

Saxon/C - Support #4942

How to prevent Saxon/C Python bindings from trying to start a new java VM when another is active? (JNI_CreateJavaVM() failed with result: -5)

2021-03-18 22:41 - Kor Foo

Status:	In Progress	Start date:	2021-03-18
Priority:	Normal	Due date:	
Assignee:	O'Neil Delpratt	% Done:	0%
Category:	Python	Estimated time:	0:00 hour
Sprint/Milestone:		Spent time:	0:00 hour
Found in version:	1.2.1		

Description

I am building a flask api that allows users to pass an xml and a transformation that returns the xml on which the transformation is performed using Saxon/C's python API (<https://www.saxonica.com/saxon-c/doc/html/saxonc.html>).

The incoming endpoint looks like this (removed logging and irrelevant info):

```
@app.route("/v1/transform/", methods=["POST"])
def transform():
    xml = request.data
    transformation = request.args.get("transformation")

    result = transform_xml(xml, transformation)

    return result
```

The transform function looks like this:

```
def transform_xml(xml: bytes, transformation: str) -> str:
    with saxonc.PySaxonProcessor(license=False) as proc:
        base_dir = os.getcwd()
        xslt_path = os.path.join(base_dir, "resources", transformation, "main.xslt")

        xslt_proc = proc.new_xslt30_processor()

        node = proc.parse_xml(xml_text=xml.decode("utf-8"))

        result = xslt_proc.transform_to_string(stylesheet_file=xslt_path, xdm_node=node)

    return result
```

The xslt's are locally available and a user should choose one of the available ones by passing the corresponding transformation name.

Now the problem is, this works (fast) for the first incoming call, but the second one crashes:

```
JNI_CreateJavaVM() failed with result: -5
DAMN ! worker 1 (pid: 517095) died :( trying respawn ...
```

What does work is changing the transform_xml function like this:

```
proc = saxonc.PySaxonProcessor(license=False)

xslt_path = self.__get_path_to_xslt(transformation)

xslt_proc = proc.new_xslt30_processor()

node = proc.parse_xml(xml_text=xml.decode("utf-8"))

result = xslt_proc.transform_to_string(stylesheet_file=xslt_path, xdm_node=node)
```

```
return result
```

But this leads to the resources never getting released and over time (1k+ requests) this starts to fill up the memory. I have tried fiddling with the JET garbage collector settings (-Djet.gc.no.swap -Djet.gc.ratio -Djet.gc.heaplimit -Djet.gc.defrag.holes.threshold), but this doesn't help to reduce the memory footprint.

It seems like Saxon is trying to create a new VM while the old one is going down.

I found this thread from 2016: <https://saxonica.plan.io/boards/4/topics/6399> but this didn't clear it up for me. I looked at the github for the pysaxon repo, but I have found no answer to this problem.

History

#1 - 2021-03-19 10:37 - O'Neil Delpratt

- Category set to Python
- Status changed from New to In Progress
- Assignee set to O'Neil Delpratt
- Found in version set to 1.2.1

Thank you for submitting the bug issue. Linking it to the stack overflow post:

<https://stackoverflow.com/questions/66693687/how-to-prevent-saxon-c-python-bindings-from-trying-to-start-a-new-java-vm-when-a>

Investigating it now.

#2 - 2021-03-19 12:09 - Kor Foo

I have made a very simple application showcasing the problem: <https://github.com/RudolfDG/saxon-flask-api>

Kind regards

#3 - 2021-03-19 13:06 - O'Neil Delpratt

Thank you for supplying the github project. I got down to the following command:

```
4. flask run
```

but fails with this error:

```
No command 'flask' found, did you mean:  
Command 'flasm' from package 'flasm' (universe)  
flask: command not found
```

Is there something I have missed

#4 - 2021-03-19 13:34 - Kor Foo

Are you running Linux? Did you activate your virtual environment? I have added the activation of the virtual environment to the steps in the repo.

I tried to follow the steps myself in a new folder, starting from git clone to flask run and it worked. These are the exact commands used:

```
git clone git@github.com:RudolfDG/saxon-flask-api.git && cd saxon-flask-api  
python3 -m venv env  
source env/bin/activate  
pip install -r requirements.txt  
export FLASK_APP=app.py  
export PYTHONPATH=~/.Downloads/Saxonica/SaxonHEC1.2.1/Saxon.C.API/python-saxon  
flask run
```

Note that this is on Linux and I have built the python saxon package in ~/.Downloads/Saxonica/SaxonHEC1.2.1/Saxon.C.API/python-saxon

#5 - 2021-03-19 14:05 - O'Neil Delpratt

Yes I am on linux.

python3 -m venv env gives me the error:

```
saxon-flask-api/env/bin/python3', '-Im', 'ensurepip', '--upgrade', '--default-pip']] returned non-zero exit st  
atus 1.
```

#6 - 2021-03-19 14:23 - Kor Foo

Normally the venv package is in the standard library of Python3 <https://docs.python.org/3/library/venv.html>. If you don't have it you should be able to use

```
python -m venv env --without-pip
source env/bin/activate
curl https://bootstrap.pypa.io/get-pip.py | python
```

This makes a new virtual environment without pip and only installs pip in the new virtual environment after activating it. This way doesn't change anything to your global python, but should allow it to work for this project.

It should thus look like this:

```
git clone git@github.com:RudolfDG/saxon-flask-api.git && cd saxon-flask-api
python3 -m venv env --without-pip
source env/bin/activate
curl https://bootstrap.pypa.io/get-pip.py | python
pip install -r requirements.txt
export FLASK_APP=app.py
export PYTHONPATH=~/.Downloads/Saxonica/SaxonHEC1.2.1/Saxon.C.API/python-saxon
flask run
```

#7 - 2021-03-19 14:27 - O'Neil Delpratt

Hi,

So I did not have the python virtual environment as you mentioned. I had to run:

```
sudo apt-get install python3.9-venv
```

#8 - 2021-03-19 14:34 - O'Neil Delpratt

So I have got down to the following command:

```
flask run
```

But getting this error:

```
No command 'flask' found, did you mean:
  Command 'flasm' from package 'flasm' (universe)
flask: command not found
```

#9 - 2021-03-19 14:42 - Kor Foo

Okay, I have made a change so the flask command is not used anymore. Can you pull the newest version? You should be able to now start the application using python app.py in your virtual environment.

```
$ python app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

#10 - 2021-03-19 17:57 - O'Neil Delpratt

Hi, I managed to get the server running.

When I run the command:

```
curl --request POST \
  --url http://127.0.0.1:5000/transform \
  --header 'Content-Type: application/xml' \
  --data '<?xml version="1.0"?>
<Article>
  <Title>My Article</Title>
  <Authors>
    <Author>Mr. Foo</Author>
    <Author>Mr. Bar</Author>
  </Authors>
  <Body>This is my article text.</Body>
</Article>'
```

I am getting the output:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<title>Redirecting...</title>
<h1>Redirecting...</h1>
<p>You should be redirected automatically to target URL: <a href="http://127.0.0.1:5000/transform/">http://127.0.0.1:5000/transform/</a>. If not click the link.
```

#11 - 2021-03-19 19:09 - Kor Foo

Okay, that is a good start :D Can you add -L to the curl command? It tells curl to automatically follow redirects:

```
$ curl -L --request POST \
  --url http://127.0.0.1:5000/transform \
  --header 'Content-Type: application/xml' \
  --data '<?xml version="1.0"?>
<Article>
  <Title>My Article</Title>
  <Authors>
    <Author>Mr. Foo</Author>
    <Author>Mr. Bar</Author>
  </Authors>
  <Body>This is my article text.</Body>
</Article>'

Article - My Article
Authors:
- Mr. Foo
- Mr. Bar%
```

#12 - 2021-03-19 19:50 - O'Neil Delpratt

Thanks.

I managed to run the command. I get back the results:

```
Body>This is my article text.</Body></Article>'
curl: (52) Empty reply from server
```

On the server I see:

```
127.0.0.1 - - [19/Mar/2021 18:45:09] "POST /transform HTTP/1.1" 308 -
Segmentation fault (core dumped)
```

I will investigate what is going wrong.

#13 - 2021-03-19 20:02 - O'Neil Delpratt

Please ignore my last message. On the first run I am now getting the result:

```
<Body>This is my article text.</Body></Article>'

Article - My Article
Authors:
- Mr. Foo
- Mr. Baron
```

When I run the same query I have managed to reproduce the error you reported from the server:

```
JNI_CreateJavaVM() failed with result: -5
```

I will investigate this further to see how we can work around this JVM loading issue.

#14 - 2021-03-20 00:47 - O'Neil Delpratt

Hi,

From what I see you are creating a PySaxonProcessor object each time a request is made. Is it possible to create the PySaxonProcessor once (i.e. globally) then reuse it between requests? I think this should help with the memory issue as the Jet vm should live between requests and only release when the server is closed.

Also what is the best way to monitor the growth of memory as requests are made?

#15 - 2021-03-22 22:31 - Kor Foo

Hello

In the first example I am indeed always making a new processor for each request. This way the resources should always get released.

I have made a branch (<https://github.com/RudolfDG/saxon-flask-api/tree/global-saxon-processor>) on the github repository with a global processor in the flask application, but as you can see (I updated the readme with the error message) this crashes after the first call.

I am now trying to produce an example in which I keep the processor alive and add a little script to replicate the memory issue.

Sorry for the delay :D

#16 - 2021-03-22 22:52 - Kor Foo

I have now also made a branch (<https://github.com/RudolfDG/saxon-flask-api/tree/keep-processor-in-memory>) in which the Jet VM is being kept alive and for each request there is a new saxonprocessor. This works fast and can process a lot of calls, but since it never releases the Jet VM, it keeps on adding extra memory.

Note that this does now run uwsgi as a webserver instead of the development server of Flask. You will have to do another pip install -r requirements.txt to install uwsgi and requests.

After the stresstest.py is completed the Jet VM does a big garbage collection, but this errors out and it posts the message you can see in the readme of the branch.

#17 - 2021-03-22 22:57 - Kor Foo

Basically there are 3 versions now:

1. Here we use the python context manager to release the resources used by the saxonprocessor. This has the effect that the Jet VM shuts down, but when a request comes in during the shutdown, the error -5 is thrown since it can't create a new one while shutting the other down.
2. Keep it as a global variable in the flask application. This breaks even on the first call since it can't initialize the JIT compiler thread.
3. Make a new PySaxonProcessor for each request, but never releasing at the end of a call. Using this method the Jet VM never shuts down and new calls go blazing fast. The problem with this one is that resources never get released and memory slowly builds up for each new call to the api. Finally crashing the application when the Jet VM tries to run his garbage collection. (In this minimal example the effect is less noticeable since the xslt and xml are both very small, but 20k requests seems to do the trick on my laptop, going up to 30% memory used on a 16gb laptop)

#18 - 2021-03-24 14:27 - O'Neil Delpratt

Hi,

I am investigating this issue again since your latest git changes. But I am having issue with

```
ImportError: No module named 'flask'
```

Is it possible you can remove dependency on flask on all branches? I have tried installing it.

#19 - 2021-03-24 15:00 - O'Neil Delpratt

Just seen your notes on uwsgi, Trying to setup it up now

#20 - 2021-03-24 15:06 - O'Neil Delpratt

Still having problems.

I ran:

```
uwsgi -i uwsgi.ini
```

But the server does not start. Getting the output:

```
detected binary path: /usr/bin/uwsgi-core
your processes number limit is 57781
your memory page size is 4096 bytes
detected max file descriptor number: 1024
lock engine: pthread robust mutexes
thunder lock: disabled (you can enable it with --thunder-lock)
The -s/--socket option is missing and stdin is not a socket.
```

#21 - 2021-03-24 15:07 - O'Neil Delpratt

The following command seems to start the server:

```
uwsgi -i -s --thunder-lock uwsgi.ini
```

#22 - 2021-03-24 15:25 - O'Neil Delpratt

On the server I get the following output:

```
*** Starting uWSGI 2.0.12-debian (64bit) on [Wed Mar 24 14:04:24 2021] ***
compiled with version: 5.4.0 20160609 on 28 September 2018 15:49:44
os: Linux-4.4.0-159-generic #187-Ubuntu SMP Thu Aug 1 16:28:06 UTC 2019
nodename: ondl-VirtualBox
machine: x86_64
clock source: unix
pcre jit disabled
detected number of CPU cores: 4
current working directory: /media/ondl/c8f4c380-ed5d-4791-b60d-90df7fd39798/work/Saxonica/SaxonEE1.2.1/Saxon.
C.API/python-saxon/saxon-flask-api
detected binary path: /usr/bin/uwsgi-core
*** WARNING: you are running uWSGI without its master process manager ***
your processes number limit is 57781
your memory page size is 4096 bytes
detected max file descriptor number: 1024
lock engine: pthread robust mutexes
thunder lock: disabled (you can enable it with --thunder-lock)
uwsgi socket 0 bound to UNIX address uwsgi.ini fd 3
your server socket listen backlog is limited to 100 connections
your mercy for graceful operations on workers is 60 seconds
mapped 72768 bytes (71 KB) for 1 cores
*** Operational MODE: single process ***
*** no app loaded. going in full dynamic mode ***
!!!!!!!!!!!!!! WARNING !!!!!!!!!!!!!!!
no request plugin is loaded, you will not be able to manage requests.
you may need to install the package for your language of choice, or simply load it with --plugin.
!!!!!!!!!!!!!! END OF WARNING !!!!!!!!!!!!!!!
spawned uWSGI worker 1 (and the only) (pid: 23798, cores: 1)
```

On the client. I run the command:

```
python3.7 stresstest.py
```

I get the following errors:

```
Traceback (most recent call last):
  File "/usr/lib/python3/dist-packages/urllib3/connection.py", line 136, in _new_conn
    conn = connection.create_connection(
  File "/usr/lib/python3/dist-packages/urllib3/util/connection.py", line 91, in create_connection
    raise err
  File "/usr/lib/python3/dist-packages/urllib3/util/connection.py", line 81, in create_connection
    sock.connect(sa)
ConnectionRefusedError: [Errno 111] Connection refused
```

During handling of the above exception, another exception occurred:

```
Traceback (most recent call last):
  File "/usr/lib/python3/dist-packages/urllib3/connectionpool.py", line 558, in urlopen
    httplib_response = self._make_request(conn, method, url,
  File "/usr/lib/python3/dist-packages/urllib3/connectionpool.py", line 354, in _make_request
    conn.request(method, url, **httplib_request_kw)
  File "/usr/lib/python3.9/http/client.py", line 1255, in request
    self._send_request(method, url, body, headers, encode_chunked)
  File "/usr/lib/python3.9/http/client.py", line 1301, in _send_request
    self.endheaders(body, encode_chunked=encode_chunked)
  File "/usr/lib/python3.9/http/client.py", line 1250, in endheaders
    self._send_output(message_body, encode_chunked=encode_chunked)
  File "/usr/lib/python3.9/http/client.py", line 1010, in _send_output
    self.send(msg)
  File "/usr/lib/python3.9/http/client.py", line 950, in send
    self.connect()
  File "/usr/lib/python3/dist-packages/urllib3/connection.py", line 162, in connect
    conn = self._new_conn()
  File "/usr/lib/python3/dist-packages/urllib3/connection.py", line 145, in _new_conn
    raise NewConnectionError(
requests.packages.urllib3.exceptions.NewConnectionError: <requests.packages.urllib3.connection.HTTPConnection
object at 0x7f8033d64c70>: Failed to establish a new connection: [Errno 111] Connection refused
```

During handling of the above exception, another exception occurred:

Traceback (most recent call last):

```
File "/usr/lib/python3/dist-packages/requests/adapters.py", line 366, in send
    resp = conn.urlopen(
File "/usr/lib/python3/dist-packages/urllib3/connectionpool.py", line 609, in urlopen
    retries = retries.increment(method, url, error=e, _pool=self,
File "/usr/lib/python3/dist-packages/urllib3/util/retry.py", line 283, in increment
    raise MaxRetryError(_pool, url, error or ResponseError(cause))
requests.packages.urllib3.exceptions.MaxRetryError: HTTPConnectionPool(host='localhost', port=5000): Max retries
exceeded with url: /transform/ (Caused by NewConnectionError('<requests.packages.urllib3.connection.HTTPCon
nection object at 0x7f8033d64c70>: Failed to establish a new connection: [Errno 111] Connection refused'))
```

During handling of the above exception, another exception occurred:

Traceback (most recent call last):

```
File "/media/ondl/c8f4c380-ed5d-4791-b60d-90df7fd39798/work/Saxonica/SaxonEEC1.2.1/Saxon.C.API/python-saxon/
saxon-flask-api/stresstest.py", line 21, in <module>
    response = requests.request("POST", url, headers=headers, data=payload)
File "/usr/lib/python3/dist-packages/requests/api.py", line 53, in request
    return session.request(method=method, url=url, **kwargs)
File "/usr/lib/python3/dist-packages/requests/sessions.py", line 480, in request
    resp = self.send(prepared_request, **send_kwargs)
File "/usr/lib/python3/dist-packages/requests/sessions.py", line 588, in send
    r = adapter.send(request, **kwargs)
File "/usr/lib/python3/dist-packages/requests/adapters.py", line 437, in send
    raise ConnectionError(e, request=request)
requests.exceptions.ConnectionError: HTTPConnectionPool(host='localhost', port=5000): Max retries exceeded wit
h url: /transform/ (Caused by NewConnectionError('<requests.packages.urllib3.connection.HTTPConnection object
at 0x7f8033d64c70>: Failed to establish a new connection: [Errno 111] Connection refused'))
```

Not sure how to proceed.

#23 - 2021-03-24 15:45 - Kor Foo

Hello, do you perhaps have docker installed? I could make a Dockerfile for each branch and that way we can eliminate the installation problems :D

#24 - 2021-03-24 15:52 - O'Neil Delpratt

Yes that would work for me. I do have docker installed.

#25 - 2021-03-25 11:35 - Kor Foo

All three branches now have a Dockerfile and can be built/ran using:

```
docker build -t saxon-flask-api .
docker run -p 5000:5000 saxon-flask-api:latest
```

The port is still 5000 and the ip should be unchanged, so the same curl calls can be used to test.

#26 - 2021-03-26 14:27 - O'Neil Delpratt

Hi,

Thanks for your post on docker. I managed to run the sever from docker and use curl to reproduce the errors you reported.

But how do I go about experimenting and debugging the code. Is there a docker command to access the container and edit the Saxon/C python files and the app.py files please?

#27 - 2021-03-29 13:40 - Kor Foo

Hello, you can always run `docker run -it -p 5000:5000 --entrypoint /bin/bash saxon-flask-api` to start a bash shell in a new container. In the container you can edit app.py or other files and run the application like you would locally.

You can also just edit the python files and rebuild the docker container before running it. To edit the saxon files I would have to include the source in the library as currently the Dockerfile just downloads the .zip from saxonica and installs it.

#28 - 2021-03-29 17:03 - O'Neil Delpratt

Thanks for the prompt on the docker command. I ran the server on docker and reproduced the error you reported. My focus is on the git branch global-saxon-processor. I suspect the PySaxonProcessor is not created in the cache. I tried to tweak it a bit by creating the PySaxonProcessor object once in the transform_xml function.

But this resulted in the following crash on the server:

```
172.17.0.1 - - [29/Mar/2021 14:49:26] "POST /transform/ HTTP/1.0" 200 -
```

```
JET RUNTIME HAS DETECTED UNRECOVERABLE ERROR: runtime error
Thread 1F ["main"] is terminated without notifying the JVM. Probably, "DetachCurrentThread" function was not called
Please, contact the vendor of the application.
Core dump will be written to "/app/core"
Extra information about error is saved in the "jet_err_28.txt" file.
```

I just don't know enough of this python/flask server setup to understand what is happening. Nevertheless, let see if we can get to the bottom of this issue. On the server I did noticed the following statement:

```
Serving Flask app "app2" (lazy loading)
```

Is there any way to disable this?

#29 - 2021-03-29 17:34 - Kor Foo

```
docker run -it -p 5000:5000 --entrypoint /bin/bash saxon-flask-api
```

```
appuser@4d7562e109fb:/app$ python3 app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
172.17.0.1 - - [29/Mar/2021 15:30:40] "POST /transform HTTP/1.1" 308 -
Error: failed to allocate an object
[JR_lazyJITCodeGen @ 1]: Failed to initialize lazy JIT compiler thread
```

```
JET RUNTIME HAS DETECTED UNRECOVERABLE ERROR: system exception at 0x00007f60a3011d49
Please, contact the vendor of the application.
Core dump will be piped to "/usr/share/apport/apport %p %s %c %d %P %E"
Extra information about error is saved in the "jet_err_6.txt" file.
```

```
Aborted (core dumped)
```

Hmm, when do you get the "app2"? The above is what I get after building the image and using the command.

And what do you want to disable? Just the "Serving Flask" text?

#30 - 2021-03-29 17:41 - O'Neil Delpratt

Sorry, I created new version of app.py I call app2.py.

Still experimenting.

app2.py:

```
from flask import Flask, abort, jsonify, request
import saxonc
import os
app = Flask(__name__)

cache = {
    "processor": None
}
cache1 = {
    "check":1
}

@app.route("/transform/", methods=["POST"])
def transform():
    data = request.data

    result = transform_xml(data)

    return result

def transform_xml(xml: bytes) -> str:

    if cache1["check"] == 1:
        print("cp0")
        cache1["check"] = 0
```



```

processor = saxonc.PySaxonProcessor(license=False)
cache["processor"] = processor
elif cache["processor"] != None:
    print("cpl")
    processor = cache["processor"]

base_dir = os.getcwd()
xslt_path = os.path.join(base_dir, "test.xslt")

xslt_proc = processor.new_xslt30_processor()

node = processor.parse_xml(xml_text=xml.decode("utf-8"))

result = xslt_proc.transform_to_string(stylesheets_file=xslt_path, xdm_node=node)

return result

if __name__ == "__main__":
    app.run(host="0.0.0.0")

```

#31 - 2021-03-29 17:46 - O'Neil Delpratt

According to https://bugs.java.com/bugdatabase/view_bug.do?bug_id=8201235 the error:

```
[JR_lazyJITCodeGen @ 1]: Failed to initialize lazy JIT compiler thread
```

seems to suggest we are creating a large number of compiler threads. It is still not clear to me how this is done on the server side.

#32 - 2021-03-30 05:24 - Daniel Haley

- File *try_saxon2.zip* added

I'm also running into this issue when trying to create a processor more than once. In this case, from two different functions.

I've attached a zip with a modified Python script from <https://saxonica.plan.io/boards/4/topics/8157>

#33 - 2021-03-30 09:11 - O'Neil Delpratt

Only one PySaxonProcessor object should be created.

Also I recommend not to use the processor as a context if you are using Saxon in a server setting where you are keeping alive the app. This is because the `release()` function will be called preventing it to be used again in that process.

Files

File Name	Size	Created	Author
try_saxon2.zip	1.52 KB	2021-03-30	Daniel Haley