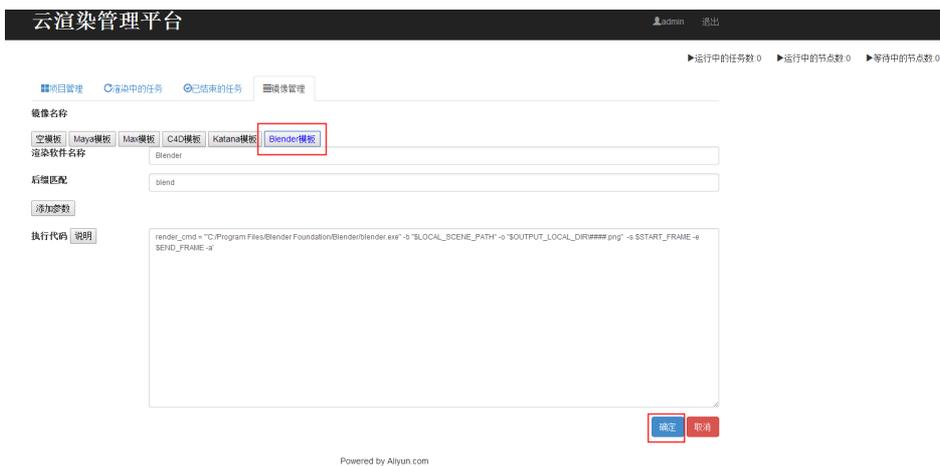
添加软



件

选择blender模板并确定

，执行代码中的render_cmd即是执行渲染的命令

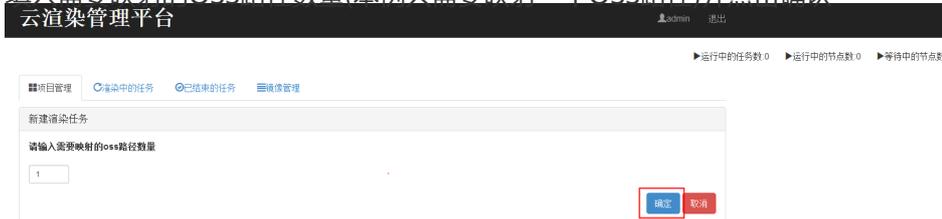


9. 创建项目

1. 项目管理->新建项目



2. 填入需要映射的OSS路径数量(本例只需要映射一个OSS路径)并点击确认



3. 填入项目名称: blender_test。

4. 镜像选择上面创建的镜像。

5. OSS映射中的选择/输入路径为/renderbucket/blender/。

6. OSS映射的目的地为盘符G: (本例中使用的镜像系统为Windows2008 server)。

7. OSS输出目录填写为/renderbucket/rm_test/output/

8. 虚拟机中的输出目录填写为C:\render_output\, 该路径用于渲染节点中临时存放渲染结果, 并且该目录里的渲染结果会被传输到OSS上输出目录里。



9. 确定提交。

10. 提交渲染任务

项目管理->提交渲染
云渲染管理平台



选择场景所在的



OSS路径前缀

点开

项目根目录文件夹，直到看到场景文件cube.blend，选中monkey文件夹可以看到页面下部出现场景选择，勾选场景，选择渲染软件，填入渲染起止帧1~5，并点击提交渲染按钮



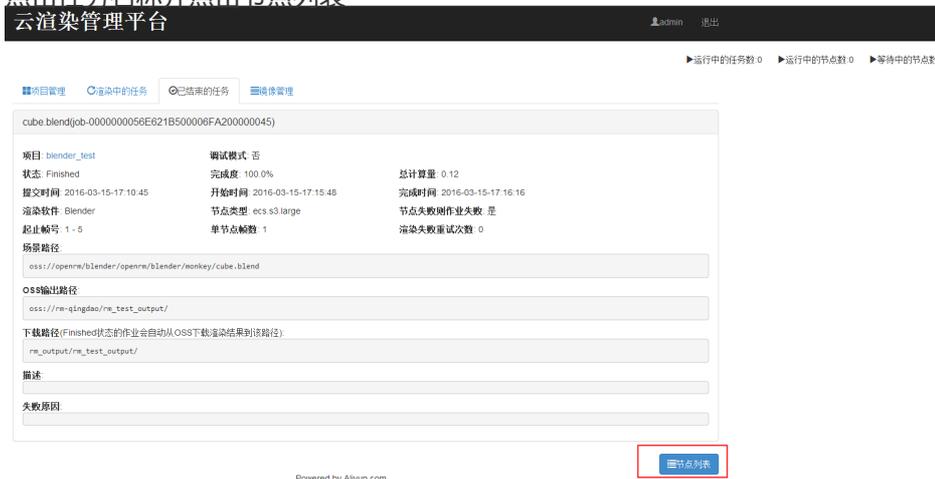
点击渲染中的任务

可查看刚才提交的作业



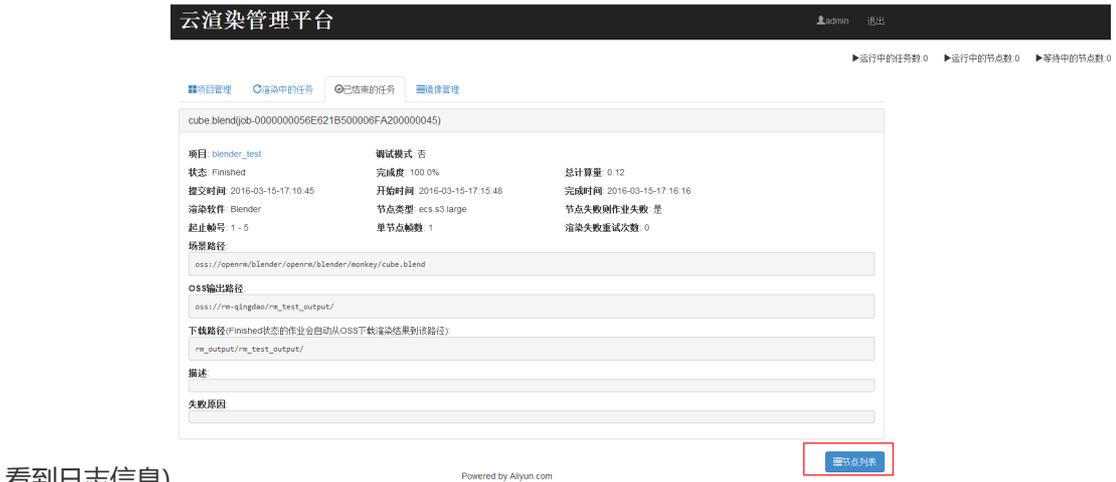
11. 查看渲染日志

点击任务名称并点击节点列表

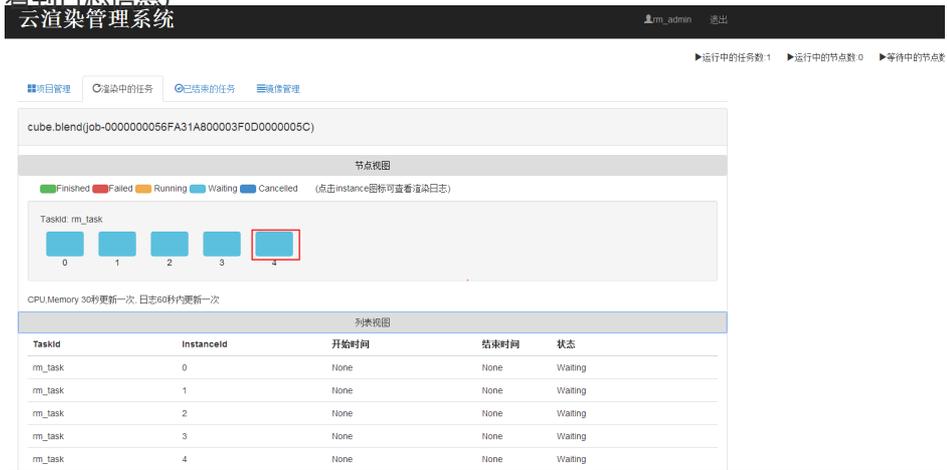


点击想查看的节点

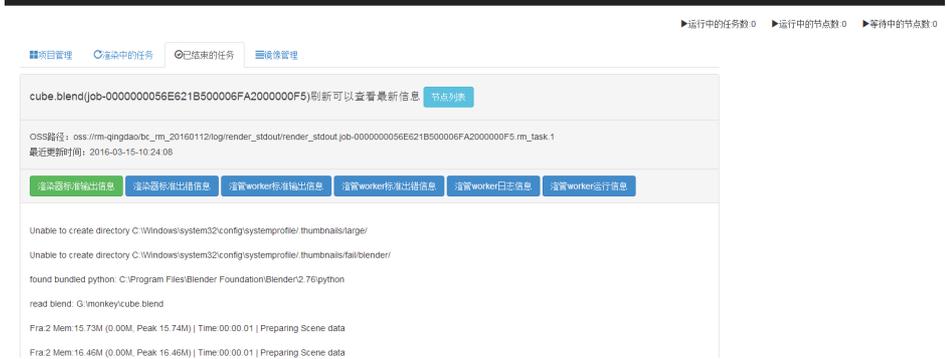
, 可以看到渲染器和渲管worker的各种日志、标准输出以及标准出错信息(计算节点运行起来后才能



看到日志信息

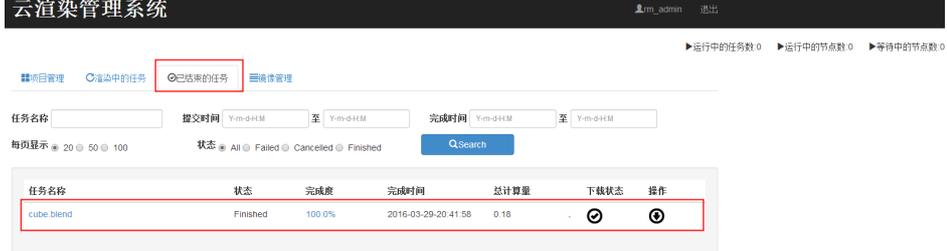


云渲染管理平台



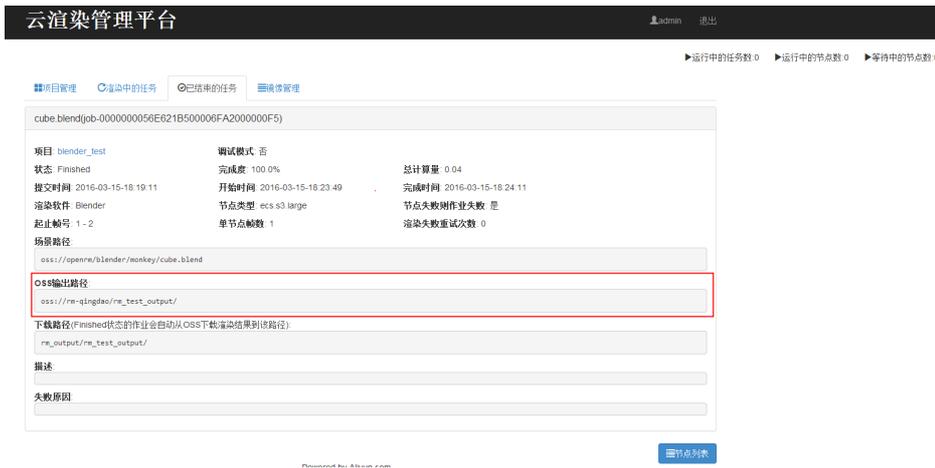
12. 查看渲染结果

等待作业结束后，在已结束的任务中可以可以看到任务状态为Finished



点击任务名称，可

以查看OSS上的输出路径



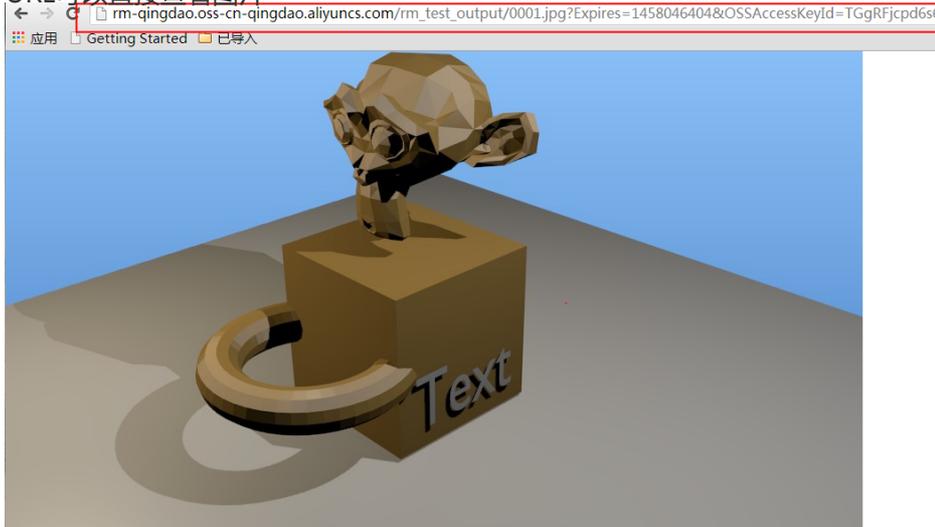
在OSS控制台上查

看对应输出路径，获取地址后点击获取URL并复制URL



在浏览器粘贴

URL可以直接查看图片

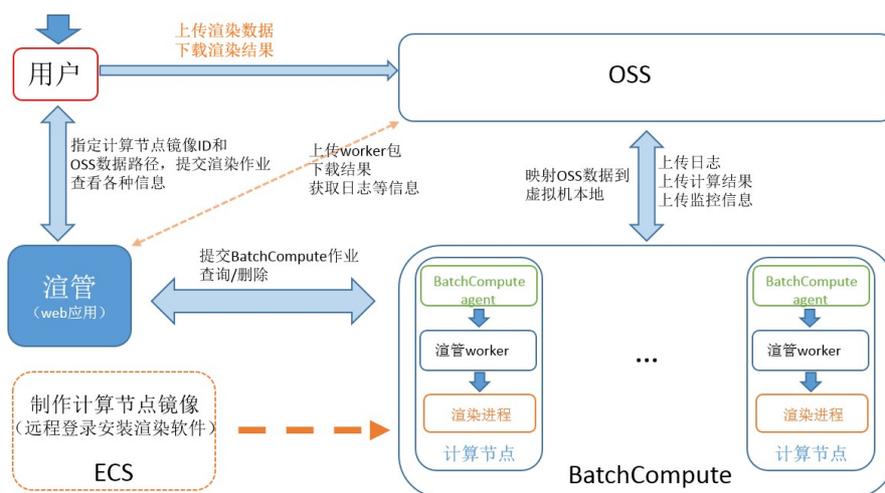


恭喜您已跑通云上

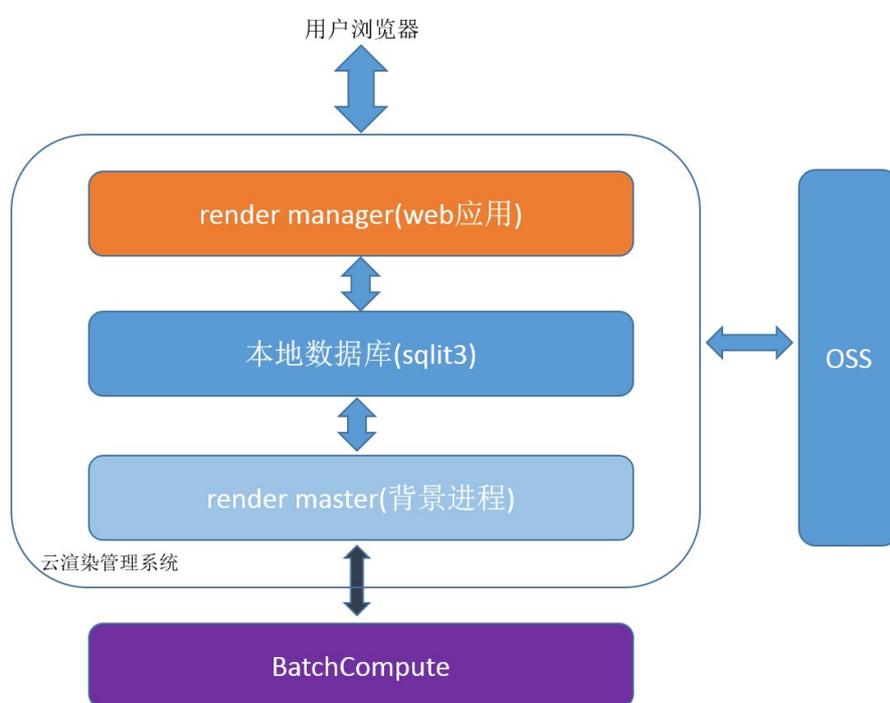
的Blender渲染测试

1. 渲管系统结构

1.1 渲管与各云产品的详细关系



1.2 渲管系统内部结构



render manager: 一个web应用，基于flask框架开发，处理用户的各种操作
 render master：使用python编写的一个背景进程，负责与BatchCompute交互
 本地数据库：存放各种信息，以及render manager与render master之间传递的消息。

2. 渲管的部署

在阿里云云市场里有已经安装了渲管的ECS镜像免费售卖，在启动ECS实例时将镜像指定为镜像市场中的rendermanager，启动即可使用。

2.1 部署流程

建议安装环境：Ubuntu 14.04 64位

```
# 安装flask
sudo apt-get install python-flask -y
# 安装uwsgi
sudo apt-get install uwsgi uwsgi-plugin-python -y
# 安装nginx
sudo apt-get install nginx -y
# 修改nginx配置，在http模块里添加新的server
#
# server {
# listen 1314; #listen port
# server_name localhost;
# location / {
# include uwsgi_params;
# uwsgi_pass 0.0.0.0:8818;#this must be same app_config.xml
# }
# }
#
vi /etc/nginx/nginx.conf
# 启动nginx或重启
nginx
# 获取最新版渲管
wget http://openrm.oss-cn-qingdao.aliyuncs.com/render_manager_release/latest/rm.tar.gz
# 解压
tar -xf rm.tar.gz
# x.x.x为版本号
cd rm-x.x.x
# 指定安装目录部署
python deploy.py /root/rm_install/
# 启动
cd /root/rm_install/rm_install_s && python rm_cmd.py start
# 登陆渲管http://installed_machine_ip:1314/rm/login
# 初始账号：rm_admin 密码：rm_admin@123
# 若监听在公网，建议采用https
```

3. 渲管系统升级



▶运行中的任务数:0 页面右上角的版本信息中可

以查看是否有可升级的新版本，第一次使用渲管前，建议升级到最新版本后再使用渲管（每次只能升级到下一版本，所以升级后请查看是否已是最新版本）。

4. 渲管系统配置



▶运行中的配置页面里有渲管系统的各

种系统设置。第一次使用渲管时，必须先填入SECURITY_ID,SECURITY_KEY,OSS_BUCKET这三个值，不然渲

管无法使用。

- SECURITY_ID和SECURITY_KEY即阿里云账号的AccessKeys信息，可以在阿里云官网控制台创建。
- OSS_BUCKET可以在OSS的控制台创建，用于存储渲管自身的worker包已经渲染数据。渲管默认使用青岛（华北1）区域，如果使用其他区域的BatchCompute,请修改配置中的OSS_HOST(OSS_BUCKET必须与OSS_HOST属于同一个region)与BATCHCOMPUTE_REGION，每个region的OSS_HOST也可以工单咨询获取。

■项目管理 ●渲染中的任务 ●已结束的任务 ■镜像管理 ■集群管理

| 配置名称 | 值 | 说明 |
|--------------|---------------|----------------|
| USER | rm_admin | |
| PASSWORD | rm_admin@123l | |
| SECURITY_ID | | |
| SECURITY_KEY | | |
| OSS_BUCKET | | The OSS bucket |

其他配置项可以看页面上的说明。

5. OSS数据上传

一定要在提交渲染作业前将渲染用到的数据上传OSS，在计算节点启动后再上传的数据将不能在计算节点中访问到。

由于OSS页面控制台上传数据有大小限制，所以上传数据建议使用OSS的命令行工具（类linux系统）、windows客户端或者MAC客户端。这里有更多OSS工具。

6. 计算节点镜像制作

渲染客户需要自己制作计算节点镜像。具体制作的流程如下：

1. 在ECS控制台，点击创建实例进入购买页面（购买链接）
2. 选择按量付费类型和所需的region（与要使用的batchcompute的region相同），可用区随机即可



3. 选择一个已有的安全组或创建一个新安全组
4. 按需选择实例规则（提交渲染作业时重新指定实例规格）
5. 带宽选择按量付费并将峰值适当调大（目前流入ECS实例的流量不收钱，流出收费）



6. 镜像选择镜像市场中运行环境类目中的batchcompute基础镜像（搜索batchcompute即可看到，根据所需的操作系统选择，基础镜像中已内置了batchcompute的基础组件）



7. 系统盘默认40G（可以按需选择合适的系统盘大小）。

8. 设置管理员（windows是Administrator，linux是root）密码，并记住密码



9. 确认购买并开通

开通成功后，进入管理控制台



等ECS实例状态为运行中后，可以通过实例的公网IP远程登录（windows账号为Administrator，linux为root，密码即创建实例时设置的密码），登录方法有：

- Linux类操作系统可以ssh
- Windows操作系统可以使用远程桌面（在运行中输入mstsc）



- 所有类型的实例都可以通过实例页面上的连接管理终端选项登入ECS实例。

12. 远程登录后，在实例中安装渲染软件以及各种插件，将测试用的渲染数据拖进ecs实例中，并用渲染命令行测试（window在运行里输入cmd可以调出命令行窗口），确保可进行命令行渲染。
13. 将ecs实例关机，待实例状态为停止后，在实例信息页面点击创建自定义镜像选项。



14. 等待镜像完成度为100%时，将镜像共享给账号：1190847048572539。
15. 在不需要该ecs实例时，可以选择立即释放该实例，因为不释放将会产生持续的费用。



7. 计算节点镜像管理

7.1 添加计算节点镜像

在镜像管理页面可以添加计算节点镜像ID

7.2 给计算节点镜像配置渲染软件信息

在添加完计算节点镜像ID后，在镜像信息页面可以点击添加软件并配置软件信息。



在配置软件信息时，需要填

入渲染软件的名称，渲染文件的后缀（用于识别渲染文件）以及执行代码。执行代码（要求python语法）是一段会在渲管worker中执行的代码，render_cmd变量即渲染时的命令行，命令行应根据实际安装的渲染软件来填写，比如渲染软件的路径，以及一些参数。渲管中的模板只是个示例，实际使用需要微调。



和函数，在执行代码中可以调用这些变量和函数，例如\$CPU在执行期会被替换成实际的cpu核数，\$START_FRAME在执行期会被替换成起始帧号。如果想增加自定义参数，可以选择添加参数，添加的自定义参数会需要在提交作业时填入。关于所有的可用变量可在软件配置页面点击查看。\$OUTPUT_LOCAL_DIR这个变量即创建项目时配置的节点内临时输出路径，渲染的输出结果应该放在该路径下（大部分渲染器都支持在命令行中指定输出路径），在渲染结束后该目录下的数据会被传输到OSS。

8. 项目管理

8.1 项目创建

创建项目时需要指定OSS数据映射，计算节点镜像，虚拟机内的临时输出路径，OSS输出路径。

8.1.1 计算节点镜像

创建项目时选择的计算节点镜像（需要先在镜像管理页面添加计算节点镜像）是提交AutoCluster作业时使用的镜像，如果提交作业时指定了集群（在集群管理页面可以创建）则作业直接跑在所指定的集群中。

8.1.2 OSS数据映射

OSS数据映射（或者称OSS数据挂载），可以将OSS上的数据映射到计算节点的本地路径（windows是盘符），一个作业中的所有计算节点可以共享访问到相同的数据。OSS数据挂载有如下功能或限制：

1. 映射的目的路径必须根据计算节点镜像实际的操作系统类型进行填写，否则会导致挂载失败，windows只能映射到盘符（例G:），linux必须是绝对路径。
2. 可共享读取访问OSS上的数据。
3. 不支持修改OSS上已存在的文件和文件夹名称。
4. 选择WriteSupport后，支持本地（挂载路径下）文件和文件夹的创建，以及新建文件的修改。
5. 挂载的本地路径里的改动只是本计算节点可见，不会同步到OSS。
6. 在Windows系统中，在挂载时刻已存在的文件夹中创建的文件或文件夹将不支持删除操作，linux系统可以。
7. 选择LockSupport后，将可以使用文件锁功能（只影响windows）。

8. OSS数据挂载会有分布式cache（集群内），所以在大规模并发读取数据时性能较好（能达到10MB~30MB，200台并发，读取20G数据）。
9. OSS路径必须以'/'结尾。

8.1.3 OSS输出目录与临时本地输出目录

渲染作业结束时，计算节点中的临时输出目录中的数据将会被传输到OSS输出目录中。临时输出路径格式必须与节点的操作系统类型对应，不然会出错。

8.2 提交渲染任务



[新建项目](#)

选择目的集群和场景所在的

OSS路径前缀后进入提交的详细页面，选中场景文件的上一级目录，可以被提交渲染的场景文件则会被列出，勾选想要渲染的文件，选择配置的渲染软件和起止帧，即可提交渲染作业。可指定节点数量，如果指定集群，并发数量上限是集群的节点数上限。填入的起止帧会均匀的分布在各个计算节点被渲染。



任务结束后可以在OSS上查

看输出结果，如果开启自动下载（配置页面设置），渲管会在任务结束后将OSS上的输出结果下载到渲管部署的机器上。

8.3 渲染日志

在节点列表页面，点击节点可以查看各种日志，渲管worker日志里都是渲管系统worker的日志，里面可以查看该计算节点中运行的实际渲染命令行。渲染器标准输出和渲染器标准输出里的日志就是渲染软件的输出日志。



9. 调试

新启动的渲管需要进行配置，并进行调试然后再提交大规模的渲染任务。配置完，应该先提交1帧测试任务，查看错误日志（渲管worker日志和渲染器标准输出）调整渲染软件配置（主要是修改渲染命令行），走通全流程并确认结果没有问题后才进行正式生产渲染。当作业失败的时候可以在作业信息中查看失败原因项。

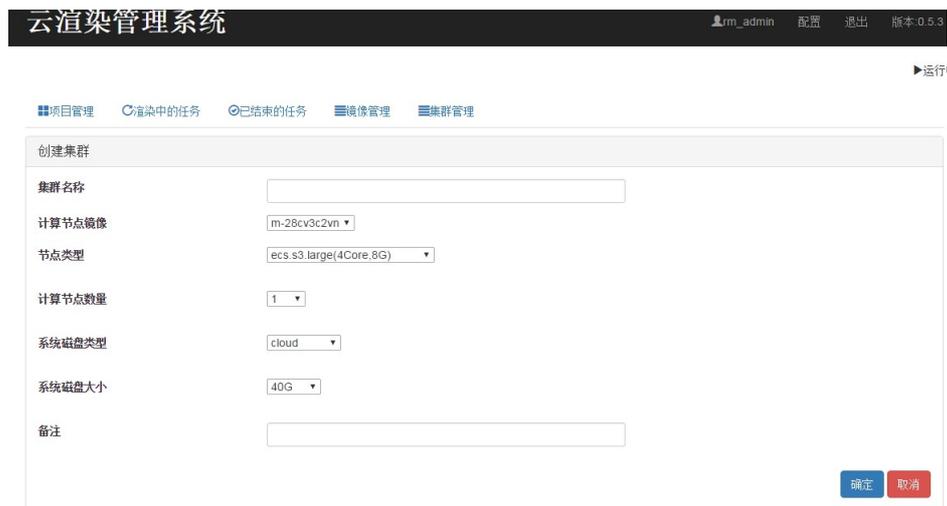


建议创建一个集群然后将作

业提交到该集群进行调试（AutoCluster的作业需要启停计算节点，比较费时）

10. 集群管理

在集群管理页面可以创建自定义集群，需要选择所需的计算节点镜像ID，节点的实例类型（BatchCompute的不同区域可能支持的实例类型和磁盘类型不同，详细可以提工单咨询），磁盘类型和磁盘大小（根据实际制作的计算节点镜像的磁盘大小选择，选择过小会导致无法启动计算节点）。创建好的集群可以动态调整节点数量



, 甚至调整数量到0。

A：使用自定义集群，可长期维持在一定数量，满足日常的渲染需求，当波峰来临时，可以提交AutoCluster任务或者调高集群规模（波峰过去调低数量），省钱又省力。

Q：制作完场景后我要上传哪些数据到OSS？

A：场景文件，还有场景引用了的贴图、素材及渲染中使用的其他数据，建议在制作场景时所有使用的数据和场景文件都在一个目录里，这样上传一个目录即可。要保证在镜像中访问数据的路径同制作场景时相同，有些渲染软件也可设置素材路径。

Q：我的计算节点可以连接公网么？

A：目前BatchCompute启动的计算节点只有内网IP，无法连接公网，但同一个作业里的计算节点可以互相连通。

Q：渲染软件需要连接licence server怎么办？

A：由于batchcompute启动的渲染节点是无公网IP的虚拟机，所以对于需要连接licence server的渲染软件，可以直接将licence server做在镜像里，这样每个计算节点都会有一个licence server。

Q：我想一个阿里云账号部署多个渲管怎么办？

A：在配置中将RENDER_FLAG设置成不同的值，千万不要使用同一个RENDER_FLAG部署多份渲管实例，会出错的。

Q：我的作业跑的时间超出1天怎么办？

向batchcompute客服提工单增加timeout的quota，并且修改配置中RENDER_TIMEOUT值。

Q：提交的作业失败了，渲染器标准输出为空，怎么办？

A：在节点日志页面，查看worker运行信息以及其它几个日志信息，相信能找到蛛丝马迹。

Q：我制作的场景使用的很多贴图分布在各个路径，渲染时如何办？

A: 上传数据到OSS时, 保持目录结构, 在数据映射时填好前缀 (可能需要多个映射), 尽量保证在计算节点中看到的渲染数据文件结构与制作时一样。

Q: 我制作的场景使用了远程的文件怎么办 (windows) ?

A: 制作镜像时, 将远程nas的名字设置成本地机器的别名, 在执行代码中执行命令将目标文件夹共享, 如果数据小, 也可以直接将数据制作进镜像, 并共享。

Q: 我的数据分布在多个盘符 (windows) 里怎么办?

A: 在创建项目时, OSS数据映射项, 直接映射多个盘符。

Q: 虚拟机内临时输出路径必须在C盘下么 (windows) ?

A: 是的, 虚拟机只有一个C盘 (默认40G)。

Q: 系统盘40G不够大怎么办?

A: 在制作计算节点镜像时可以使用更大的系统盘, 在使用该计算节点镜像创建集群时也需要选择足够大的磁盘容量, 但使用超过40G的磁盘, BatchCompute可能会收取少量费用。

Q: 我想节点并发数量大于100怎么办?

A: 提工单给BatchCompute修改配额, 并在渲管配置页面修改MAX_NODE_NUM。

Q: 对于集群和AutoCluster有什么使用建议么?

A: 看场景。AutoCluster类型的作业每个节点都要经历启停 (启停时间在分钟级别), 对于运行时间很短的任务比较不划算, 而且可能因为资源紧张而等待, 大量小任务建议创建集群进行渲染。对等待时间有要求的用户也应该使用自定义集群, 这样提交任务到该集群, 马上就可以运行。但AutoCluster的任务不用担心集群利用率的问题。

Q: 我是程序员, 我可以改代码么?

A: 渲管是开源的 (apache 2.0), 想怎么改怎么改, 记得贡献回社区哦。