paceval / **paceval** Public

paceval / examples_sources / Python_examples / **paceval-4.25.4** /

paceval Merge branch 'main' of https://github.com/paceval/paceval

8660dc1 · 1 minute ago

Name	Name	Last commit date
..		
dist	Updated Python package	2 minutes ago
paceval.egg-info	Added Python package 4.2...	24 minutes ago
src	Added Python package 4.2...	24 minutes ago
LICENSE	Added Python package 4.2...	24 minutes ago
pyproject.toml	Added Python package 4.2...	24 minutes ago
readme.md	Update readme.md	13 minutes ago
setup.cfg	Added Python package 4.2...	24 minutes ago

readme.md

paceval with Python - the Mathematical Engine as a Service (e.g. for multi-party computations)

paceval at PyPI

paceval at SwaggerHub

WHY DO I NEED A MATHEMATICAL ENGINE?

Many connected devices or so-called IoT solutions require **complex mathematical calculations to work correctly or make decisions**. This can be, for example, a smartphone or a remote device that performs predictive analysis or a self-flying drone that evaluates objects in recorded videos or images in real time during flight. These devices do not have the computing power to perform such mathematical calculations themselves. Or these devices, because they are battery powered, can't even do the **intensive math calculations because that consumes battery time**.

This Mathematical Engine as a Service provides a powerful and fast **remote math coprocessor service for IoT solutions** based on a Linux server for x64 (Intel and AMD) or ARM64 (e.g. Raspberry Pi or APPLE M1/M2) processors. Equipped with a simple interface, it will allow battery-powered devices to perform complex mathematical operations remotely and very quickly, **avoiding increasing power consumption of the device itself**.

HOW CAN I USE CALCULATIONS WITH THIS MATHEMATICAL ENGINE?

To create a calculation the device simply calls the following function:

```
paceval.Demo("http://paceval-service.com", "-sin(x\\\*cos(x))\^(1/y)",
"2", "x;y","0.5;2", "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.

Of course you can specify any mathematical function and any number of variables and also other and longer variable names. :)

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. **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_Computation: 115626720", simply call up the following function for a further calculation with the values 0.46577 for x and 2.61 for y.

```
paceval.GetComputationResult("http://paceval-service.com", "115626720",
"0.46577;2.61")
```

This allows you to **perform complex calculations of any length and with any number of variables on the server**. Please note that this is our test server. :)

This test server is a 4-core ARM64 Linux server with only 4GB of memory, although it's pretty fast.

HOW CAN I GET THIS MATHEMATICAL ENGINE ON MY OWN SERVER?

Just run this command line in the terminal to get and start the service with Docker:

LINUX FOR x64 PROCESSORS (Intel and AMD)

```
sudo docker pull paceval/paceval-service_linux_x64:latest

sudo docker run -p 8080:8080 -d paceval/paceval-service_linux_x64
```

LINUX FOR ARM64 PROCESSORS (e.g. Raspberry Pi or APPLE M1/M2)

```
sudo docker pull paceval/paceval-service_linux_arm64:latest

sudo docker run -p 8080:8080 -d paceval/paceval-service_linux_arm64
```

paceval. - website

Send email to paceval

Code example

```
import paceval

import json

from types import SimpleNamespace

#paceval Server
url = "http://paceval-service.com"

demoResponse = paceval.Demo(url, "-sin(x\*cos(x))\^(1/y)", "2",
"x;y","0.5;2", "yes").text
```

```
print(demoResponse)
```

Return:
{
"handle_pacevalComputation":140660720254832,
"result":"-0.651801782452278",
"interval-min-result":"-0.651801782452306",
"interval-max-result":"-0.65180178245225",
"error-type-number":0,
"error-position":"",
"error-type":"[NO ERROR]",
"error-message":"No error has occurred for this computation object (PACEVAL_ERR_NO_ERROR).",
"time-calculate":"0.000859s",
"version-number":4.25
}

```
createComputationResponse = paceval.CreateComputation(url, "-sin(x\*cos(x))\^(1/y)", "2", "x;y", "yes").text

print(createComputationResponse)
```

Return:
{
"handle_pacevalComputation":140660720337664,
"function-10chars":"-sin(x*cos(...)",
"function-length":20,
"error-type-number":0,
"error-position":"",
"error-type":"[NO ERROR]",
"error-message":"No error has occurred for this computation object (PACEVAL_ERR_NO_ERROR).",
"time-create":"0.000253s",
"version-number":4.25
}

```
getComputationResultResponse = paceval.GetComputationResult(url, "140660720254832", "0.5;2").text

print(getComputationResultResponse)
```

Return:
{
"handle_pacevalComputation":140660720254832,
"result":"-0.651801782452278",
"interval-min-result":"-0.651801782452306",
"interval-max-result":"-0.65180178245225",
"error-type-number":0,
"error-position":"",
"error-type":"[NO ERROR]",
"error-message":"No error has occurred for this computation object (PACEVAL_ERR_NO_ERROR).",
"time-calculate":"0.000264s",
"version-number":4.25
}

```
getErrorInformationResponse = paceval.GetErrorInformation(url,
"140660720254832").text

print(getErrorInformationResponse)
```

Return:
{
"handle_pacevalComputation":140660720254832,
"hasError":false,
"error-type-number":0,
"error-position":"",
"error-operator":"(not defined)",
"error-type":"[NO ERROR]",
"error-message":"No error has occurred for this computation object (PACEVAL_ERR_NO_ERROR).",
"version-number":4.25
}

```
getComputationResultExtResponse = paceval.GetComputationResultExt(url,
"140660720254832", "3", "0.5;2;0.4;2;0.3;2").text

print(getComputationResultExtResponse)
```

Return:

{
"number-of-multiple-values":3, "handle_pacevalComputation":140660720254832,
"hasError":false,
"results": ["-0.651801782452278", "-0.600121659758506", "-0.531689249020161"],
"interval-min-results": ["-0.651801782452306", "-0.600121659758535",
"-0.531689249020193"],
"interval-max-results": ["-0.65180178245225", "-0.600121659758477",
"-0.53168924902013"],
"error-type-numbers": [0,0,0],
"time-calculate":"0.000502s",
"version-number":4.25
}

```
getComputationInformationXMLResponse =
paceval.GetComputationInformationXML(url, "140660720254832").text

print(getComputationInformationXMLResponse)
```

Return:

{
"handle_pacevalComputation":140660720254832,
"information-XML":"<br><br><paceval.-Computation><br><br>\
<version>4.04</version><br><br> <function50Characters>-sin(x*cos(x))^(1/y)
</function50Characters><br><br> <functionLength>20</functionLength><br>
<br> <numberOfVariables>2</numberOfVariables><br><br>
<useInterval>true</useInterval><br><br> <errorMessage>No error has occurred
for this computation object (PACEVAL_ERR_NO_ERROR).</errorMessage><br><br>
<errorDetails>[NO ERROR]</errorDetails><br><br> <maxPrecisionType>long
double</maxPrecisionType><br><br> <numberOfNodes>11</numberOfNodes>
<br><br> <numberOfCores>20</numberOfCores><br><br>
<numberOfThreads>1</numberOfThreads><br><br>
<numberOfThreadsFailure>0</numberOfThreadsFailure><br><br>
<cacheTypes>Inner Caching, Outer Caching, Lookahead Caching</cacheTypes><br>
<br> <cacheHitsACC>3</cacheHitsACC><br><br></paceval.-Computation><br>
<br>", "version-details":"[libpaceval_linux_staticLIB.a] and
[libpaceval_linux_sharedLIB.so][4.25, 64 bit] developer version (non-commercial use
only) - Copyright 2015-2024. - All rights reserved. (paceval.[Registered Trade Mark])",
"version-number":4.25
}

```
getMultipleComputationsResultsResponse =
paceval.GetMultipleComputationsResults(url, "989554800;988662768", "2",
"0.5;2;0.4;2").text

print(getMultipleComputationsResultsResponse)
```

Return:
{
"number-of-multiple-computations":2,
"handle_pacevalComputations":[989554800, 988662768], "hasError":false,
"results":["-0.651801782452278", "-0.651801782452278"],
"interval-min-results":["-0.651801782452306", "-0.651801782452306"],
"interval-max-results":["-0.65180178245225", "-0.65180178245225"],
"error-type-numbers":[0,0],
"time-calculate":"0.000675s",
"version-number":4.25
}

```
getMultipleComputationsResultsExtResponse =
paceval.GetMultipleComputationsResultsExt(url, "117867040;118054176",
"2", "3", "0.61;3.1;53.21;0.62;3.2;53.22;0.63;3.3;53.23").text

print(getMultipleComputationsResultsExtResponse)
```

Return:
{
"number-of-multiple-computations":2,
"number-of-multiple-calculations":3,
"handle_pacevalComputations":[117867040, 118054176],
"hasError":false,
"results":["-51.319", "-51.236", "-51.151", "-164.341", "-169.684", "-175.029"],
"interval-min-results":["-51.319", "-51.236", "-51.151", "-164.341", "-169.684", "-175.029"],
"interval-max-results":["-51.319", "-51.236", "-51.151", "-164.341", "-169.684", "-175.029"],
"error-type-numbers":[0, 0, 0, 0, 0, 0],
"time-calculate":"0.000728s",
"version-number":4.25,
}