README

WHAT IS PACEVAL SERVICE?

paceval-service is a Linux server for **ARM64 processors** (e.g. Raspberry Pi and APPLE M1/M2) to perform mathematical calculations on a remote computer or server. pacevalservice is based on the mathematical engine paceval., which is fully described on the homepage https://paceval.com.
(Annotation: For **x64 processors (Intel and AMD)** see https://hub.docker.com/repository/docker/paceval/paceval-service)

HOW CAN I GET PACEVAL SERVICE ON MY LINUX?

Just run the following command lines in the terminal to get and start the paceval-service with Docker:

sudo docker pull paceval/paceval-service_linux_arm64:latest sudo
docker run -p 8080:8080 -d paceval/paceval-service_linux_arm64

HOW CAN I USE PACEVAL SERVICE TO CREATE CALCULATIONS?

(see https://app.swaggerhub.com/apis-docs/paceval/paceval-service/4.04 for more details)

Jumpstart :)

The paceval-service listens on port 8080 by default. The port can be freely configured, see the link to the source code below. Calculations are then created via HTTPS request, for example in the browser or with cURL, https://en.wikipedia.org/wiki/CURL.

To create a calculation, simply go to the following URL in your browser:
(**Note:** Use the correct encoding in the functionString in the URL (GET) and data (POST), e.g. replace the '+' character with '%2B'.)

http://localhost:8080/Demo/?functionString=sin(x*cos(x))^(1/y)&numberOfVariables=2&variables=x;y&values=0.5;2&interval=yes

This creates a calculation object for the function "-sin (x * cos (x)) ^ (1 / y)" and immediately performs the calculation with the "2" variables "x; y" for the values "0.5; 2". Variables and values are always separated by a ";". With "interval=yes" it is indicated that in addition to the computer-precise calculation, the upper and lower interval of the calculation is also given. The exact value of the calculation is then in this interval. You
can find more detailed explanations about the interval here:
https://paceval.com/developers/#precision.

In addition, with the calculation you receive a reference to the generated calculation object for the function. From now on you can simply use this reference to get calculations for further values. In this implementation of the paceval-service, references are valid for 1 hour, which is extended to 1 hour from the time of access each time a reference is accessed. If only the

reference to a calculation object is used, the sometimes very long function does not have to be passed every time. That saves time and computing power.

For example, if you have received a reference "handle_pacevalComputation: 2370169936048", simply call up the following URL for a further calculation with the values 0.2 and 2 for x and y:

http://localhost:8080/GetComputationResult/?handle_pacevalComputation=2370169936048&values=0.2;2

This allows you to perform complex calculations of any length on the server. A common use case is, for example, energy saving on battery-operated IoT devices such as quadrocopters (so-called drones) or smartphones. Of course, you can also do these calculations with GET and POST with cURL. In the examples above, these would be the following command lines: (**Note:** Use the correct encoding in the functionString in the URL (GET) and data (POST), e.g. replace the '+' character with '%2B'.)

GET –
curl -X GET -k "http://localhost:8080/Demo/?functionString=-sin(x*cos(x))^(1/y)&numberOfVariables=2&variables=x;y&values=0.5;2&interval=yes"

or POST –
curl --data "functionString=-sin(x*cos(x))^(1/y)&numberOfVariables=2&variables=x;y&values=0.5;2&interval=no" -X POST http://localhost:8080/Demo/

You can also use the reference to the calculation object to carry out further calculations with new variables. For example, if you have received the reference "handle_pacevalComputation: 2430243604080" this time, just use the command line with curl for another calculation with the values 0.2 and 2 for x and y:

GET – **curl -X GET -k "http://localhost:8080/GetComputationResult/?handle_pacevalComputation=2430243604080&values=0.2;2"**

or POST – **curl --data "handle_pacevalComputation=2430243604080&values=0.2;2" -X POST http://localhost:8080/GetComputationResult/**

The source code for this paceval-service is on https://github.com/paceval/paceval/tree/main/examples_sources/NodeJS/pacevalservice_linux , see paceval-server.js.
With the libraries from paceval. you can easily create additional services, e.g. for Microsoft Windows or Apple macOS.