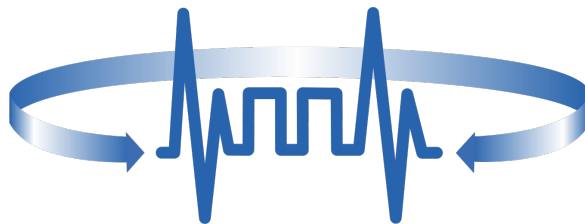


Перенос технологии сбалансированной идентификации математических моделей на платформу распределенных вычислений Everest

Владимир Волошинов, Александр Соколов
<http://bit.ly/VoloshinovGScholar>

Центр распределенных вычислений,
Институт проблем передачи информации им. А.А. Харкевича, РАН



Исследование выполнено за счет гранта Российского научного фонда
№ 22-11-00317, <https://rscf.ru/project/22-11-00317/>

SvF (Simplicity vs Fitting) – технология верификации математических моделей (структурных, не только “регрессионных”) на основе экспериментальных данных.

Успешно применяется в биологии, популяционной динамике, биологии растений, физике плазмы, метеорологии, экологии... Доступна в исходных кодах

<https://github.com/distcomp/SvF>

Основные публикации

1. Sokolov A., Voloshinov V. **Model Selection by Balanced Identification: the Interplay of Optimization and Distributed Computing** // Open Computer Science, 2020
- Соколов А.В., Волошинов В.В. **Выбор математической модели: баланс между сложностью и близостью к измерениям** // International Journal of Open Information Technologies, 2018
 - другие ссылки здесь <https://github.com/distcomp/SvF#references>
 - руководство пользователя https://github.com/.../SvF/SvF_UserGuide.pdf

Сценарий расчетов основан на решении **двухуровневой задачи** оптимизации специального типа: на нижнем уровне решается набор независимых задач, напоминающих обратные задачи с Тихоновской регуляризацией; на верхнем уровне — выбираются «наилучшие» коэффициенты регуляризации.

Схема работы SvF-технологии

Главная задача: сформировать и верифицировать мат. модель исследуемого явления по имеющемуся набору экспериментальных данных.

На вход: данные экспериментов (реляц. БД TXT, XLS, CSV ... см. User Manual)

+ предположение о структуре мат. модели и ее математическая запись (набор уравнений, неравенств, интегро-дифференциальных уравнений и т.п.) с явным указанием того, какие параметры или функциональные зависимости требуется определить (идентифицировать).

Все остальные действия выполняются автоматически:

1. Формирование задачи двухуровневой оптимизации
2. Дискретизация дифференциальных и/или интегральных уравнений, если они есть в модели.
3. Генерация вычислительного сценария в виде **Python**-программы
4. Выполнение SvF-сценария полностью на локальной машине, либо еще привлекая ресурсы платформы Everest, <http://everest.distcomp.org/>

Пример. Обратная задача для ОДУ

Идентифицировать динамическую систему по набору точек на траектории.

Гипотеза: ODE (1-го или 2-го порядка). Нужно определить правую часть.

$$\left. \begin{array}{l} \{(\tilde{x}_d, t_d) : d=1:N_d\}, \tilde{x}_d = x(t_d) \pm \text{err} \\ \dot{x} = F(t, x(t)) \mid \ddot{x} = F(t, x(t)), t \in [t_L, t_U] \end{array} \right\} \Rightarrow \mathbf{x}(t), \mathbf{F}(t, \mathbf{x}) - ?$$

SvF задачи нижнего уровня: задачи кросс-валидации

Пример. Обратная задача для ОДУ

Идентифицировать динамическую систему по набору точек на траектории.

Гипотеза: ODE (1-го или 2-го порядка). Нужно определить правую часть.

$$\left\{ \left\{ (\tilde{x}_d, t_d) : d=1:N_d \right\}, \tilde{x}_d = x(t_d) \pm \text{err} \right. \\ \left. \dot{x} = F(t, x(t)) \mid \ddot{x} = F(t, x(t)), t \in [t_L, t_U] \right\} \Rightarrow x(t), F(t, x) - ?$$

SvF CV-задачи нижнего уровня: на “тренировочных наборах данных”

$$D_k \doteq \{d=1:N_d, d \neq k\}, k=1:N_d$$

коэффициенты
регуляризации

data fitting (точность) simplicity (гладкость.) $\alpha = (\alpha_t, \alpha_x)$

$$\frac{1}{|D_k|} \sum_{d \in D_k} (\tilde{x}_d - x(t_d))^2 + R(\alpha, F(\cdot, \cdot)) \rightarrow \min_{x(\cdot), F(\cdot, \cdot)},$$

$$\dot{x} = F(x(t)) \mid \ddot{x} = F(x(t)), t \in [t_L, t_U],$$

$$x(\cdot) \in C^2[t_L, t_U], F(\cdot) \in C^2[x_L, x_U].$$

$$R(\alpha, F(\cdot, \cdot)) =$$

$$\int_{t_L}^{t_U} \int_{x_L}^{x_U} \left(\alpha_t^2 (F''_{tt}(t, x))^2 + 2\alpha_t \alpha_x (F''_{tx}(t, x))^2 + \alpha_x^2 (F''_{xx}(t, x))^2 \right) dt dx$$

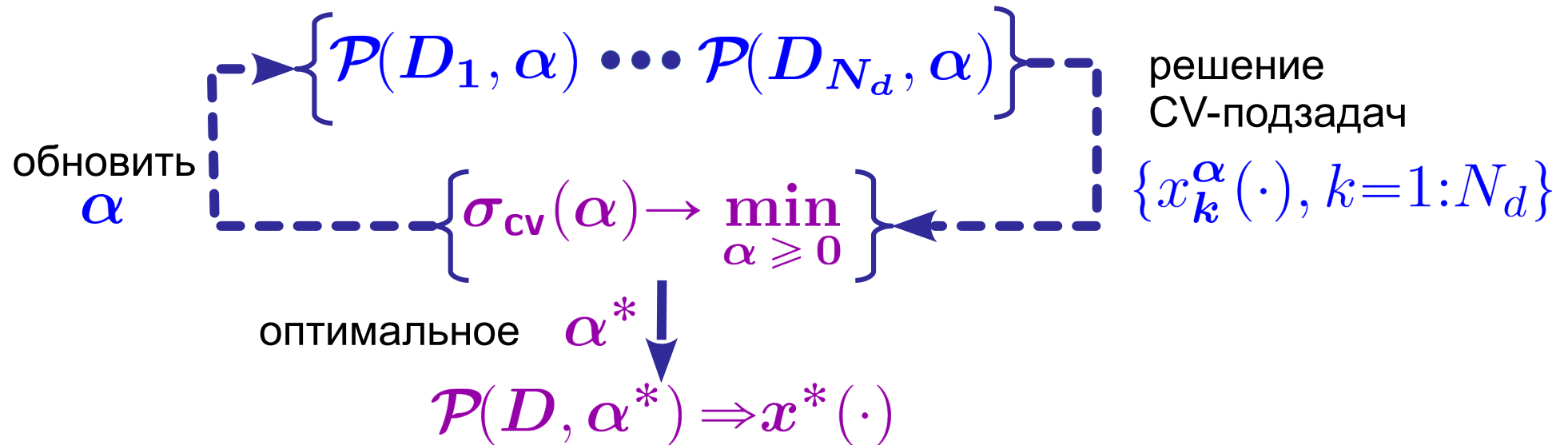
SvF задачи нижнего уровня: задачи кросс-валидации $\mathcal{P}(D_k, \alpha) \Rightarrow x_k^\alpha(\cdot)$

Пример. Обратная задача для ОДУ. Двухуровневая опт.

CV-ошибка для заданного $\alpha = (\alpha_t, \alpha_x)$ $\mathcal{P}(D_k, \alpha) \Rightarrow x_k^\alpha(\cdot)$ ($k=1:N_d$)

$$\sigma_{cv}(\alpha) = \sqrt{\frac{1}{N_d} \sum_{k=1:N_d} (\tilde{x}_k - x_k^\alpha(t_k))^2}$$

SvF задача 1-го уровня: минимизировать CV-ошибку по α



$$\sigma_{mse}(\alpha^*) = \sqrt{\frac{1}{N_d} \sum_{k=1:N_d} (\tilde{x}_k - x^*(t_k))^2}$$

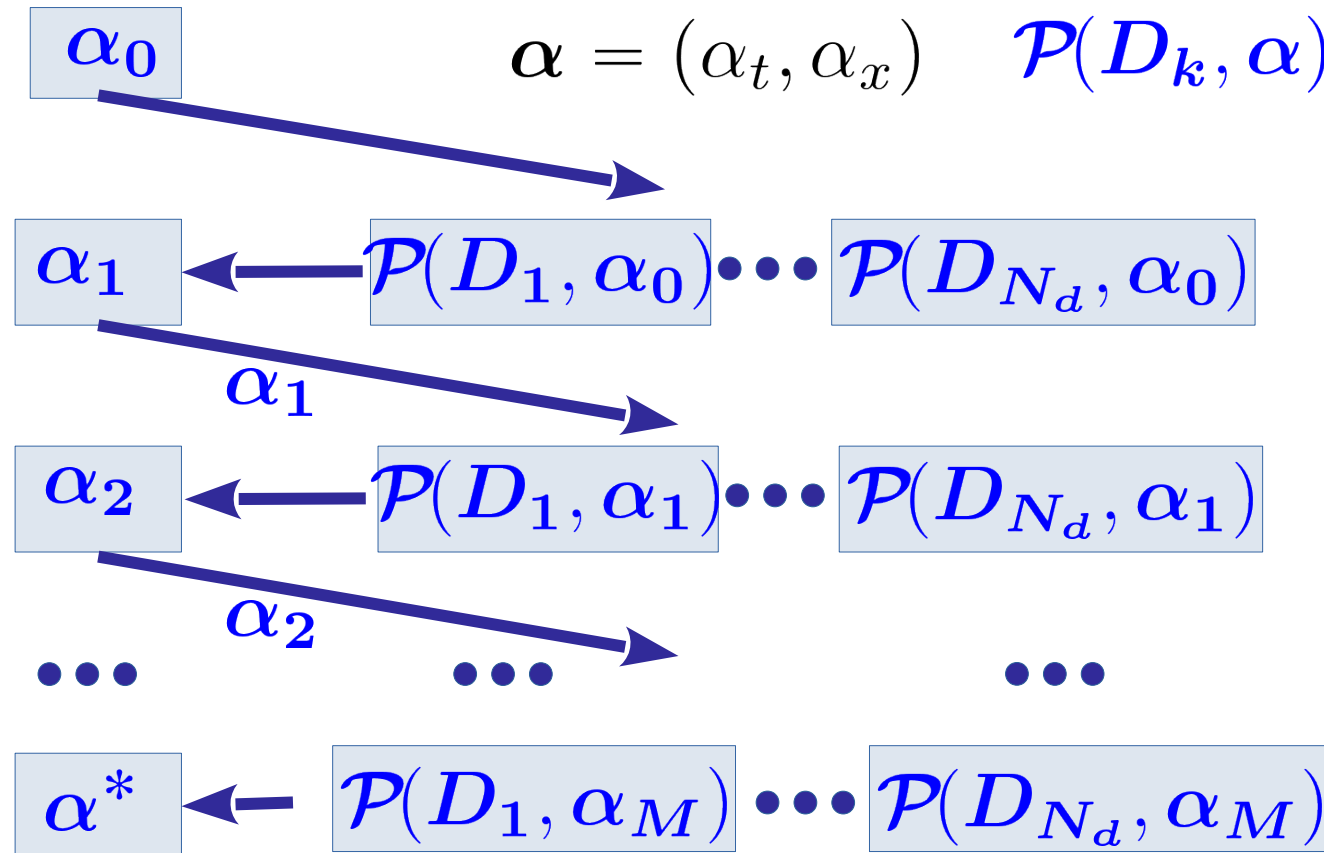
Ответ: средне-квадратичная ошибка (RMSE)

Сейчас все “вариационные” выражения дискретизируются, чтобы получить **конечномерную задачу матем. программирования**, которую можно решать универсальными солверами: **Ipopt** и **SCIP** (глобальная оптимизация)

Типовая цепочка задач оптимизации

$$\alpha = (\alpha_t, \alpha_x) \quad \mathcal{P}(D_k, \alpha) \Rightarrow x_k^\alpha(\cdot) \quad (k=1:N_d)$$

$$\sigma_{cv}(\alpha) \rightarrow \min_{\alpha \geq 0}$$



$$\mathcal{P}(D, \alpha^*) \Rightarrow x^*(\cdot)$$

$$\sigma_{mse}(\alpha^*) = \sqrt{\frac{1}{N_d} \sum_{k=1:N_d} (\tilde{x}_k - x^*(t_k))^2}$$

Почему SvF-сервис & распределенные вычисления

- SvF-приложение можно запускать “локально”: все задачи решаются на том же хосте, где выполняется управляющий SvF-сценарий
- Но, м.б. придется решать десятки достаточно трудоемких задач
- SvF-приложение уже может использовать сервис оптимизации Everest для одновременного решения независимых подзадач на удаленных ресурсах
- Но, даже с помощью Everest расчет может занять много часов и все это время управляющая машина должна быть включена.

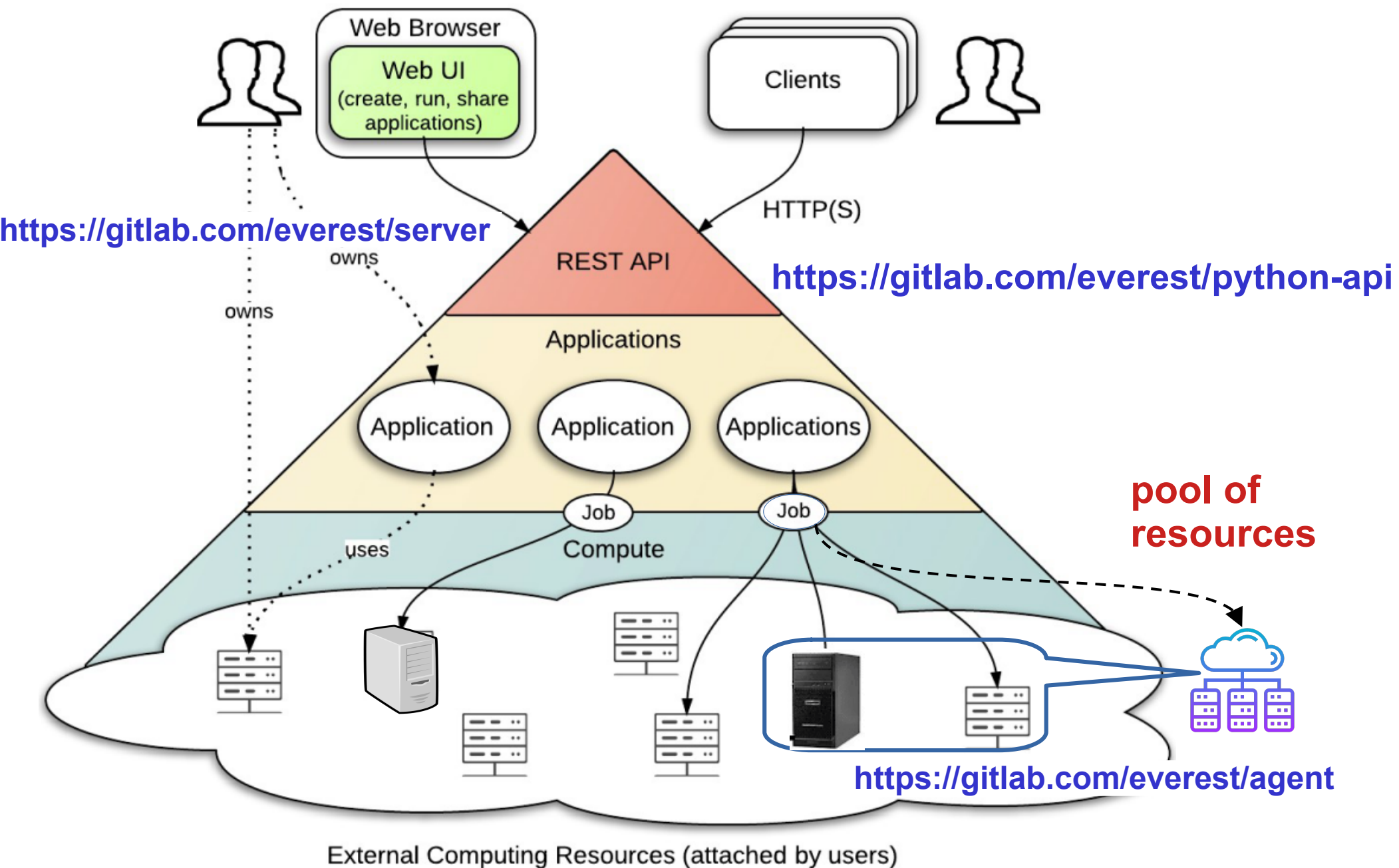
Была бы удобной возможность запускать расчет в режиме
“отправил данные и жди оповещения на почту”...

Возможности SvF-инструментария позволяют это реализовать !

- Python 3.7+ - основной язык
- пакет **Pyomo**, www.pyomo.org, **PY**thon **O**pt. **M**odeling **O**bjects, готовит исходные данные для основных типов ЗМП: LP, QP, NLP, MILP, MINLP, Stochastic ..., DAE (Differential and Integral equations), ...
Pyomo совместим почти со всеми свободными и коммерческими решателями **Ipopt**, **SCIP**, **CBC**, **CPLEX**, **Gurobi**, **COPT**, и т.п.
- платформа Everest, <http://everest.distcomp.org/>, и ее подсистема оптимизации <https://optmod.distcomp.org>
 - Everest Python API, <https://gitlab.com/everest/python-api>
 - Everest-приложение, <https://optmod.distcomp.org/apps/vladimirv/SSOP> для решения набора независимых задач
 - Everest App., <https://optmod.distcomp.org/apps/vladimirv/svf-remote> для удаленного запуска SvF-сценария

Архитектура платформы Everest (пул ресурсов!)

Describe/Develop/Deploy REST-services representing existing applications



Everest-приложение svf-remote (in / out parameters)

Everest Opt

APPLICATIONS

List

Create new

JOB

RESOURCES

List

Create new

GROUPS

ABOUT

DOCUMENTATION

Optimization Modelling Services



Vladimir Voloshinov
vladimirv

apps > svf-remote

READY

0.1.0

Unstar



4

Export

Edit

About

Parameters

Submit Job

Discussion

Inputs

	Title	Name	Type	Values	Default	Description
	Data and *.res	data	array[URI]			
✓	MNG-file	mng	URI			

Input data (txt, xls)

MNG-файл (txt, odt) с SvF-заданием

Outputs

	Title	Name	Type	Description
	results	results	URI	Results *.sol, *.res, *.png
✓	stderr	stderr	URI	Copy of console output
✓	stdout	stdout	URI	Copy of console output

ZIP-архив

Пример MNG-файла, формат *.ODT. Маятник с трением.

Here you can place arbitrary text and pictures

BoF-SvF

```
Runmode = 'P&S'
SvF.DrawMode = 'File'          #'File&Screen'
SvF.Resources = ["pool-scip-ipopt"] #["abc_pc" ]
CVNumOfIter      10 # Number of iter.
CVstep           21 # Numberof CV subproblems
```

Select x, t from Spring5.dat

GRID:

t ∈ [-1., 2.5, 0.025]

```
Var:      x ( t )
          v ( t )
          K
          μ      # will be replaced to muu in Python code
          xr
```

SchemeD1 = Central

EQ: $\frac{d^2}{dt^2}(x) == -K * (x - xr) - \mu * v$

$v == \frac{d}{dt}(x)$

EQ:

$$\frac{d^2}{dt^2}(x) == -K * (x - xr) - \mu * v$$
$$v == \frac{d}{dt}(x)$$

OBJ: x.Complexity (Penal[0]) + x.MSD()

Draw


EOF

Some remarks, figures, formulas may be added here

Запись формул в символьном виде,
в т.ч. LaTeX (в *.ODT документах)

Простота реализации svf-remote (благодаря возможностям пакета SvF)

https://optmod.distcomp.org/apps/vladimirv/svf-remote/edit ☆

Optimization Modelling Services  Vladimir Voloshinov
vladimirv

apps > svf-remote > edit

Cancel Delete Update

Metadata Description Parameters Configuration Files Resources

Access Versions

Command

runSvF-remote.sh

Refer to input values as \${param}

Input Mappings + Add input mapping Output Mappings + Add output mapping

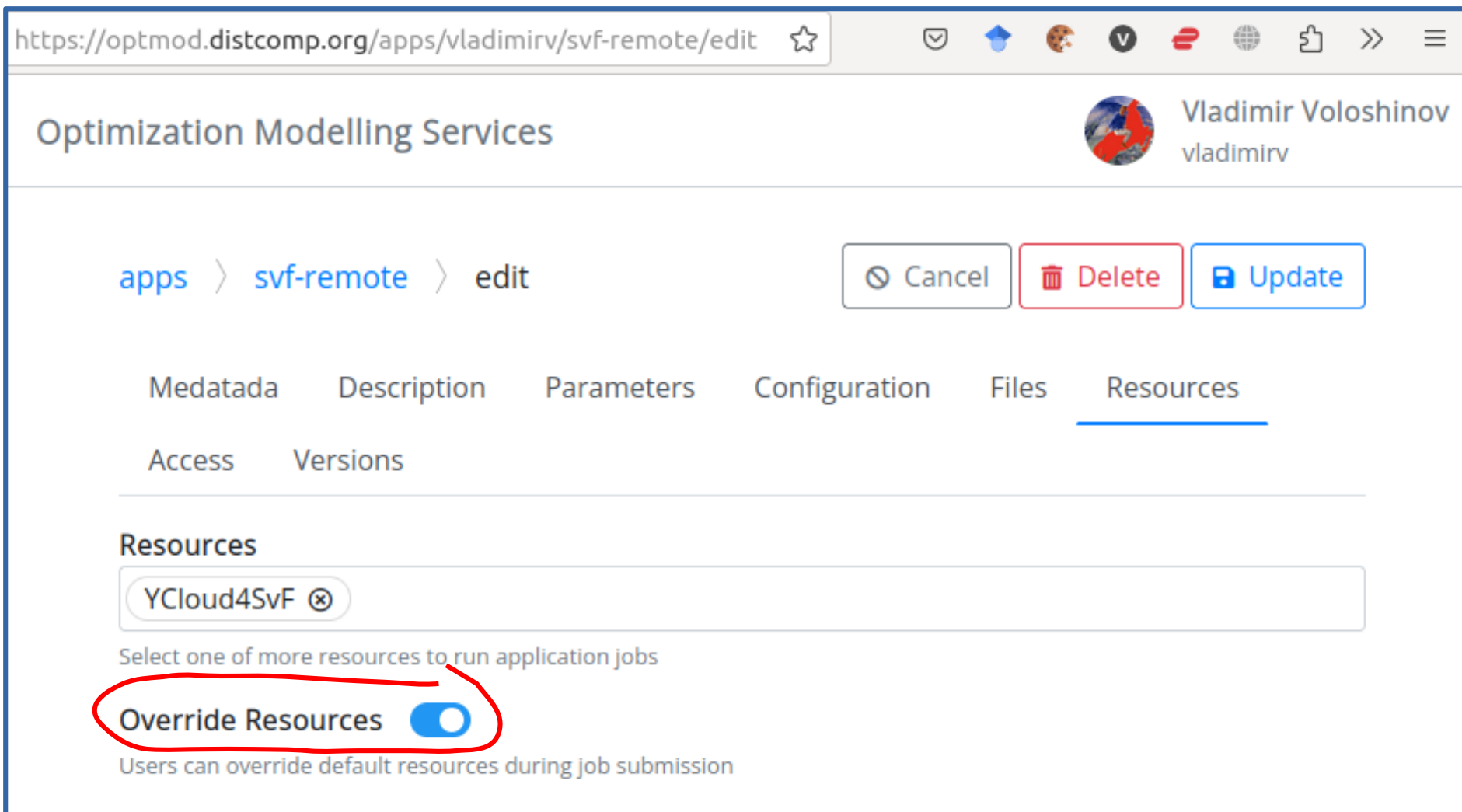
```
#!/bin/bash
```

```
<PATH TO SvF Folder>/SvF/runSvF31.sh  
zip results.zip *.sol *.png *.res
```

путь к runSvF-remote.sh – прописать в \$PATH

svf-remote позволяет выбирать головной ресурс

Пользователь Everest может выбирать ресурс для запуска управляющего скрипта: где установлено SvF-инструментарий и настроен runSvF-remote.sh.



The screenshot shows the 'svf-remote' edit page in the 'Optimization Modelling Services' interface. The browser address bar displays 'https://optmod.distcomp.org/apps/vladimirv/svf-remote/edit'. The user profile 'Vladimir Voloshinov' with the handle 'vladimirv' is visible in the top right. The breadcrumb navigation shows 'apps > svf-remote > edit'. Action buttons for 'Cancel', 'Delete', and 'Update' are present. A tabbed interface includes 'Metadata', 'Description', 'Parameters', 'Configuration', 'Files', and 'Resources' (which is selected). Below the tabs are 'Access' and 'Versions' links. The 'Resources' section contains a text input field with 'YCloud4SvF' and a close icon. Below this is the instruction 'Select one of more resources to run application jobs'. At the bottom, the 'Override Resources' toggle switch is turned on and circled in red, with the explanatory text 'Users can override default resources during job submission' underneath.

https://optmod.distcomp.org/apps/vladimirv/svf-remote/edit

Optimization Modelling Services

Vladimir Voloshinov
vladimirv

apps > svf-remote > edit

Cancel Delete Update

Metadata Description Parameters Configuration Files Resources

Access Versions

Resources

YCloud4SvF

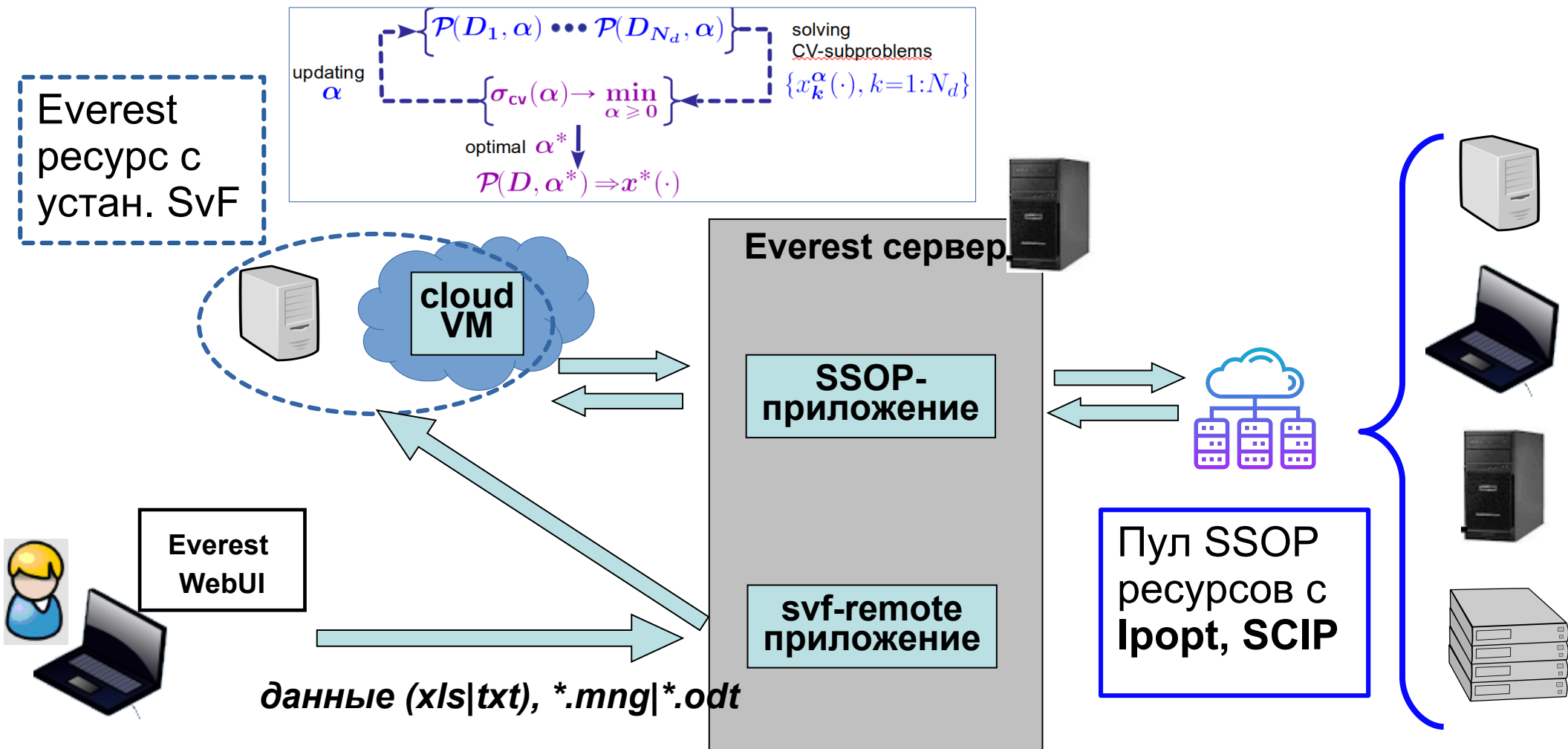
Select one of more resources to run application jobs

Override Resources ☒

Users can override default resources during job submission

Схема работы и потоков данных *svf-remote*

Сервис ***svf-remote*** позволяет через браузер отправлять данные и MNG-файл с SvF-заданием на выполнение.



Результаты

- Сервис **svf-remote** использует все возможности SvF-инструментария:
 - символьную запись задачи идентификации MNG-файле (ASCII текст, OpenOffice *.odt) включая LaTeX-формулы;
 - доступ ко всем Everest-ресурсам, где установлены решатели;
 - возврат результатов в численной и графической форме.
- Теперь SvF легко применять «в браузере» для продолжительных расчетов по принципу «запустил и жди оповещения на почту».
- Возможно выбирать Everest-ресурс (с установленным SvF-инструментарием), где будет выполняться управляющий сценарий (сервер, настольная машина, VM, и т. п.).

Дальнейшие планы

- Установить **svf-remote** на наших оживших серверах в ИППИ РАН.
- Для экономии денег сменить “обычный” Everest-агент на Yandex Cloud VM на Everest YCC Agent (for Yandex Cloud Container).
- Попробовать вместо Everest-сервиса Jupyter-notebook (на удаленной машине) как интерфейс к SvF (Everest SSOP останется с нами).

Благодаю за внимание.

Вопросы?

<http://bit.ly/VoloshinovGScholar>

vladimir.voloshinov@gmail.com

Дискретизация автономного ОДУ: полином + сетка

One of approach in SvF: outer function is replaced with polynomial with unknown coefficients, inner function — unknown values on a meshgrid (by t):

$$\begin{aligned} F(x) \sim \mathcal{P}(x) &= \sum_{p=0}^P c_p \cdot x^p & x(t) &\sim \{(x_i, t_k) : i = 0:N_x, k=0:N_t\}, \\ & & x_i &= x_{\mathbf{L}} + i \cdot \Delta x, \Delta x = \frac{x_{N_x} - x_{\mathbf{L}}}{N_x}, \\ & & t_k &= t_{\mathbf{L}} + k \cdot \Delta t, \Delta t = \frac{t_{N_t} - t_{\mathbf{L}}}{N_t}, \\ \left\{ \dot{x} = F(x(t)) \right\} &\sim \left\{ \frac{x(t_k) - x(t_{k-1}))}{\Delta t} = \sum_{p=0}^P c_p \cdot \left(\frac{x(t_k) + x(t_{k-1}))}{2} \right)^p, k=1:N_t \right\} \\ \left\{ \ddot{x} = F(x(t)) \right\} &\sim \left\{ \frac{x(t_{k+1}) - 2x(t_k) + x(t_{k-1}))}{\Delta t^2} = \sum_{p=0}^P c_p \cdot x(t_k)^p, k=1:(N_t-1) \right\} \\ x(t_d) &\sim \hat{x}_d = \frac{t_k - t_d}{\Delta t} x(t_{k-1}) + \frac{t_{k+1} - t_d}{\Delta t} x(t_k), \quad t_d \in [t_{k-1}, t_k] \\ \frac{1}{D} \sum_{d=1:D} (\tilde{x}_d - \hat{x}_d)^2 + \alpha \sum_{\dots} \left(\frac{\mathcal{P}(x_{i+1}) - 2\mathcal{P}(x_i) + \mathcal{P}(x_{i-1}))}{\Delta x^2} \right)^2 \Delta x &\rightarrow \min_{x_i, c_p}. \end{aligned}$$

This we have **nonlinear mathematical programming problem** with polynomials of non-small degrees (up to 7, 8)