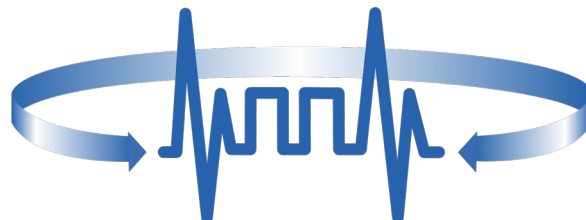


A comprehensive approach to running optimization workflows in the Everest platform

Sergey Smirnov

Center for Distributed Computing,
Institute for Information Transmission Problems
of the Russian Academy of Sciences (Kharkevich Institute)



Optimization subsystem:

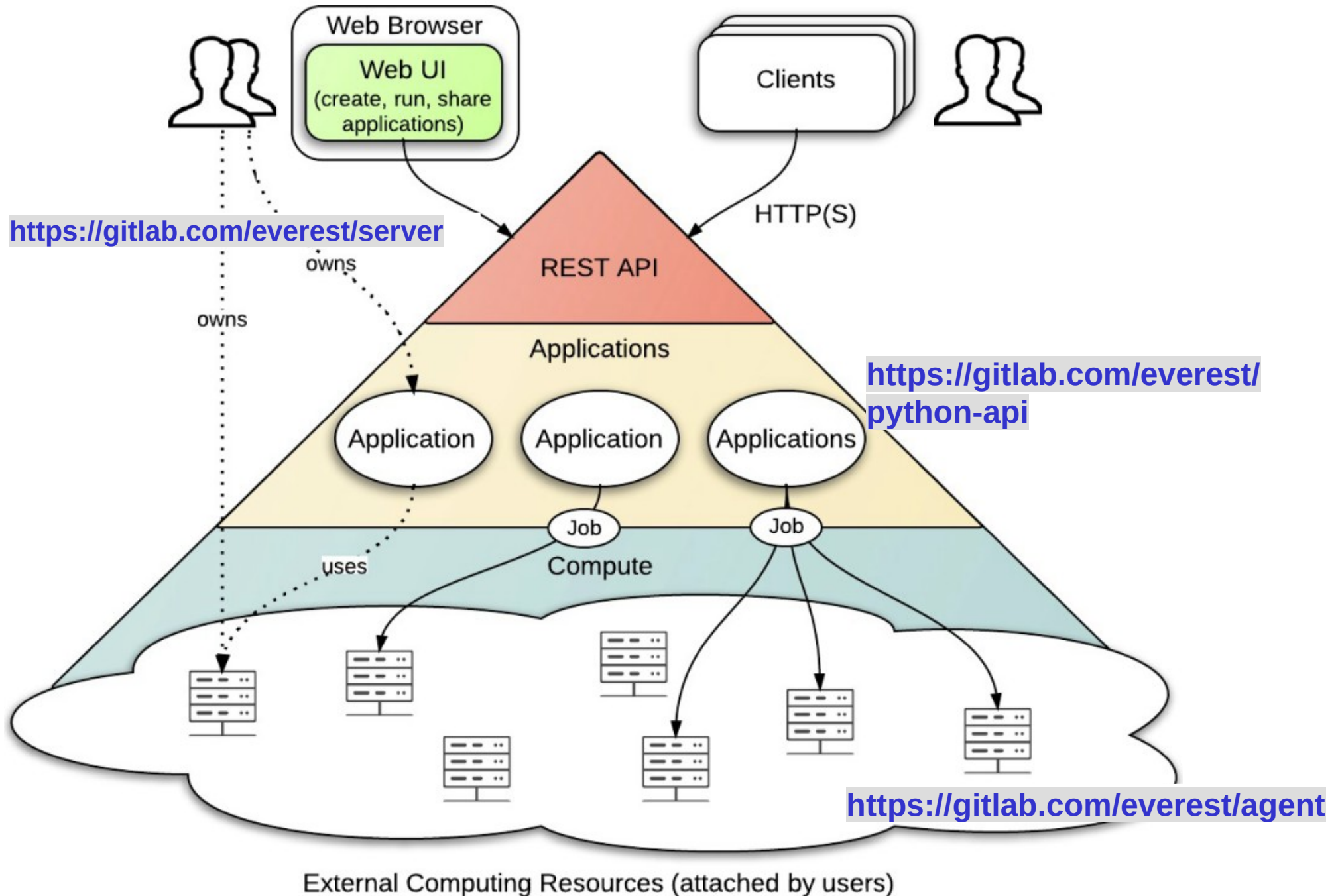
- Gets input data
- Checks data consistency
- Prepares data for further optimization steps
- Performs one or more optimization steps (build and solve a model with a solver)
- Saves results for further processing

It is good to be able to vary:

- Settings
- Version
- Data

Everest Web-based Platform, everest.distcomp.org

Describe/Develop/Deploy REST-services representing existing applications



Everest Job Inputs

Inputs:

- Model's version
- JSON parameters (editable text field)
- Input data as archive
- Computing resource can be selected

Resources

GCE_4_15_PVM_GUROBI ⓘ

GCE_8_64_DONT_GUROBI

GCE_8_32_DONT_GUROBI

GCE_8_30_PVM_CPLEX

GCE_4_26_DONT_GUROBI

GCE_4_16_AMD_DONT_GUROBI

✓ GCE_4_15_PVM_GUROBI

GCE_4_15_DONT_GUROBI

About

Parameters

Submit Job

Discussion

+ Create preset

Job Name

RRTEST-5059___Crew rostering (over

Версия
алгоритма

RR-2261

Настройки
алгоритма

```
{
  "month_sfhb_start": "2023-06-01",
  "fact_start": "2023-06-01",
  "fact_start_date": "2023-06-01",
  "period_start_date": "2023-06-28",
  "allow_change_published_plan": false,
  "committed_plan_start_date": "2023-07-06",
  "free_plan_start_date": "2023-07-31",
  "period_end_date": "2023-07-12"
}
```

Файл с детальными настройками алгоритма

Zip file with data
files

Upload

2023-07-03T18_16_32_Def... 

Zip-архив со входными данными (остальные параметры эвереста имеют над ним приоритет)

Everest Job Outputs

Outputs:

1. Results as archive
2. stderr (summary)
3. stdout viewed in one click
4. Archive with files for debugging purposes (controlled by parameter)
5. Model's version
6. Last status message
7. CPU and memory statistics collected by a simple script

Resubmit button is very handy

The screenshot shows the Everest Job Outputs interface for job `RRTEST-5059__Crew rostering (over generated pairings)_2023-07-03T18_16_32_Default`. At the top, there are buttons for `Delete`, `+ Create preset`, and `Resubmit`. Below the job name, there are tabs for `Job Info`, `Share`, and `Tasks`. The `Job Info` tab is active, showing details: Submitted by `ssmir`, Application `Rosterize core (version 0.0.2)`, Runtime `04 Jul 2023 13:42:30 - 04 Jul 2023 14:15:26`, Data Size `6.50 MB`, and a `View log` link. A green `DONE` status is shown in the top right. The main content area is divided into `Inputs` and `Outputs` tabs. The `Outputs` tab is active, showing a list of output files with their names and download links. The outputs are: `Результаты rosterинга` (results.xlsx), `host-stats` (host-stats.txt), `stderr` (a large text block containing a summary of the job's execution, including repository information, deployment details, and various statistics), `version` (a text block showing the last status message), `Стандартный вывод` (stdout), `Логи солвера` (solver_logs.tbz), and `Данные для генерации отчетов` (results_raw.tbz2). Each output file has a download link and a red number indicating its position in the list: 1 for results_raw.tbz2, 2 for stderr, 3 for stdout, 4 for solver_logs.tbz, 5 for version, 6 for the last status message, and 7 for host-stats.txt.

jobs > RRTEST-5059__Crew rostering (over generated pairings)_2023-07-03T18_16_32_Default

`Delete` `+ Create preset` `Resubmit`

Job Info Share Tasks

Submitted by `ssmir` Application `Rosterize core (version 0.0.2)` `DONE`

Runtime `04 Jul 2023 13:42:30 - 04 Jul 2023 14:15:26` Data Size `6.50 MB` `View log`

Info [5] Rostering 2023-06-27_2023-07-12: Solution 103_939_100, gap 0.01%/0%, absgap 9_100/29_375 6

Inputs Outputs

Результаты rosterинга `results.xlsx`

host-stats `host-stats.txt` 7

stderr

```
Downloading rosterize core
Running core powered by gurobi:latest
REPOSITORY TAG DIGEST
deploy.rosterize.cloud:5050/gurobi latest sha256:5030ffd63cd77e150d3aet
fixed_crews_skipped 0/0
unknown_pairing_assignments 0/0
magic_pilot_assignments 0/0
fatigue_decreases 23/0
bound_pilot_skipped_total 0/0
risks.reserve_risks 249840/0
objective 103926121/23160391
pairing_terminate_fatigue 0/0
critical/error/warning 0/697/1312
```

 2

version

2 minutes ago: RR-2261 Add SKT to noplan (RR-2261, 80115e04) by Sergey Smir 5

Стандартный вывод `stdout` 3

Логи солвера `solver_logs.tbz` 4

Данные для генерации отчетов `results_raw.tbz2` 1

Everest Application Settings

Metadata Description **Parameters** Configuration Files Resources

Access Versions

Inputs + Add input

version string

profile string
Файл с детальными настройкам...

zip-file string
Zip-архив со входными данным...

Outputs + Add output

version string
Rosterize core version

results_raw_tbz2 string
Данные для генерации отчетов ...

host-stats string
Host cpu and memory load stats

solver_logs string
Архив с логами солвера

stdout string

stderr string

Metadata Description Parameters **Configuration** Files Resources Access Versions

Command

/bin/bash bootstrap.sh "\${version}" "" "" 1 1 1 false 100 false 1 RR 0 "" 1 false "" "" "" "" "\${version}"

Refer to input values as \${param}

Input Mappings + Add input mapping

profile profile.json

pattern

zip-file data.zip

pattern

Output Mappings + Add output mapping

stderr stderr

stdout stdout

solver_logs solver_logs.tbz

host-stats host-stats.txt

version rosterize-version.txt

results_raw_tbz2 results_raw.tbz2

Metadata Description Parameters **Configuration** Files Resources

Upload

bootstrap.sh
1.25 KB

Public
Stage in

Metadata Description Parameters **Configuration** Files Resources Access Versions

Allow List

@rosterize

Allow Edit List

Jobs Auto-Share

@rosterize_debug

Comma-separated list of entries, eg. "entry, @groupEntry".

Public ☐

Everyone can see the application in applications list.

Enable Workflow (experimental) ☐

Enable this application to invoke other applications on behalf of user (workflow).

Files Monitoring ☐

Model Deployment (versioning)

Model's source code can be stored on GitLab.com.

REST API: https://docs.gitlab.com/ee/api/api_resources.html

Token is needed to access a private repo.

Useful API endpoints:

- Download source code for a commit (rate limited):
GET /projects/:id/repository/archive[.format]?sha=\$COMMIT
- Download CI/CD job's artifact file (no rate limit):
GET /projects/:id/jobs/artifacts/:ref_name/raw/*artifact_path?
job=name
- Get commit's metadata:
GET /projects/:id/repository/commits/\$COMMIT

Cloud Resources in Everest

Everest creates VMs on demand and kills them if idle.

Supported providers:

- Google Cloud Platform
- Yandex Cloud

Preemptible instances are supported (much cheaper)

VM image:

- Can be set up using Packer
- Needs Python version supported by Everest Agent (2.7, 3.6–3.9) + required modules
- Python modules and solvers for the model
- Can include tokens required to download model's source code

bootstrap.sh script:

- Stored in Everest application files
- Downloads model's source code
- Runs run-task.sh script from the model
- Token to download source code can be stored there or in image's environment variables

run-task.sh script:

- Prepares human-readable version string
- Unarchives input files
- Sets up necessary environment variables
- Converts parameters to command line args of the model
- Runs model and auxiliary processes
- Collects output files

Everest task messages are used to update job's status

Environment variables:

- EVEREST_AGENT_PORT
- EVEREST_AGENT_ADDRESS
- EVEREST_AGENT_TASK_ID

```
def connect(self):
    self.connected = False
    try:
        self.sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.sock.connect((self.address, self.port))
        self.sock.sendall(struct.pack('>Ib', len(self.task_id) + 1, 0))
        self.sock.sendall(self.task_id.encode('utf-8'))
        self.connected = True
    except socket.error as err:
        print("Error connecting to Everest agent:", err)
        self.close()

def send(self, msg):
    sent = False
    while not sent:
        try:
            self.sock.sendall(struct.pack('>I', len(msg)))
            self.sock.sendall(msg.encode('utf-8'))
            sent = True
        except (RuntimeError, socket.error) as err:
            print("Error while sending status to Everest agent:", err)
            time.sleep(5)
            self.close()
            self.connect()
```

Conclusion

- Our setup allows both model debugging and production use
- Live calculation status is available
- Minimal costs: VM instances launched and destroyed on demand, preemptible instances can be used
- Model's source code can be stored outside of Everest and VM image

Thank you!

Sergey Smirnov,
IITP RAS (Kharkevich Institute)
sasmir@gmail.com