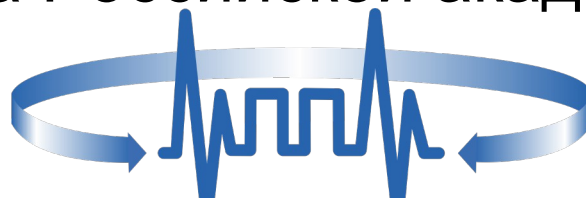


Применение контейнеризации для поддержки SvF-технологии на платформе Everest.

Сергей Смирнов
Владимир Волошинов

Центр распределённых вычислений
Института проблем передачи информации им. А.А.
Харкевича Российской академии наук



Everest – PaaS для создания REST-сервисов из приложений командной строки

Для пользователя:

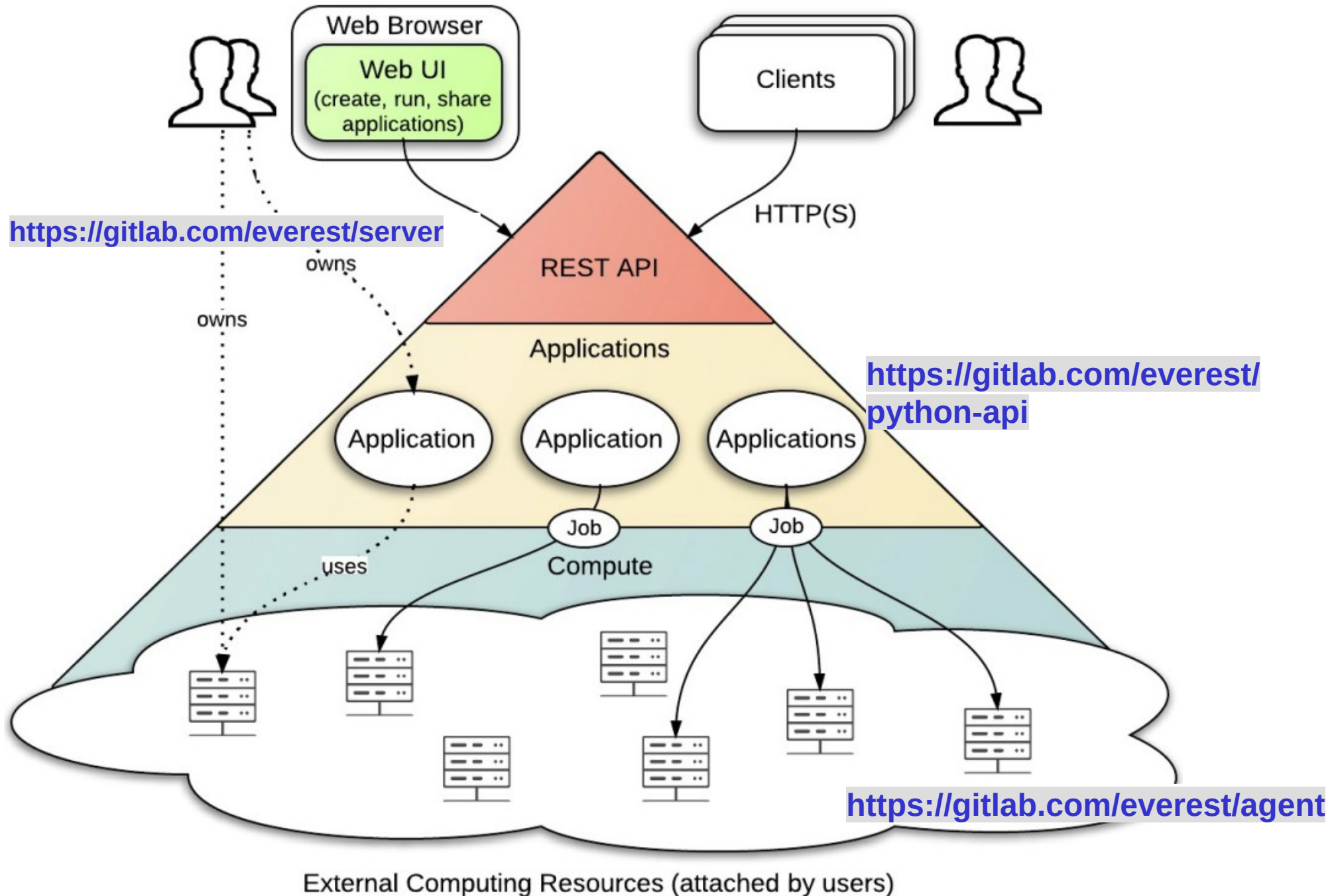
- Использование приложения через привычную Веб-форму
- Отслеживание статуса расчета
- Проведение множества расчетов одновременно

Для разработчика:

- Простая публикация приложения: настроить параметры приложения через Веб-интерфейс и запустить агента (можно на собственном ноутбуке)
- Готовый API
- Готовая авторизация пользователей

Платформа Everest, everest.distcomp.org

Позволяет быстро создавать REST-сервисы для проведения расчетов



1. Подготовка окружения:

- Установить компиляторы, интерпретаторы
- Собрать солверы и другие пакеты
- Установить агент Эвереста (скачать, создать токен, отредактировать конфиг, настроить автозагрузку)

2. Варианты получения кода приложения:

- Вручную скопировать в заранее известную папку
- Скачивать при каждом запуске задания
- Загрузить в Эверест как один из файлов приложения

Окружение меняется редко, настраивается долго

Приложение меняется часто, копируется быстро

Контейнеризация – это метод виртуализации, при котором приложение и все его зависимости упаковываются в изолированный контейнер. Основные характеристики:

- Контейнер включает все необходимое для работы приложения (библиотеки, конфигурации, файлы)
- Каждый контейнер изолирован от операционной системы и других контейнеров
- Контейнер можно легко переносить между разными системами
- Быстрый запуск - секунды вместо минут (в отличие от виртуальных машин)
- Эффективное использование ресурсов - контейнеры используют общее ядро операционной системы

SvF (Simplicity vs Fitting) – технология для поддержки построения математических моделей (структурных, не только регрессионных) на основе экспериментальных данных.

Вычислительная схема SvF основана на решении двухуровневых оптимизационных задач специального типа. На нижнем уровне необходимо решить набор независимых оптимизационных задач, аналогичных обратным задачам с регуляризацией Тихонова.

Зависимости:

- Python 3.7+ (Pyomo, Everest Python API, ...)
- Солвер Ipopt

Задания и подзадания SvF-расчета в Everest

jobs > svf-remote-3-Oscill_K_mu

Job Info Share Tasks Children

done

Task-0

shark1vvv4SvF 2m 27s Wait 0ms Run 2m 25s 0.63 KB 115.50 KB

jobs > svf-remote-3-Oscill_K_mu

Job Info Share Tasks Children

Name	Application	State	Submitted
NoName0-ipopt-1	solve-set-opt-probs	DONE	18 Nov 2024 00:18:1
NoName1-ipopt-1	solve-set-opt-probs	DONE	18 Nov 2024 00:18:3
NoName1-ipopt-1	solve-set-opt-probs	DONE	18 Nov 2024 00:18:5
NoName1-ipopt-1	solve-set-opt-probs	DONE	18 Nov 2024 00:19:1
NoName1-ipopt-1	solve-set-opt-probs	DONE	18 Nov 2024 00:19:3
NoName2-ipopt-1	solve-set-opt-probs	DONE	18 Nov 2024 00:19:5
NoName3-ipopt-1	solve-set-opt-probs	DONE	18 Nov 2024 00:20:1

jobs > NoName3-ipopt-1

Job Info Share Tasks

done Task-0
shark1vvv 6s Wait 0ms Run 5s 42.88 KB 6.16 KB

done Task-1
shark1vvv 6s Wait 0ms Run 5s 42.88 KB 6.18 KB

done Task-2
shark1vvv 6s Wait 0ms Run 5s 42.88 KB 6.16 KB

done Task-3
shark1vvv 6s Wait 0ms Run 5s 42.88 KB 6.12 KB

done Task-4
shark1vvv 6s Wait 0ms Run 5s 42.88 KB 6.18 KB

done Task-5
shark1vvv 6s Wait 0ms Run 5s 42.88 KB 6.15 KB

done Task-6
shark1vvv 6s Wait 0ms Run 5s 42.88 KB 6.15 KB

done Task-7
shark1vvv 6s Wait 0ms Run 5s 42.88 KB 6.16 KB

done Task-8
shark1vvv 6s Wait 0ms Run 5s 42.88 KB 6.12 KB

done Task-9
shark1vvv 6s Wait 0ms Run 5s 42.88 KB 6.17 KB

shark1vvv 6s Wait 0ms Run 5s 42.88 KB 6.17 KB

shark1vvv 6s Wait 0ms Run 5s 42.88 KB 6.17 KB

Пример Dockerfile

Берет солверы из <https://hub.docker.com/r/distcomp/solvers>

Агента берет из https://gitlab.com/everest/agent/container_registry/6234884

```
FROM distcomp/solvers:latest as solvers
```

```
FROM registry.gitlab.com/everest/agent:master-3.8-bullseye as agent
```

```
COPY --from=solvers /usr/local/ /usr/local/
```

```
COPY --from=solvers /usr/lib/ /usr/lib/
```

```
COPY --from=solvers /etc/alternatives/ /etc/alternatives/
```

```
COPY docker/requirements.txt /tmp/requirements.txt
```

```
RUN pip install --no-cache-dir -r /tmp/requirements.txt
```

```
RUN apt-get install zip
```

```
COPY docker/bin/ /usr/local/bin/
```

```
COPY . /home/vvv/git_work/SvF/
```

```
COPY svf-service/runSvF-remote.sh /usr/local/bin/
```


Makefile

```
IMAGE_NAME=distcomp/svf  
EVEREST_TOKEN?=TOKEN
```

build:

```
docker build -t $(IMAGE_NAME) -f Dockerfile ..
```

push:

```
docker push $(IMAGE_NAME)
```

run:

```
docker run -d -it --rm --name agent -e EVEREST_TOKEN=$(EVEREST_TOKEN) -e MAX_TASKS=16 $(IMAGE_NAME)
```

stop:

```
docker stop agent
```

Пример запуска

```
[sasmir@shark1:~]$ cd SvF/docker/
[sasmir@shark1:~/SvF/docker]$ make stop build run
docker stop agent
agent
docker build -t distcomp/svf -f Dockerfile ..
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
             Install the buildx component to build images with BuildKit:
             https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 44.64MB
Step 1/11 : FROM distcomp/solvers:latest as solvers
----> bf5bd68d831e
Step 2/11 : FROM registry.gitlab.com/everest/agent:master-3.8-bullseye as agent
----> 4adcebc33b78
...
Step 9/11 : COPY docker/bin/ /usr/local/bin/
----> Using cache
----> a26aae7bd69f
Step 10/11 : COPY . /home/vvv/git_work/SvF/
----> f731d9a9181b
Step 11/11 : COPY svf-service/runSvF-remote.sh /usr/local/bin/
----> 85bdb7a15857
Successfully built 85bdb7a15857
Successfully tagged distcomp/svf:latest
docker run -d -it --rm --name agent -e EVEREST_TOKEN=
fe3b777bc4330374f696ae683606f68b1fb202d355da147d77a687e4d9621800 -e MAX_TASKS=16 distcomp/svf
```

- Упрощается настройка окружения
- Возможна установка агента вместе с приложением в одну строку
- Возможно использовать образ не только для запуска агента, но и чтобы просто применять SvF на собственной машине

Спасибо за внимание!

Сергей Смирнов,
ИППИ им. Харкевича РАН
sasmir@gmail.com