批量计算

最佳实践

最佳实践

GATK支持

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 工作流定义语言（WDL） 编写，通过批量计算集成的 Cromwell 工作流引擎解析执行。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 cromwell 工作流引擎和 WDL 支持的 APP 。

1. 准备

A) 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。所以，需要先开通 OSS 并创建好 Bucket。

> 注意：创建 Bucket 的区域,需要和运行批量计算的 GATK 区域一致。

B) 安装 batchcompute-cli 命令行工具

```
pip install batchcompute-cli
```

安装完成后，还需要 配置 。

2. 快速运行

本示例中，运行 Broad Institute 提供的 GATK4 版本全基因分析流程，该流程分为两步：

- 第一步为 gatk4-data-processing 。
- 第二步为 gatk4-germline-snps-indels 。

在配置好 bcs 工具后，执行如下命令：

```
bcs gen ./demo -t gatk
cd demo/gatk4-data-processing
sh main.sh # 运行gatk4-data-processing 流程
cd ../gatk4-germline-snps-indels
sh main.sh # 运行gatk4-germline-snps-indels 流程
```

这样您就在批量计算上运行了以上两个 GATK4 流程。

3. 命令详解

A) 生成示例

执行如下命令生成示例：

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

它将生成以下目录结构:

```
demo
|-- Readme.md
|-- gatk4-data-processing
| |-- main.sh
| |-- src
| |-- LICENSE
| |-- README.md
| |-- generic.batchcompute-papi.options.json
| |-- processing-for-variant-discovery-gatk4.hg38.wgs.inputs.json
| |-- processing-for-variant-discovery-gatk4.hg38.wgs.inputs.30x.json
| |-- processing-for-variant-discovery-gatk4.wdl
|-- gatk4-germline-snps-indels
|-- main.sh
|-- src
|-- LICENSE
|-- README.md
|-- generic.batchcompute-papi.options.json
|-- haplotypecaller-gvcf-gatk4.hg38.wgs.inputs.json
|-- haplotypecaller-gvcf-gatk4.hg38.wgs.inputs.30x.json
|-- haplotypecaller-gvcf-gatk4.wdl
```

- gatk4-data-processing 目录中包括了运行 gatk4-data-processing 流程所需的所有配置和脚本。
- gatk4-germline-snps-indels 目录中包括了运行 gatk4-germline-snps-indels 流程所需的所有配置和脚本。
- 每个目录下面的 main.sh 脚本封装了使用 bcs 工具提交作业的命令。
- src 目录下面包括了工作流实现代码。

B) 运行 gatk4-data-processing 流程

进入 demo/gatk4-data-processing 目录，运行 main.sh，该文件内容如下：

```
#!/bin/bash

# bcs asub cromwell -h for more

bcs asub cromwell gatk-job\
--config ClassicNetwork=false\
--input_from_file_WDL src/processing-for-variant-discovery-gatk4.wdl\
--input_from_file_WORKFLOW_INPUTS src/processing-for-variant-discovery-gatk4.hg38.wgs.inputs.json\
--input_from_file_WORKFLOW_OPTIONS src/generic.batchcompute-papi.options.json\
--input_WORKING_DIR oss://demo-bucket/cli/gatk4_worker_dir/\
--output_OUTPUTS_DIR oss://demo-bucket/cli/gatk4_outputs/\
-t ecs.sn1.large -d cloud_efficiency
```

其中，部分参数描述为：

- input_from_file_WDL： WDL 流程描述文件路径。
- input_from_file_WORKFLOW_INPUTS：WDL 流程输入文件。
- input_from_file_WORKFLOW_OPTIONS：WDL 流程选项文件。
- input_WORKING_DIR：OSS上的目录，用来存储 WDL 流程中各个步骤生成的文件，bcs 会自动给您生成一个默认的路径。
- output_OUTPUTS_DIR：OSS 上的目录，用来存储 WDL 流程结束后生成的 metadata 文件，bcs 会自动给您生成一个默认的路径。

其他参数，请参考 bcs asub -h 命令。

如果希望使用此流程来运行自己的数据，需要修改 src/processing-for-variant-discovery-gatk4.hg38.wgs.inputs.json 文件中的 PreProcessingForVariantDiscovery_GATK4.flowcell_unmapped_bams_list 参数，指定存储在 OSS 上的 ubam 文件。

> 注意：该示例中的流程输入文件不是 FASTQ 格式，而是 unaligned BAM 文件。

C) 运行 gatk4-germline-snps-indels 流程

该流程的运行与 gatk4-data-processing 流程类似，参考上述章节。

- 如果希望使用此流程来运行自己的数据，需要修改 src/haplotypecaller-gvcf-gatk4.hg38.wgs.inputs.json 文件中的 HaplotypeCallerGvcf_GATK4.input_bam 参数，修改为 gatk4-data-processing 流程输出的 bam 文件路径。
- 将 HaplotypeCallerGvcf_GATK4.input_bam_index 参数修改为相应的索引文件路径。

4. 作业状态查询与日志

在提交作业后，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
```

5. 验证结果

查看 OSS 空间中的输出数据：

```
bcs o ls oss://demo-bucket/cli/gatk4_worker_dir/
```

查看 metadata 文件：

```
bcs o ls oss://demo-bucket/cli/gatk4_outputs/
```

6. 如何分析 30X 的全基因组数据

A) 生成配置文件

执行上述步骤生成本示例时，会同时生成一个适用 30X 全基因组数据分析的配置：

- processing-for-variant-discovery-gatk4.hg38.wgs.inputs.30x.json
- haplotypecaller-gvcf-gatk4.hg38.wgs.inputs.30x.json

B) 修改 processing-for-variant-discovery-gatk4 配置文件

为分析 30X 样本，需要将 processing-for-variant-discovery-gatk4.hg38.wgs.inputs.30x.json 文件中的 PreProcessingForVariantDiscovery_GATK4.flowcell_unmapped_bams_list 参数改为OSS 文件路径，该文件包括了需要分析的 30X 样本在 OSS 上的路径列表。

> 注意，30X 数据样本，格式为 unaligned BAM 文件。

C) 修改 gatk4-data-processing 流程文件

找到 gatk4-data-processing 流程的 main.sh 文件，将其中的 --input_from_file_WORKFLOW_INPUTS 参

数，修改为 src/processing-for-variant-discovery-gatk4.hg38.wgs.inputs.30x.json，加上 --timeout 172800 参数，并提交作业。

D) 修改 haplotypecaller-gvcf-gatk4 配置文件

- 将 haplotypecaller-gvcf-gatk4.hg38.wgs.inputs.30x.json 中的 HaplotypeCallerGvcf_GATK4.input_bam 参数修改为gatk4-data-processing 流程输出的 bam 文件路径。
- 将 HaplotypeCallerGvcf_GATK4.input_bam_index 参数修改为相应的索引文件路径。

E) 修改 gatk4-germline-snps-indels 流程文件

找到 gatk4-germline-snps-indels 流程的 main.sh，将其中的 --input_from_file_WORKFLOW_INPUTS 参数修改为 src/haplotypecaller-gvcf-gatk4.hg38.wgs.inputs.30x.json，加上 --timeout 172800 参数，并最后提交作业。

如遇到 QuotaExhausted 错误，请通过工单调整 Quota。

SGE集群支持

批量计算支持自动化搭建 Sun Grid Engine（SGE）集群，批量计算使用的是 CentOS 自带的 SGE 版本，请参考 SGE 。

批量计算提供了名为 BatchCompute SGE 的公共镜像，使用该镜像可快速、可靠的构建 SGE 集群，具体的流程如下：

1. 获取 BatchCompute SGE 镜像

请在云市场 搜索关键字 BatchCompute SGE 了解该镜像，它完全免费使用，使用流程请参考 如何通过镜像创建实例 。

2. 自定义镜像（可选）

本步骤可选，如对镜像没有特殊需求，可直接进入下一步。如果需要在此系统镜像基础上安装软件，必须基于 BatchCompute SGE 制作自定义镜像，请参考 自定义镜像 。

- 必须在 BatchCompute SGE 镜像基础上制作新镜像。
- 制作镜像过程中，请务必不要执行任何有关 SGE 和 bcc 工具的命令，并且不要更新 python。

3. 准备 SGE Master 节点

请指定某 ECS VM 作为 SGE 系统的 Master 节点，它负责管理整个集群，也可以充当提交作业的节点。如果采用自定义镜像，在启动 VM 时要选用自定义镜像，否则选用 BatchCompute SGE 镜像。

配置参数，请参考 创建 Linux 实例。

> 由于 Master 节点需要长期稳定运行，建议在启动 VM 时选用包年包月的付费方式；如果是测试，建议使用按量方式。

详细步骤如下:

A) 创建 VPC 和交换机

> 如果您需要使用已经存在的 VPC ，可以跳过这一步。

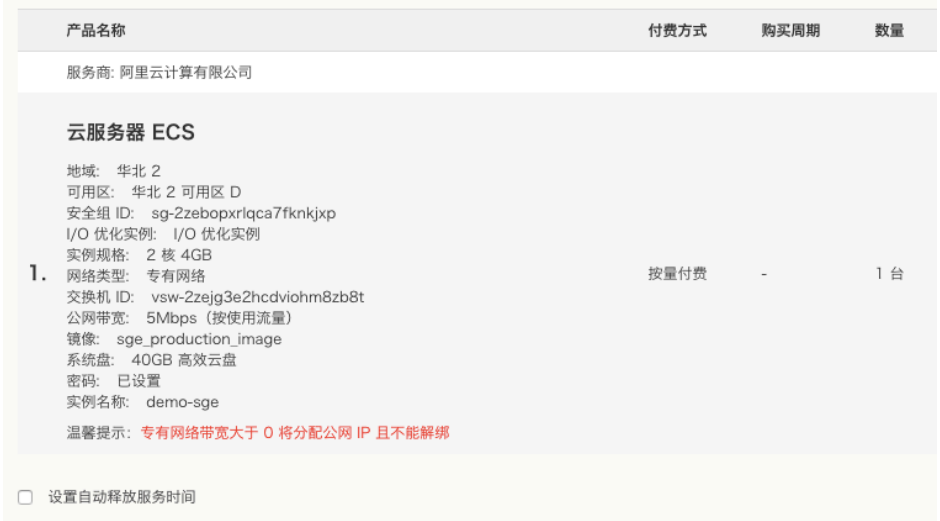
打开 ECS 官方控制台，点击专有网络 VPC 进入 VPC 控制台，然后点击”专有网络”菜单。

创建专有网络。在本示例中，设置专用网络 CIDR 为 192.168.0.0/16，而交换机 CIDR 为 192.168.0.0/24。

- 创建交换机。

B) 购买 ECS VM

- 点击刚才创建的“专有网络”，然后点击“交换机”进入交换机列表，再点击“创建实例”进入创建 ECS 实例页面。
- 公网 IP 地址选择分配。
- 选择安全组，创建专有网络时自动创建了一个安全组，这里只有一个安全组可选。
- 实例规格至少2核4GB。
- 镜像市场: BatchCompute SGE。
- 设置密码。
- 配置参考。
- 请注意创建实例时实例的名称不能修改



3. 启动 SGE 集群

批量计算提供了命令行工具 bccluster(bcc) 来帮助您管理 SGE 集群，该工具预装到 BatchCompute SGE 镜像中。

A) 登录 Master

使用 ssh 命令登录到 Master 节点，务必使用 root 用户。

```
ssh root@<外网IP>
```

然后，输入购买 ECS 时设置的密码.

B) bccluster 命令登录

bccluster（bcc） 工具用来管理 SGE 集群，包括启动、扩容和停止等操作。如果第一次登入 Master 节点，请先更新 bccluster 工具，然后执行以下命令来配置 region, accessKeyId 和 accessKeySecret。其中的 region 必须与 Master 虚拟机所在的 region 相同。

```
pip install -U batchcompute-sge #如果命令执行出错，重试该命令就可以了。
bcc login <region> <accessKeyId> <accessKeySecret>
```

- 该命令只需要第一次登入 Master 节点时执行。
- AccessKey 对应的子账号，要被授予 BatchCompute 全部权限 和 ECS 查询权限，以及 AuthorizeSecurityGroup 和 RevokeSecurityGroup 两个 ECS 写操作 API 的权限。请打开 RAM 控制台 点击“用户管理”菜单，选择相应的子用户进行授权。
- region 参考列表 。
- 执行 bcc login 命令出现如下错误说明 ECS 的名称被修改过，需要删除 ECS 实例重新创建实例然后进行登录操作。

```
fix ip hostname binding...
fixed
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0    22    0    22    0     0   2804      0 --:--:-- --:--:-- --:--:--  5500

  ERROR: Can not found master_security_group_id by i-2zeic9ggky8iuudm5jy9
```

C) start 集群

启动worker节点。

```
bcc start -n 2 -t ecs.sn2.medium -i img-sge --vpc_cidr_block=192.168.1.0/24
```

参数:

- -n 表示启动多少台 worker 节点。
- -t 表示 worker 节点使用哪种实例类型, bcc t命令可以列举可用的实例类型。
- -i 表示 worker 节点使用哪个镜像（可以指定系统镜像 ID：img-sge，或者自定义镜像 ID）。
- —vpc_cidr_block 指定集群网段, 请参考 如何选择网段 。

运行完该命令, 启动指令提交成功, 因为 worker 节点启动有一段时间, 还不能立即使用该集群，需要等待一段时间。

> SGE 集群只能运行在 vpc 网络中，因此必须指定 —vpc_cidr_block；cidr_block必须在创建master ECS实例设置的CIDR范围内，如本例创建master ecs时选的vpc cidr为 192.168.0.0/16，所以cidr_block可选范围在192.168.0.0/16-192.168.0.0/24

D) 查看批量计算集群状态

```
bcc status
```

该命令可以查看集群状态, worker 节点启动情况等。

E) 查看 SGE 集群状态

```
qhost
```

尝试运行qhost命令，看看 SGE 集群是否完全启动。

4. 释放(删除) SGE 集群

如果不再使用 worker 节点，请使用 stop 命令停止所有的 worker 节点。如果 master 节点也不再使用，可以通过控制台删除掉 master 节点。

```
bcc stop
```

> 注意：必须先停止 worker 节点，然后才能释放 master。

5. 如何启动 NAS 挂载的 SGE 集群

使用 bcc 工具可以轻松挂载 NAS，示例如下：

```
bcc start -n 2 -t ecs.sn1.medium -i img-sge --vpc_cidr_block=192.168.1.0/24 -m nas://a/b/c:/home/nas/
```

注意：如何在 VPC 里面创建 NAS 文件系统，请参考 创建文件系统 和 添加挂载点 。

6. 启动多种实例类型的集群（多个group）

运行 bcc start 命令时, 增加 option: —group_num 4 # 表示创建 4 个 group。

```
bcc start -n 2 -t ecs.sn2.medium -i img-sge --vpc_cidr_block=192.168.1.0/24 option: --group_num 4
```

- group 名称分别为: default，group1，group2，group3，并且 group1，group2，group3 的 node 数量都是 0。
- 通过 bcc update 命令批量修改所有 group 的 instanceType 或者只修改某个 group 的 instanceType ；具体参考bcc update --help。
- 通过 bcc resize 命令某一个 group 的 node 数量；具体参考bcc resize --help。

注意： group 个数在 start 后不能变更，如需变更 group 数量，必须 stop 集群后后再重新 start。

7. 如何创建包年包月的 SGE 集群

请按照前面的步骤，启动一个 SGE master 节点，然后登入并初始化 bcc 工具（更新并且 login）；登录到阿里云工单系统提交工单联系运维工程师做包年包月集群的配置处理。

注意：包年包月集群创建后不支持删除，只能等到包月时间点到之后才能释放集群；测试场景建议使用按量使用模式，待准备工作完成后再开通包月集群；bcc start 命令不支持创建包年包月的 SGE 集群。

A）控制台创建包年包月的集群

1. 登入批量计算的控制台，选择您的 master 节点所在的区域，在“集群列表”页面中，点击“创建集群”。
2. 填写必选字段，其中的“镜像 ID”需要选择为“sge(官网提供)”（如果您是自定义的镜像，那么需要选择您自定义的镜像 ID），资源类型选中“包月”。
3. 填写可选字段。
    - Bootstrap：/usr/local/bin/sge_bootstrap，必须为该值。
    - 增加 2 个环境变量。SGE_MASTER_IP_ADDRESS：对应 master 所在的 IP 地址，SGE_MASTER_HOST_NAME：对应 master 所在的 hostname。
    - VpcId：对应 master 所在 VPC。
    - CidrBlock：指定一个 CidrBlock，注意不能与该 VPC 中已有的地址段相冲突。
    - 如果需要挂载 NAS，那么需要增加“Mounts”选项。
4. 点击“提交”，创建集群。
5. 集群创建成功后，进入该集群的页面。
6. 点击“创建预付费实例”，在新的页面中，选择“项目”，“集群”，“实例组”，点击“立即购买

”，再点击“去支付”，“确认支付”。

注意，在上面第 6 步中，一定要确认“项目”，“集群”，“实例组”这三个选项。

B）命令行 attach 该集群

在命令行中，执行如下命令：

```
bcc attach <your_cluster_id>
```

执行成功后，就完成了包年包月的 SGE 集群的创建。

使用 Docker 镜像构建 App

批量计算提供了 App 功能，可以使用虚拟机（VM）镜像来定制运行环境，也可以使用 Docker 镜像，本文将介绍如何使用 Docker 镜像创建 App 和提交 App 作业。

背景

如果您的作业使用了 ISV 提供的软件或算法，可以考虑将其封装在 Docker 镜像中，再使用 App 设置作业的模板（包括资源类型和运行环境），这样一来，提交作业时只需提供输入和输出信息即可。当软件或算法有更新时，只需要更新 Docker 镜像，比如通过 Docker 镜像的 Tag 来标识不同的版本号，修改 App 中 Docker 镜像的版本号即可完成运行环境的更新。

1. 准备 App 的 Docker 镜像

A) 制作 Docker 镜像

根据自己的需求，用户可以使用官方镜像仓库中的镜像作为基础镜像，安装需要的软件或算法，制作成 Docker 镜像，完成运行环境的定制；制作镜像有两种方法：

- 使用 Dockfile 制作镜像
- 使用容器快速制作镜像

具体制作方法可参考用户指南中的 Docker 镜像制作。

建议：在制作 Docker 镜像时，最好带上 Tag，后续版本有更新时，只需要更新 Tag 即可。

B) 本地调试Docker镜像

Docker 镜像制作完成以后，可以参考用户指南中的 Docker 本地调试相关章节进行本地调试，确保 Docker 镜像在 BatchCompute 的环境下可以正常使用。

C) 推送到镜像仓库

可以将制作好的 Docker 镜像推送到 OSS 的镜像仓库。具体方法请参考用户指南中 Docker镜像上传到 OSS 的详细描述 。

2. 创建 App

BatchCompute提供了 API、SDK、控制台等三种方式创建 App，下面以控制台和 Python SDK 为例，分别介绍如何使用 Docker 镜像创建 App。

A) 使用控制台创建 App

假如 Docker 镜像被推送到 OSS 镜像仓库的路径为oss://demo-bucket/dockers/，镜像名称为localhost:5000/demodockerimage:0.1。

如上图所示，在创建 App 时，选择镜像类型为 Docker，填写 Docker 镜像的名称，以及 OSS Registry 的路径。关于控制台如何创建 App 的其他参数详情，请参考用户指南中创建 App 的描述，这里不再赘述。

B) 使用 SDK 创建 App

使用 Python SDK 创建 App 时，参考如下的形式：

```
#encoding=utf-8
import sys
from batchcompute import Client, ClientError
from batchcompute import CN_BEIJING as REGION
from batchcompute.resources import (
JobDescription, TaskDescription, DAG, AutoCluster, GroupDescription, ClusterDescription, AppDescription
)
ACCESS_KEY_ID='xxxx' # 填写您的 ACCESS_KEY_ID
ACCESS_KEY_SECRET='xxxx' # 填写您的 ACCESS_KEY_SECRET
def main():
try:
client = Client(REGION, ACCESS_KEY_ID, ACCESS_KEY_SECRET)
app_desc = {
"Name":"Docker-app-demo",
"Daemonize":False,
```

```
"Docker":{
"Image":"localhost:5000/demodockerimage:0.1",
"RegistryOSSPath":"oss://demo-bucket/dockers/"
},
"CommandLine":"python test.py",

#其他参数这里不详细展示
}
appName = client.create_app(app_desc).Name
print('App created: %s' % appName)
except ClientError, e:
print (e.get_status_code(), e.get_code(), e.get_requestid(), e.get_msg())
if __name__ == '__main__':
sys.exit(main())
```

如上面的实例代码所示，在AppDescription中填写 Docker 信息的Image和RegistryOSSPath。其他参数请参考用户指南中的创建示例。

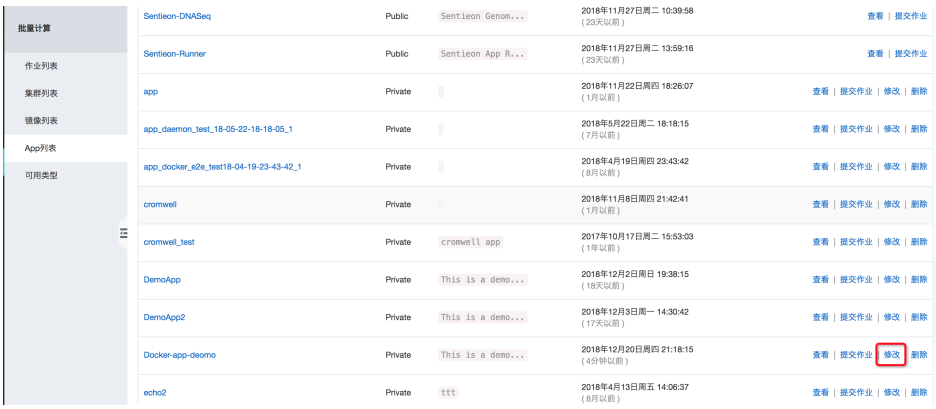
3. 提交 App 作业

提交作业时，不再涉及 Docker 相关的信息，具体方法请参考用户指南中提交 App 作业的描述。

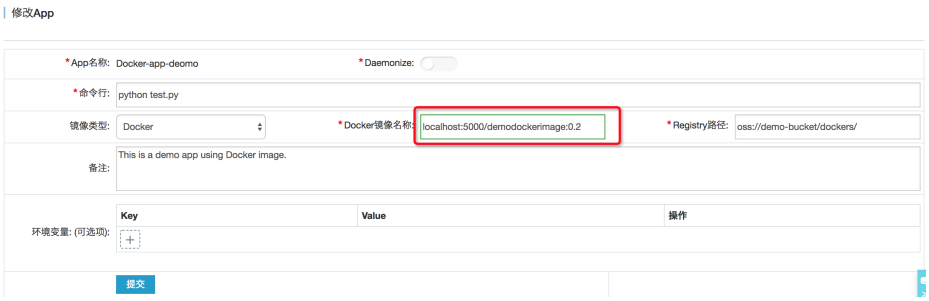
4. Docker 镜像更新

假如 App 中使用的 ISV 提供的软件或算法有更新，您只需要更新 Docker 镜像，并用 Tag 标识版本。然后更新 App 信息中的 Docker 镜像名称就可以。

A) 使用控制台更新



如上图所示，在 App 列表中找到需要更新的 App，点击**修改**按钮进入 App 的修改页面。



如上图所示，在修改页面，修改 App 的 **Docker 镜像名称**后，点击**提交**即可完成 App 的更新。

B) 使用 SDK 更新

使用 Python SDK 来更新 App 的 Docker 信息可参考如下示例：

```
#encoding=utf-8
import sys
from batchcompute import Client, ClientError
from batchcompute import CN_BEIJING as REGION
from batchcompute.resources import (
JobDescription, TaskDescription, DAG, AutoCluster, GroupDescription, ClusterDescription, AppDescription
)
ACCESS_KEY_ID='xxxx' # 填写您的 ACCESS_KEY_ID
ACCESS_KEY_SECRET='xxxx' # 填写您的 ACCESS_KEY_SECRET
def main():
try:
client = Client(REGION, ACCESS_KEY_ID, ACCESS_KEY_SECRET)
app_desc = {
"Name":"Docker-app-demo",
"Daemonize":False,
"Docker":{
"Image":"localhost:5000/demodockerimage:0.2",
"RegistryOSSPath":"oss://demo-bucket/dockers/"
},
"CommandLine":"python test.py",
"EnvVars": {}
}

res = client.modify_app("Docker-app-demo", app_desc)
print res

except ClientError, e:
print (e.get_status_code(), e.get_code(), e.get_requestid(), e.get_msg())

if __name__ == '__main__':
sys.exit(main())
```

对于简单的修改 Docker 版本号的情况，推荐使用控制台，操作更简单。

云渲染管理系统

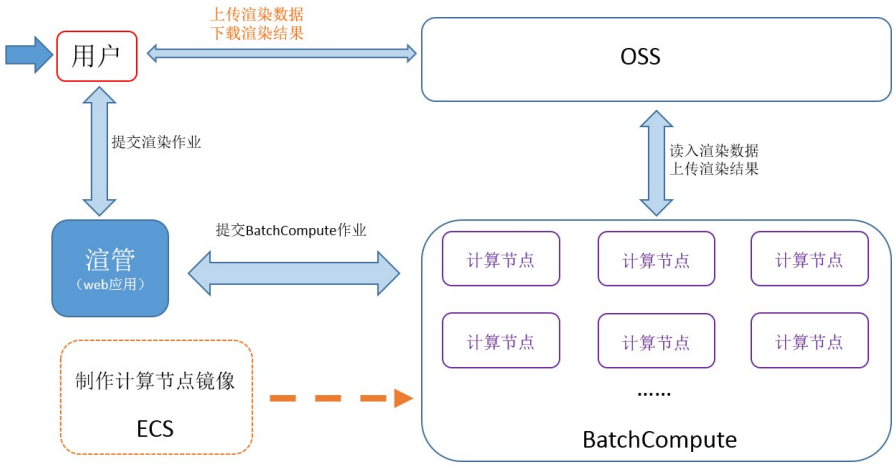
简介

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。详细介绍请参考官网，在使用渲管前，请确保已开通此三产品。

- 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



1. 使用流程

A) 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，在云服务器中安装所需的渲染软件；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。

B) 上传数据到OSS

将渲染所需要的数据上传到对应区域的OSS，并保持上传前的目录结构。

C) 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）。

D) 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。

E) 创建项目

在渲管的项目管理页面创建项目，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），计算节点中的临时输出路径。

F) 集群的创建和管理

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

G) 提交渲染作业

在项目页里提交渲染作业，要指定目的集群、渲染的帧范围以及节点数量等信息。提交完作业后，可实时查看渲染日志以及节点 CPU 使用率等信息。

> 使用 AutoCluter 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。

快速开始

BatchCompute 提供了测试用的计算节点镜像（windows server 2008，ID：m-

wz9du0xaa1pag4ylwzsu），它预装了 blender 渲染软件。使用 blender 制作一个小场景的 演示视频 已上传 OSS（测试时，需下载并上传到您的 OSS bucket）。

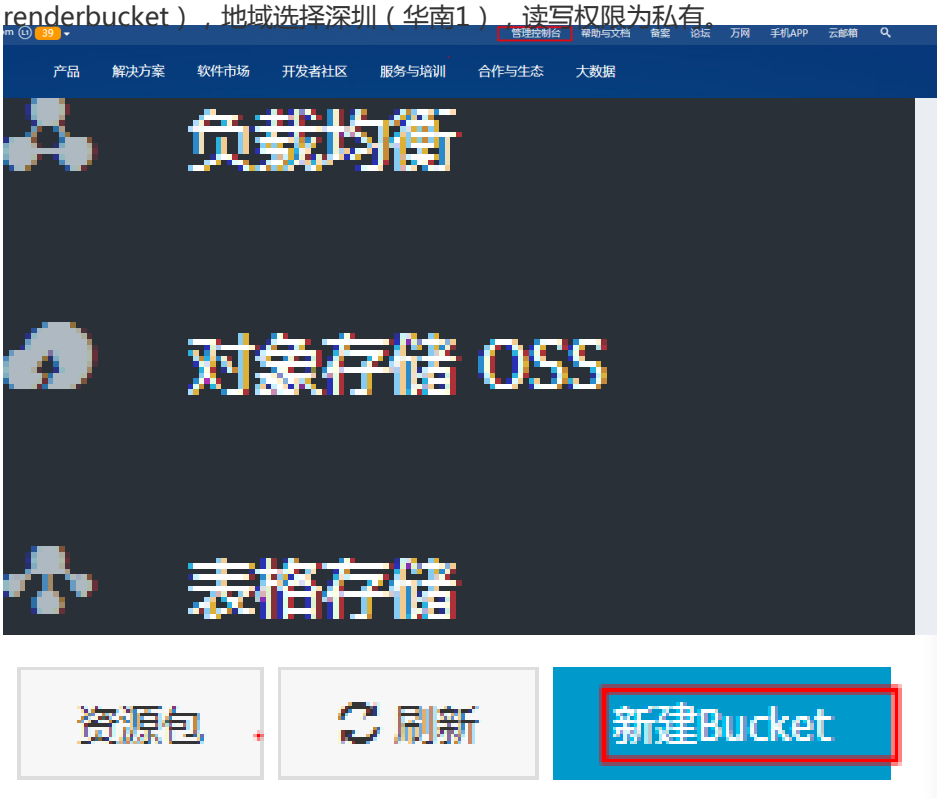
实际生产时，请根据需求制作合适的计算节点镜像。

1. 准备工作

- 注册阿里云账号并开通 OSS、ECS 和 BatchCompute 服务。
- 创建AccessKey。账号信息->AccessKeys->创建 Access Key，记录 Access Key 信息。

2. 渲染示例

A) 创建 OSS bucket阿里云官网->管理控制台->对象存储 OSS->创建 bucket（例如，名字为 renderbucket），地域选择深圳（华南1），读写权限为私有。



3. 获取blender场景并上传到您的 OSS bucket

- 在浏览器输入 http://openrm.oss-cn-qingdao.aliyuncs.com/blender/monkey/cube.blend 。
- 下载示例场景文件（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

A) 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，然后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。


包年包月
按量付费
购买云盘
按量付费服务不支持备案。
基本配置
地域：
华北 1
华北 2
华东 1
华东 2
华南 1
香港
亚太东南 1 (新加坡)
美国东部 1 (弗吉尼亚)
亚太东北 1 (东京)
不同地域之间的产品内网不互通；订购后不支持更换地域，请谨慎选择 教我选择>> 查看我的产品地域
可用区：
随机分配
查看实例分布详情>>
网络
网络类型：
经典网络
专有网络
教我选择>>
经典网络与专有网络不能互通，购买后不能更换网络类型，请谨慎选择
安全组名称all_allown / sg-9498ck2b2 (已有8个实例, 还可以加入992个实例)
请确保此安全组开放包含 22 (Linux) 或者 3389 (Windows) 端口，否则无法远程登录ECS。您可以进入 ECS控制台 设置。
重新选择安全组
安全组类似防火墙功能，用于设置网络访问控制，您也可以到管理控制台 创建新安全组>>> 教我选择>>
实例
实例系列：
系列 I
系列 II
系列 I 采用 Intel Xeon CPU，DDR3 的内存。
I/O 优化：
I/O 优化实例
实例规格：2 核 2GB (标准型 s2 , ecs.s2.small)
请选择实例规格
带宽
公网带宽：
按使用流量
按固定带宽
带宽峰值：
25M
50M
100M
100Mbps
系统会分配公网 IP (不能解绑)，若不需要分配公网 IP，请选择按固定带宽计费，带宽值 0M。
阿里云免费提供最高 5Gbps 的恶意流量攻击防护，了解更多>> 提升防护能力>>
镜像
镜像类型：
公共镜像
自定义镜像
共享镜像
镜像市场
公共镜像即基础操作系统。镜像市场在基础操作系统上，集成了运行环境和各类软件。
镜像名称：RenderManager 0.5.3
重新选择镜像
系统盘：
普通...
40 GB
200～500 IOPS
系统盘设备名：/dev/xvda
如何选择 SSD云盘 / 高效云盘 / 普通云盘，请看 详细说明>>
镜像市场[华南 1]
如需选购镜像包月套餐，请点击镜像名称购买，访问 云市场 发现更多软件和优惠！
全部
运行环境
管理与监控
建站系统
应用开发
数据库
服务器软件
企业软件
云安全市场
已购买的镜像
已订阅的镜像
rendermanager
筛选
RenderManager
来源: 阿里云计算有限公司
云渲染管理系统 (Render Manager简称渲管) 是一个开源的we...
0.5.3
¥0.00 /时
同意并使用
同意《镜像使用协议》
RenderManager
来源: 阿里云计算有限公司
集成软件：安装了云渲染管理系统，轻松搭建私人云上渲染农场。
V1.2
¥0.00 /时
同意并使用
同意《镜像使用协议》
上一页
1
下一页
从镜像市场选择 (含操作系统)

B) 购买后，点击进入管理控制台，在实例列表中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

在本地浏览器输入 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：

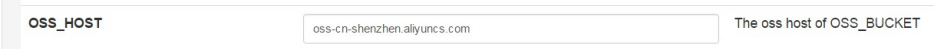
- rm_admin
- rm_admin@123

生产系统，请一定更改账号和密码。

6. 配置渲管

A) 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_BUCEKET 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。

B) 设置 OSS_HOST 为 oss-cn-shenzhen.aliyuncs.com；REGION 的选择主要和计算节点的镜像归属有关，必须和计算节点镜像归属 REGION 保持一致；本例采用的官方计算节点镜像（该镜像部署在深圳 REGION）所以此处设置在深圳 REGION 。

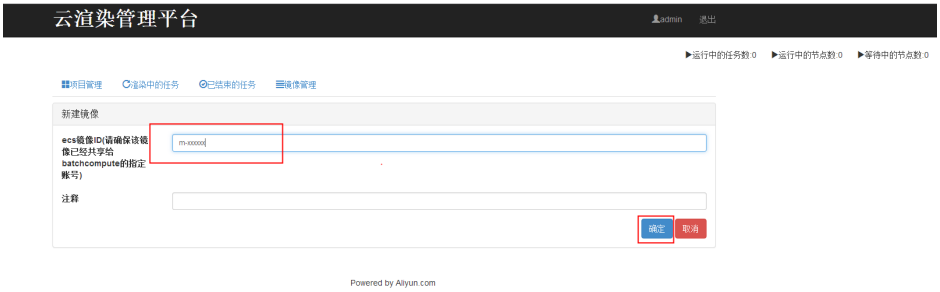


C) 设置 BATCHCOMPUTE_REGION 为 cn-shenzhen；设置深圳 REGION 原因同上。

D) 点击保存。

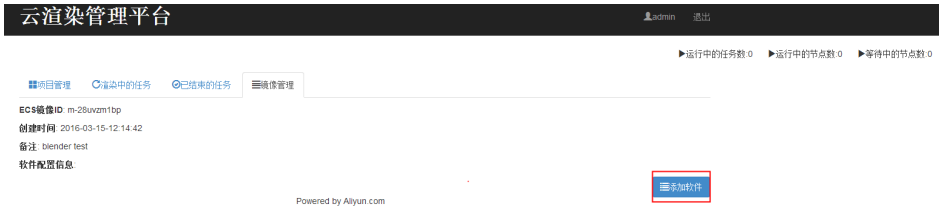
7. 添加计算节点镜像

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，需要用户制作所需要的计算节点镜像，具体制作流程请参考 操作手册）。



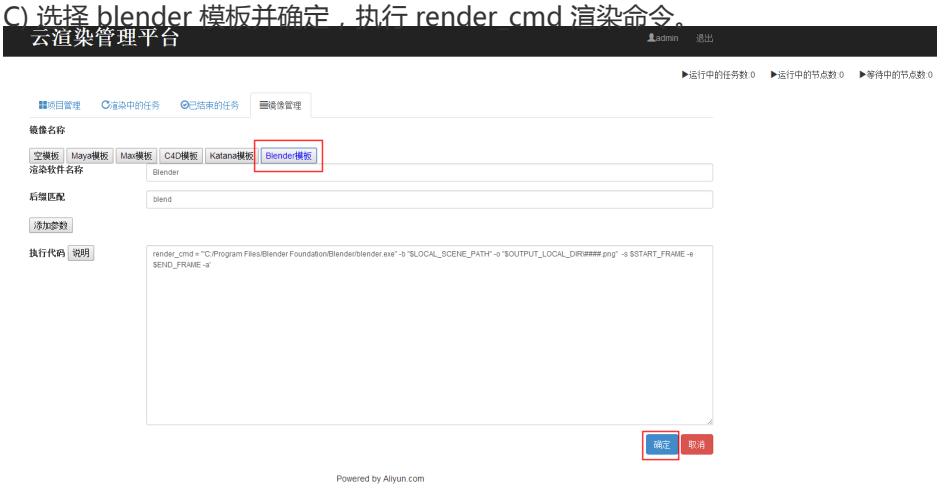
8. 配置渲染软件信息

A) 镜像管理->软件配置。



B) 添加软件。

C) 选择 blender 模板并确定，执行 render_cmd 渲染命令。

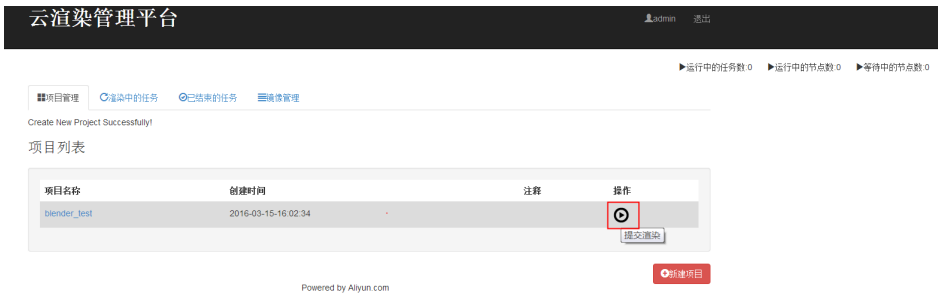


9. 创建项目

A) 项目管理->新建项目。 B) 填入需要映射的 OSS 路径数量(本例只映射一个OSS路径)，并点击确认。

C) 填入项目名称: blender_test。D) 镜像选择上面创建的镜像。E) OSS 映射中的选择/输入路径为 /renderbucket/blender/。F) OSS 映射的目的地为盘符 G: (本例中使用的镜像系统为 Windows2008 server)。G) OSS 输出目录填写为 /renderbucket/rm_test/output/。H) 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。I) 确定提交。

10. 提交渲染任务



A) 项目管理->提交渲染。

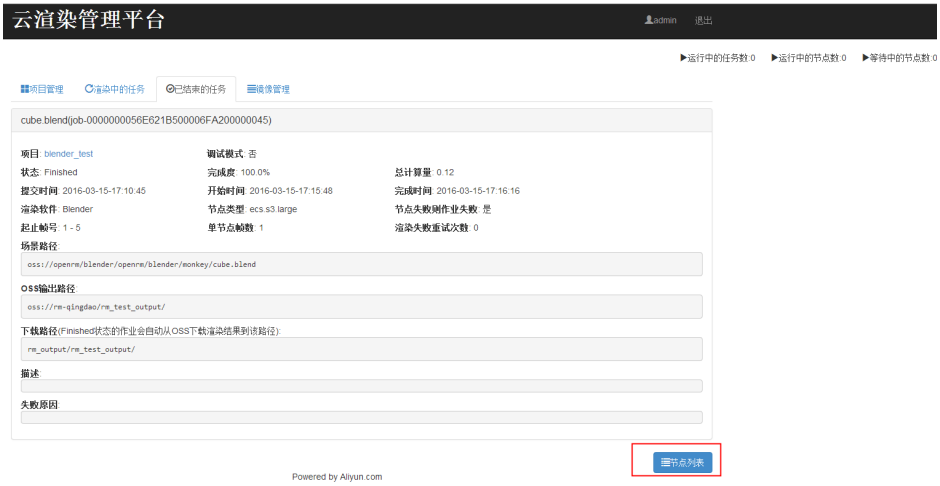
B) 选择场景所在的 OSS 路径前缀。

C) 选择项目根目录， 直到场景文件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。

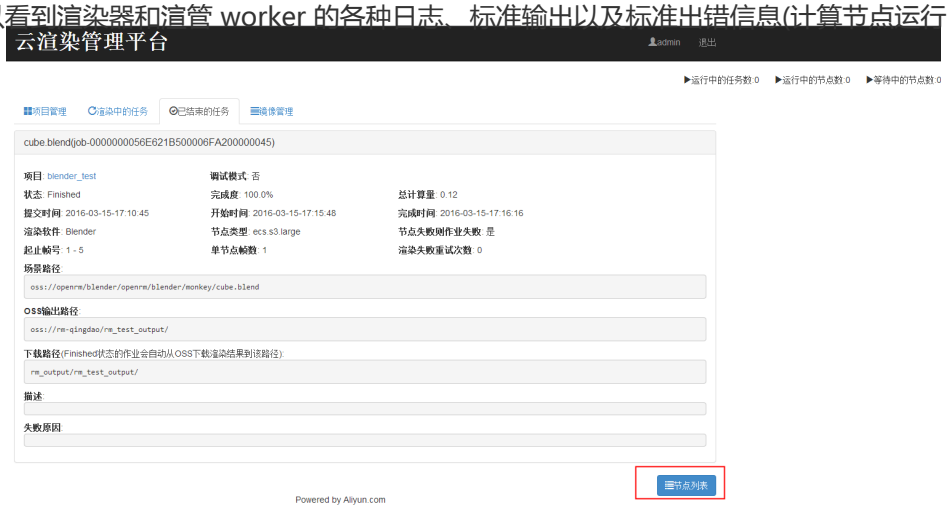
D) 选择渲染中的任务，可查看刚才提交的作业。

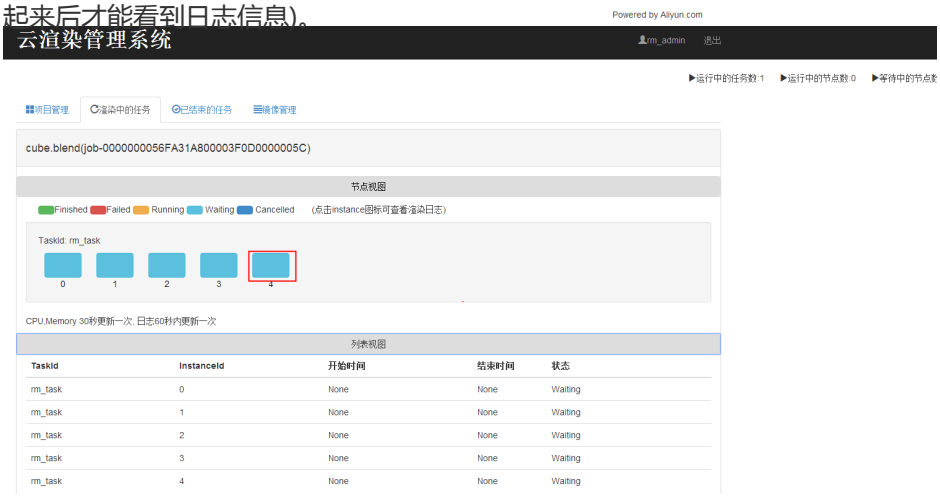
11. 查看渲染日志

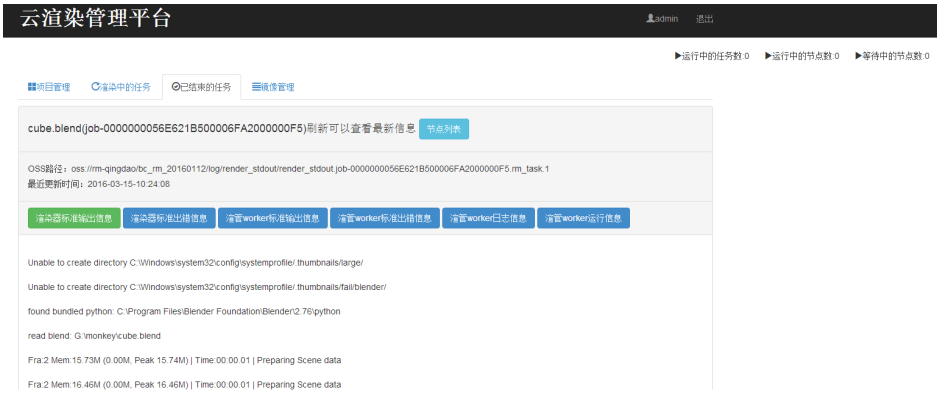
A) 点击任务名称并点击节点列表。



B) 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)。







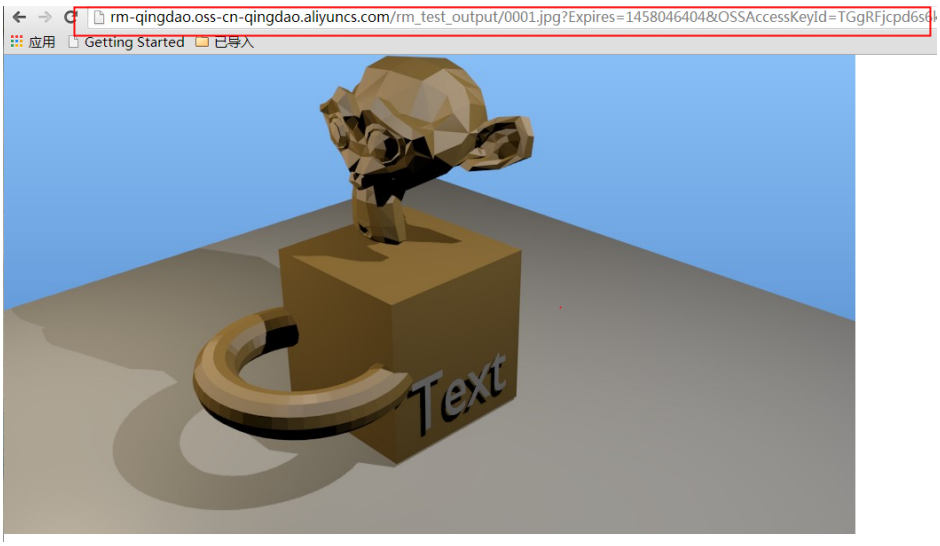
12. 查看渲染结果

A) 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。

B) 点击任务名称，可以查看 OSS 上的输出路径。

C) 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。

D) 在浏览器粘贴 URL 可以直接查看图片。



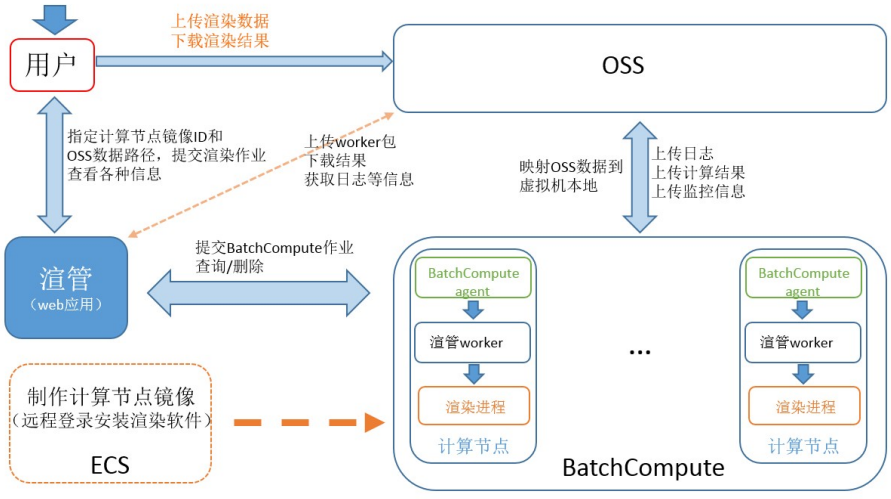
E) 恭喜您已跑通云上的 Blender 渲染测试。

操作手册

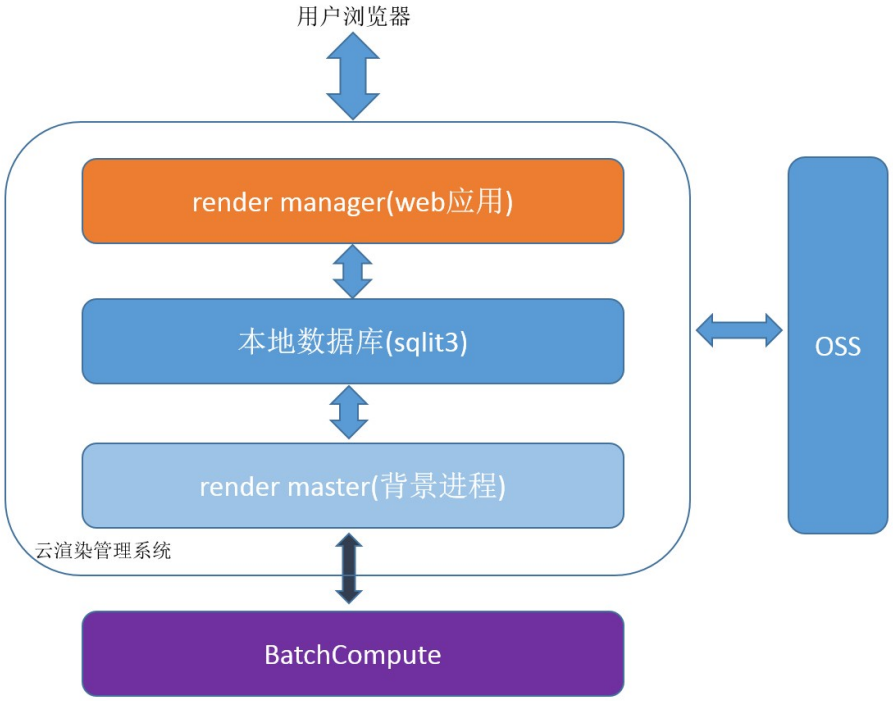
1. 渲管系统结构

A) 渲管与各云产品的详细关系

渲管与各云产品的依赖如下图所示。



B) 渲管系统内部结构



渲管系统由如下 3 部分组成：

- render manager: 基于 flask 框架开发web 应用，主要负责和用户进行人机交互，接收用户请求。
- render master：后台背景进程，根据人机交互的结果进行作业提交以调度。
- 本地数据库：主要存放用户提s交的渲管请求，待渲管任务结束后自动删除该信息。

2. 渲管的部署

在阿里云云市场有已安装了渲管的 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。

A) 获取渲管镜像

官方渲管镜像：RenderManager 镜像，创建 ECS 实例时，选择镜像市场，直接搜索以上关键字即可获取。自定义渲管镜像：基础镜像建议采用 Ubuntu 14.04 64 位，按照以下步骤安装渲管系统。

```
# 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 localchost;
# 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
```

B) 开通 ECS 实例

请指定某 ECS 实例部署渲管系统，配置参数，请参考创建 Linux 实例

- 公网 IP 地址选择分配。
- 镜像市场: RenderManager 或者自定义镜像
- 设置密码

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，必须设置SECURITY_ID，SECURITY_KEY，OSS_BUCKET 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。 区域的选择和计算节点的镜像区域保持一致，若计算节点镜像在深圳区域，则渲管的区域信息也必须是深圳，同时 OSS BUCKET 也必须是该 REGION 下的 BUCKET；若使用批量计算官方提供的计算节点镜像则需要选择深圳 REGION。

其他配置项，请参考页面上的说明。

5. OSS数据上传

提交渲染作业前，一定要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 。

参考 更多 OSS工具 。

6. 计算节点镜像制作

渲染客户如希望定制计算节点镜像，请参考：自定义镜像。

7. 计算节点镜像管理

A) 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。

B) 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）会在渲管 worker 中执行，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。

渲管已经预定义了一些变量和函数，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

A) 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。

i. 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。

ii. 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。

iii. 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。

B) 提交渲染任务

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。

C) 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。

常见问题

Q：我有大量渲染作业，但是波峰波谷明显，有什么好建议？

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lisence 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lisence server 的渲染软件，可以直接将 lisence server 做在镜像里，这样每个计算节点都会有一个 lisence 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？

A： 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，请记得贡献回社区哦。