

+ 23

Powered by AI: A Cambrian Explosion for C++ Software Development Tools

EMERY BERGER

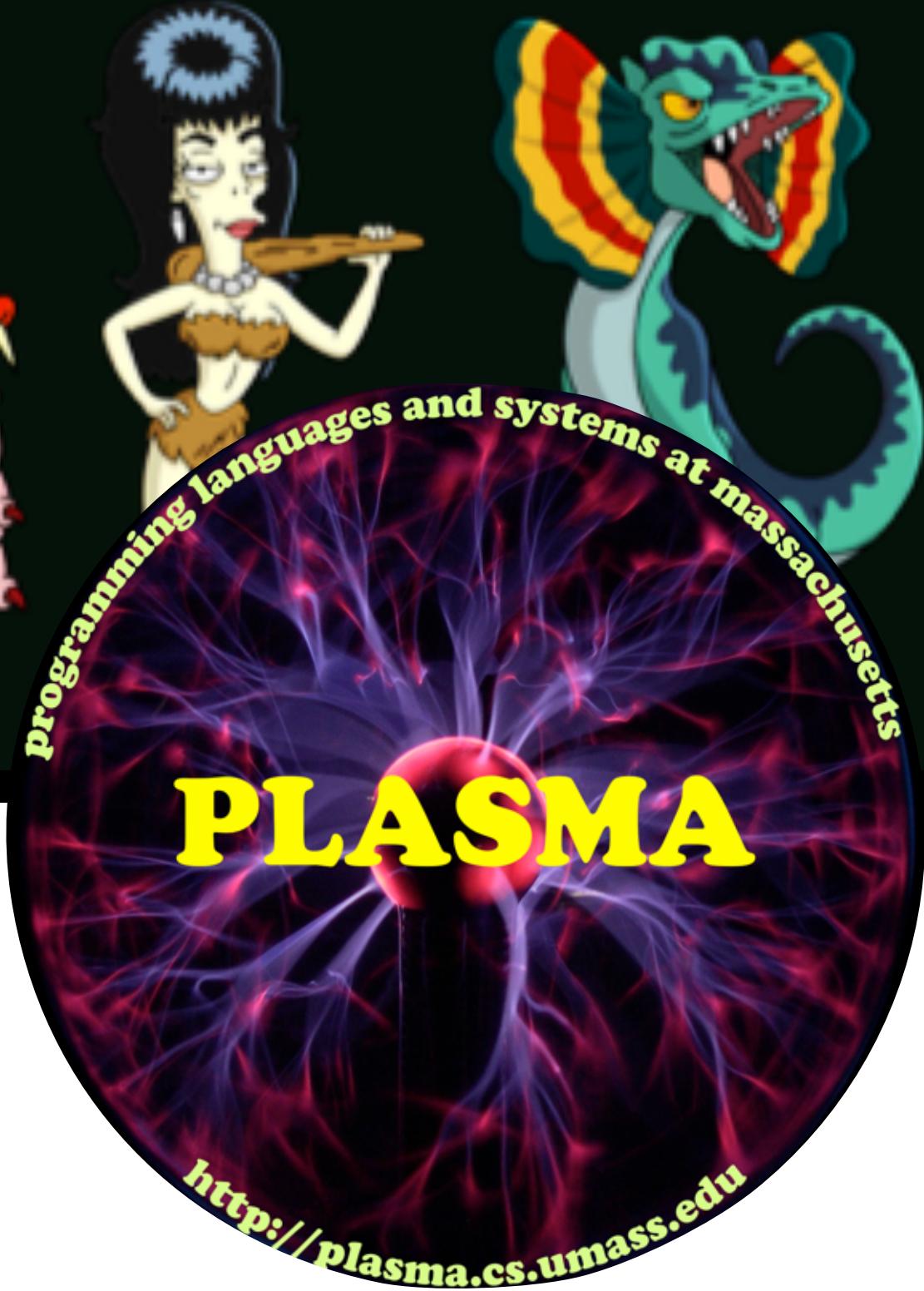


Cppcon
The C++ Conference

20
23



October 01 - 06



Powered by AI: A Cambrian Explosion for C++ Software Development Tools

Emery Berger

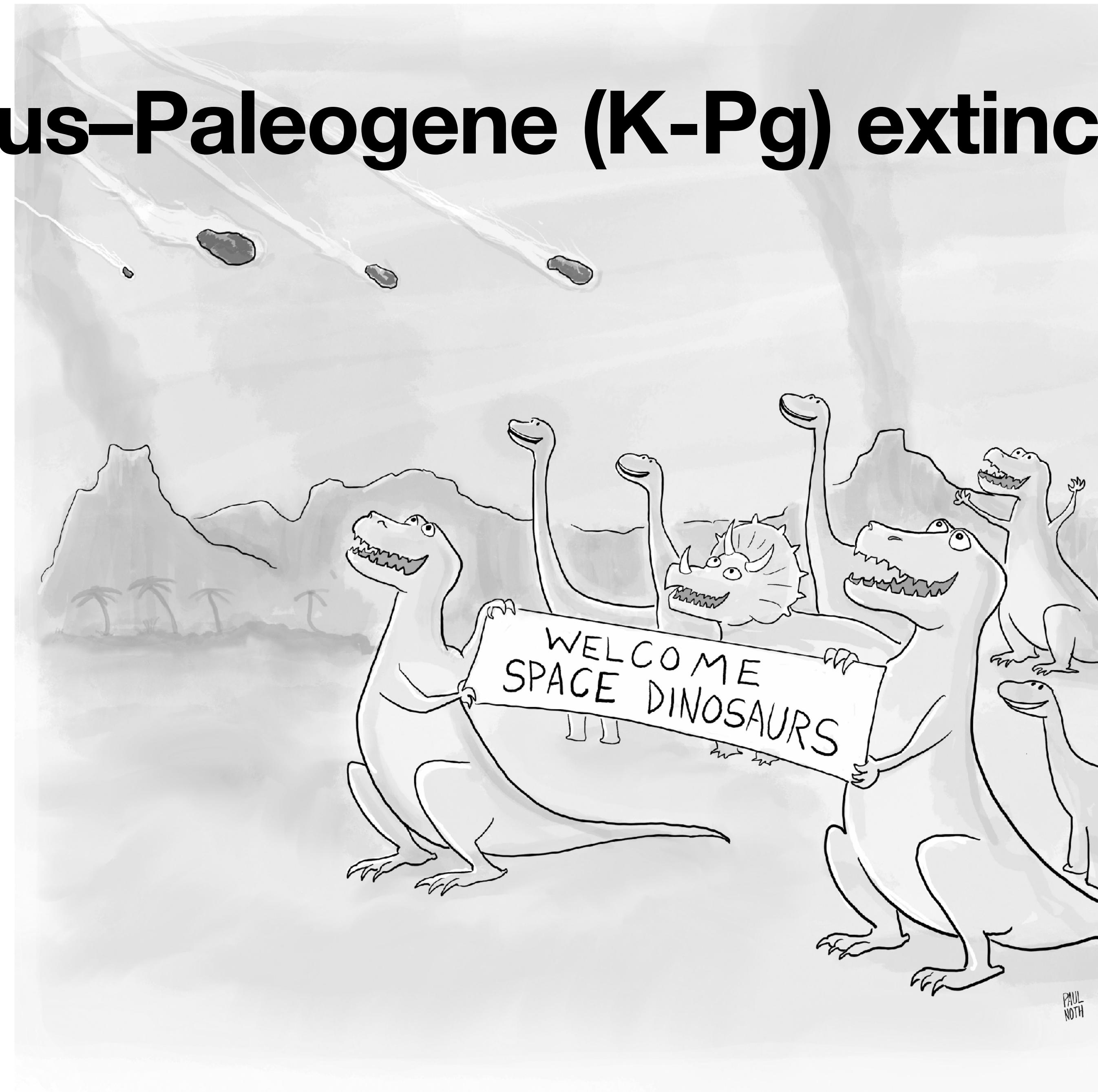
University of Massachusetts Amherst

Cretaceous–Paleogene (K-Pg) extinction event

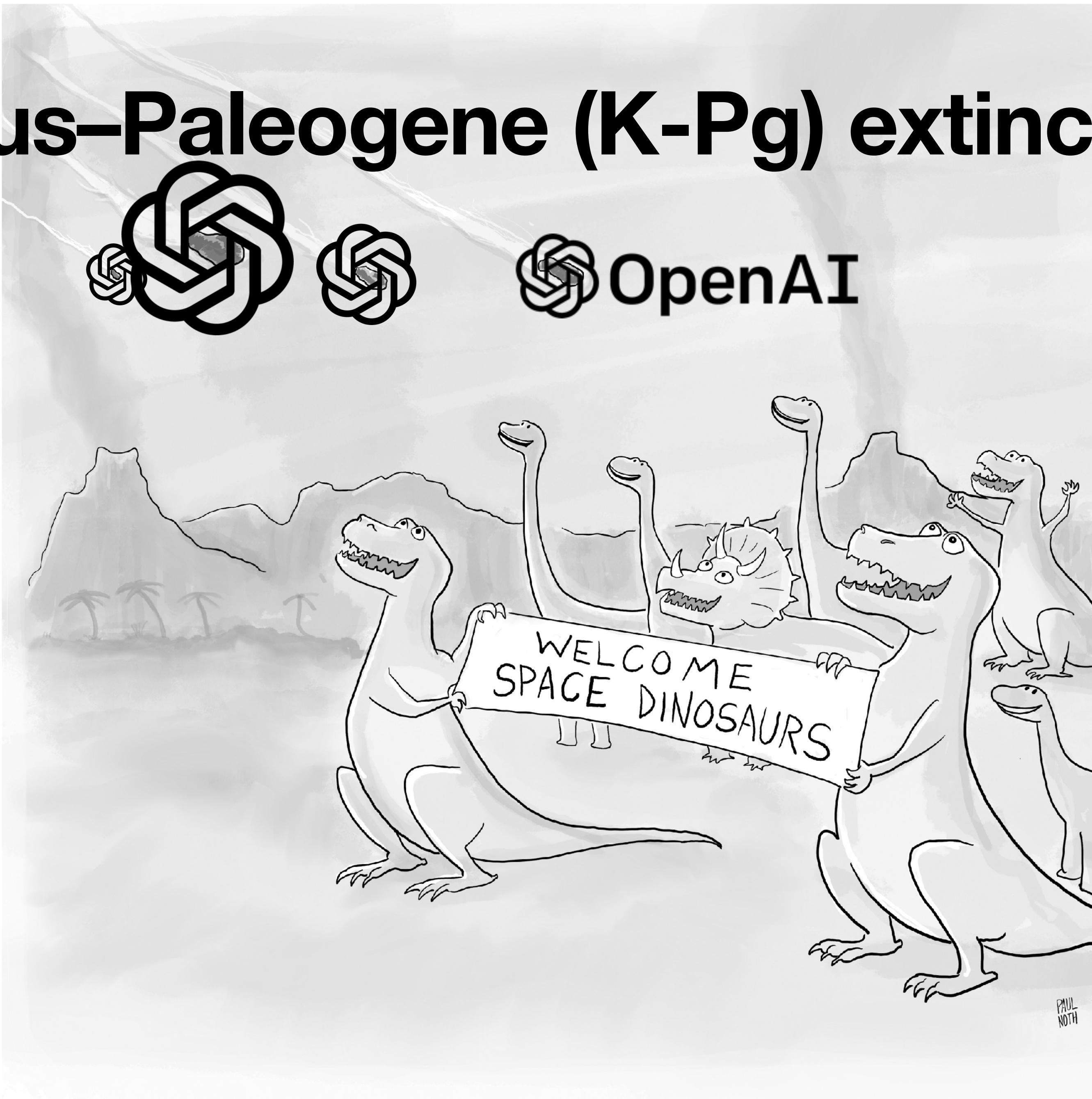
Cretaceous–Paleogene (K-Pg) extinction event



Cretaceous–Paleogene (K-Pg) extinction event

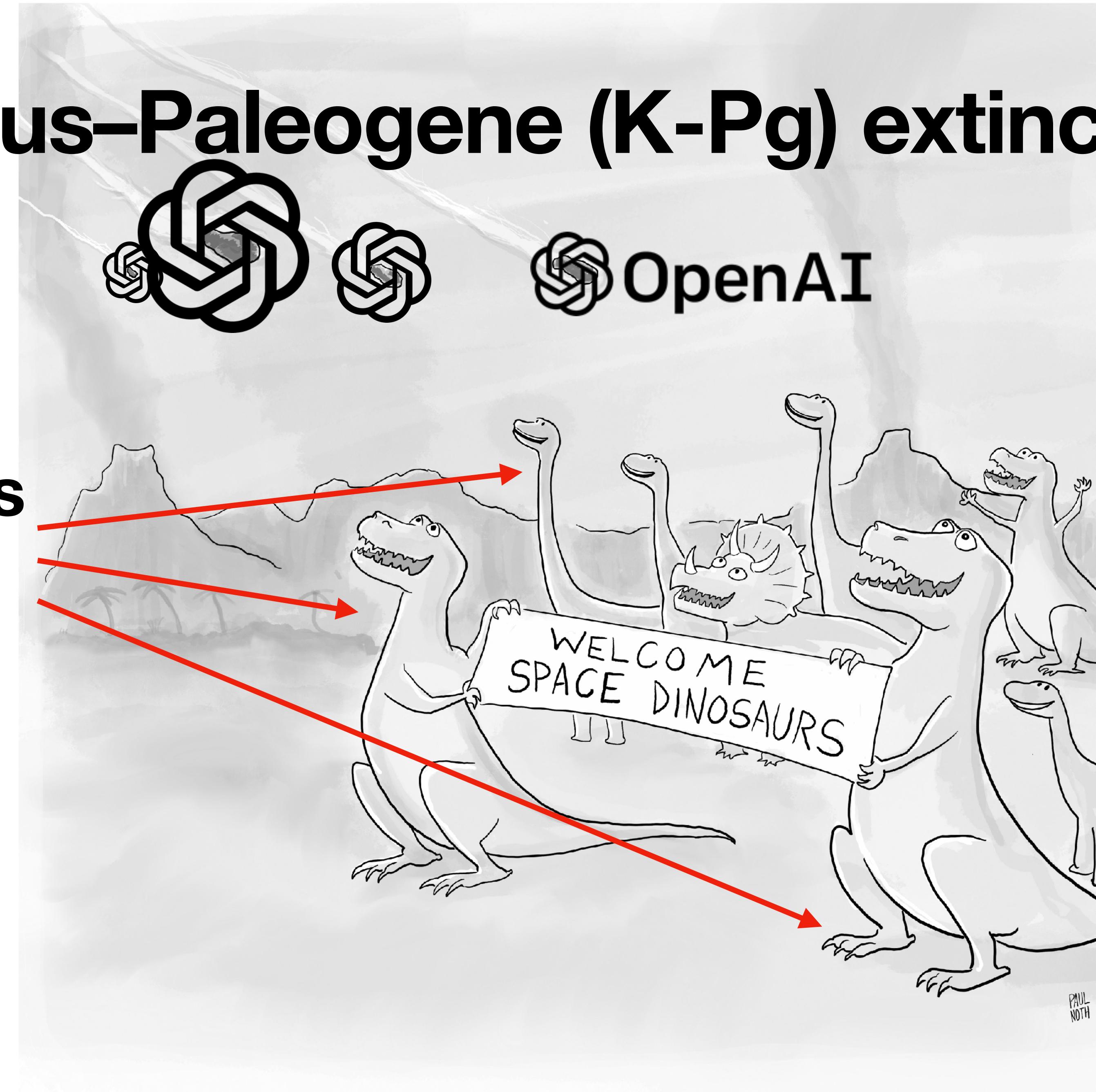


Cretaceous–Paleogene (K-Pg) extinction event



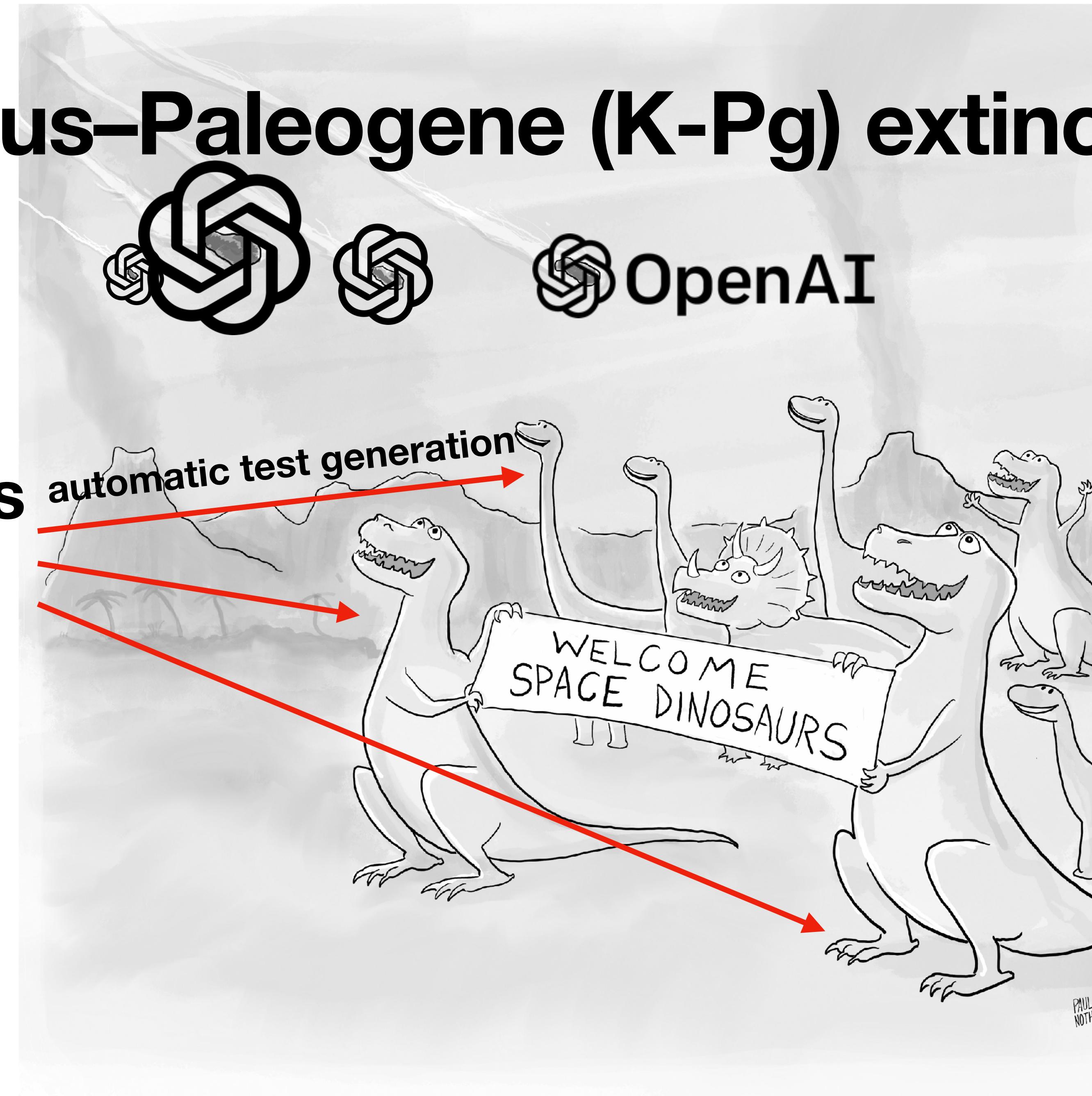
Cretaceous–Paleogene (K-Pg) extinction event

whole lines
of PL/SE
research



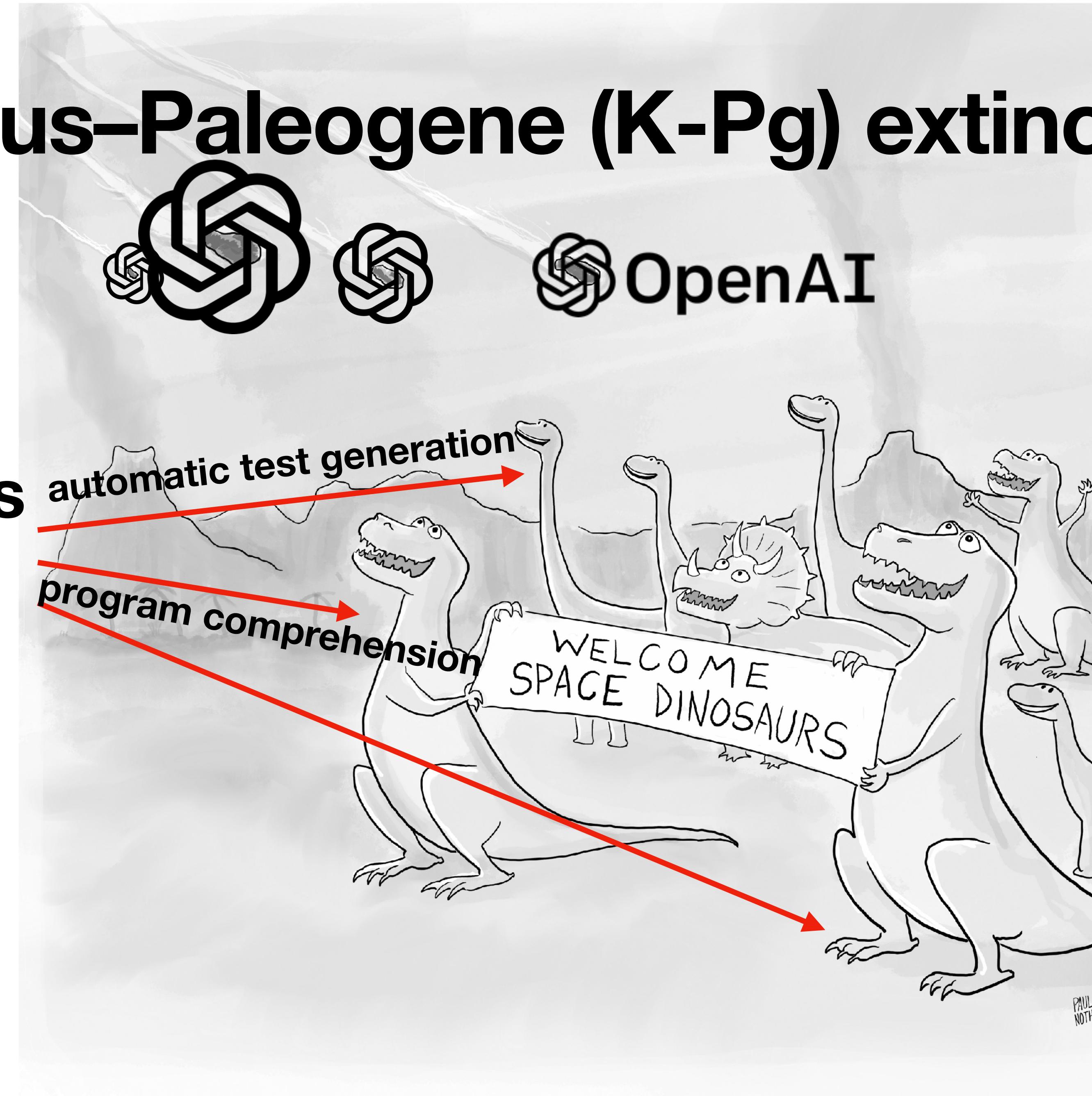
Cretaceous–Paleogene (K-Pg) extinction event

whole lines
of PL/SE
research



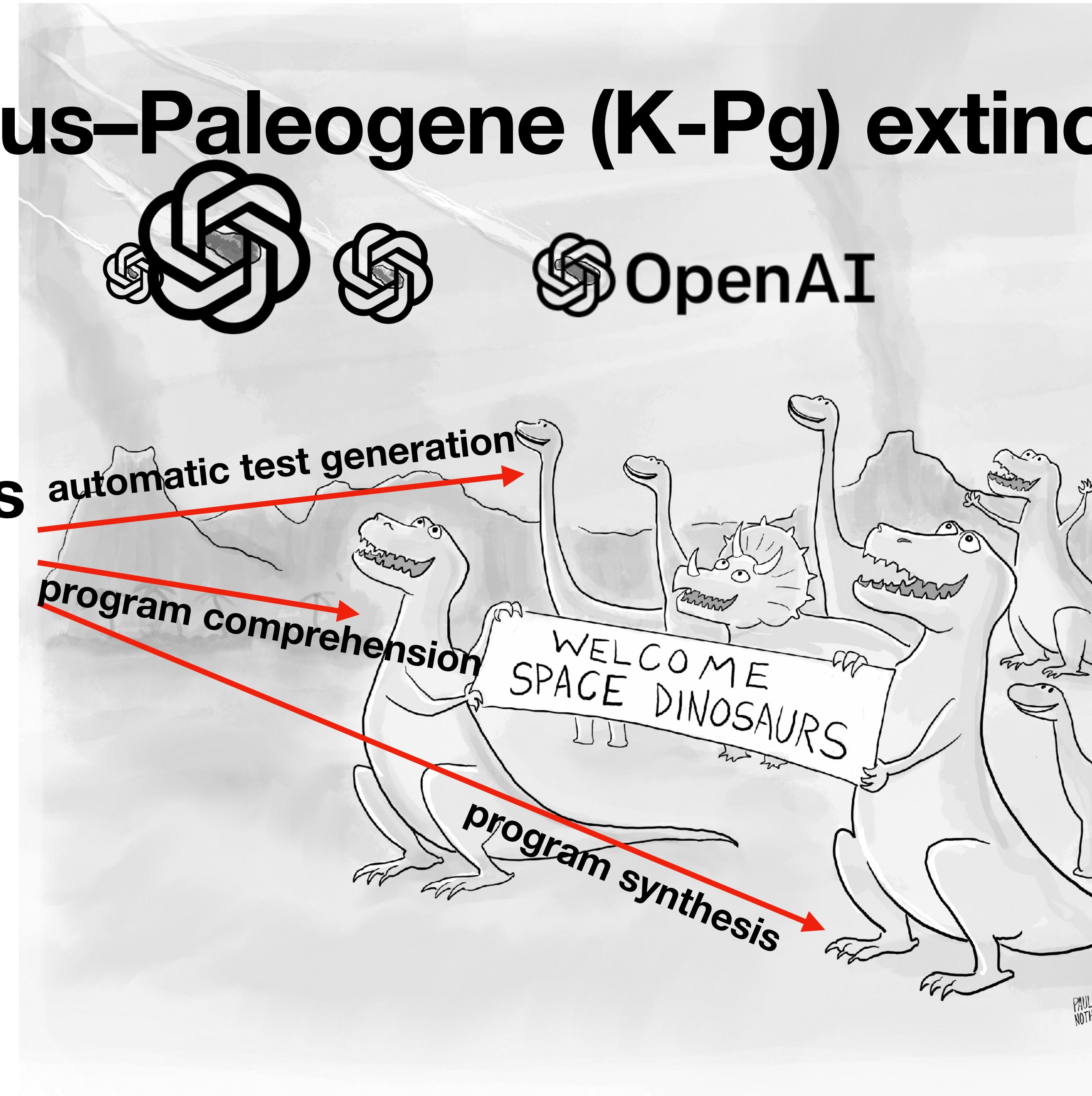
Cretaceous–Paleogene (K-Pg) extinction event

whole lines
of PL/SE
research



Cretaceous–Paleogene (K-Pg) extinction event

whole lines
of PL/SE
research



[OOPSLA 2022]

[OOPSLA 2022]

```
1  public List<String> getUserRoles( String uID , List<Policy> policies ) {  
2      List<String> roles = new ArrayList<>();  
3      for ( Policy policy : policies ) {  
4          for ( Role role : policy.getRoles() ) {  
5              if ( role.getIds().contains(uID) )  
6                  roles . add( role.getName() );  
7          }  
8      }  
9      return roles;  
10 }
```

Fig. 2. Imperative Java Program

[OOPSLA 2022]

```
1  public List<String> getUserRoles(String uID, List<Policy> policies) {  
2      List<String> roles = new ArrayList<>();  
3      for (Policy policy : policies) {  
4          for (Role role : policy.getRoles() ) {  
5              if ( role.getIds().contains(uID) )  
6                  roles.add( role.getName() );  
7          }  
8      }  
9      return roles;  
10 }
```

Fig. 2. Imperative Java Program

```
1  public List<String> getUserRoles(String uID, List<Policy> policies) {  
2      return policies.stream()  
3          .flatMap(policy -> policy.getRoles() .stream())  
4          .filter(role -> role.getIds().contains(uID))  
5          .map(role -> role.getName())  
6          .collect(Collectors.toList());  
7  }
```

Fig. 3. Java Stream Program

[OOPSLA 2022]

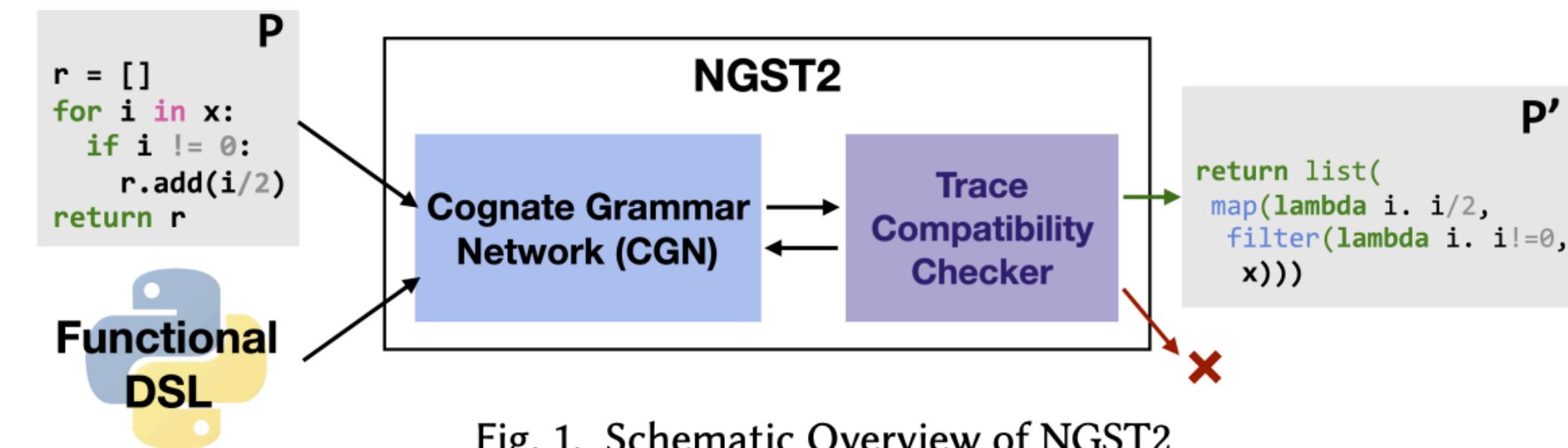


Fig. 1. Schematic Overview of NGST2

```

1: procedure TRANSPILE( $\mathcal{P}, \mathcal{M}_\theta, \mathcal{G}$ )
2:   input: Source program  $\mathcal{P}$ 
3:   input: Neural model  $\mathcal{M}_\theta$ 
4:   input: Context-free grammar  $\mathcal{G} = (V, \Sigma, R, S)$ 
5:    $\mathcal{W} \leftarrow \{S\}$                                 ▷ initialize worklist to  $S$ , the empty partial program
6:    $\mathcal{E} \leftarrow \emptyset$                             ▷ initialize counterexamples to empty set
7:   while  $\mathcal{W} \neq \emptyset$  do
8:      $\mathcal{P}' \leftarrow \text{CHOOSEBEST}(\mathcal{W}, \mathcal{M}_\theta, \mathcal{P})$       ▷ Dequeue top candidate from priority queue
9:     if IsComplete( $\mathcal{P}'$ ) then
10:       $(v, \sigma) \leftarrow \text{IsEquivalent}(\mathcal{P}, \mathcal{P}', \mathcal{E})$ 
11:      if  $v$  then                                         ▷ Candidate is equivalent
12:        return  $\mathcal{P}'$ 
13:         $\mathcal{E} \leftarrow \mathcal{E} \cup \{\sigma\}$                       ▷ Add new counterexample
14:        continue
15:      if  $\neg \text{IsFEASIBLE}(\mathcal{P}', \mathcal{P}, \mathcal{E})$  then    ▷ Check feasibility of partial program
16:        continue
17:       $N \leftarrow \text{ChooseNonterminal}(\mathcal{P}')$ 
18:      for  $r \in \text{Productions}(N)$  do
19:         $\mathcal{W} \leftarrow \mathcal{W} \cup \{\text{Expand}(\mathcal{P}', r)\}$           ▷ Add expansions of  $\mathcal{P}'$  to worklist
20:    return  $\perp$ 

```

[OOPSLA 2022]

◀ 2 / 2 ▶ 

Convert this Java program to use streams.



```
```
public List<String> getUserRoles(String uID, List<Policy> policies) {
 List<String> roles = new ArrayList<>();
 for (Policy policy : policies) {
 for (Role role : policy.getRoles()) {
 if (role.getIDs().contains(uID))
 roles.add(role.getName());
 }
 }
 return roles;
}
````
```



To convert the given Java program to use streams, you can follow the steps below:



```
java
Copy code

import java.util.List;
import java.util.stream.Collectors;

public List<String> getUserRoles(String uID, List<Policy> policies) {
    return policies.stream()                                // Convert the policies
                    .flatMap(policy -> policy.getRoles().stream())   // Flatten
                    .filter(role -> role.getIDs().contains(uID))     // Filter roles
                    .map(Role::getName)                            // Map the names
                    .collect(Collectors.toList());                  // Collect the results
}
```

[OOPSLA 2022]

< 2 / 2 > 

Convert this Java program to use streams.



```
```  
public List<String> getUserRoles(String uID, List<Policy> policies) {
 List<String> roles = new ArrayList<>();
 for (Policy policy : policies) {
 for (Role role : policy.getRoles()) {
 if (role.getIDs().contains(uID))
 roles.add(role.getName());
 }
 }
 return roles;
}
```
```



To convert the given Java program to use streams, you can follow the steps below:



java

Copy code

```
import java.util.List;  
import java.util.stream.Collectors;  
  
public List<String> getUserRoles(String uID, List<Policy> policies) {  
    return policies.stream()  
        .flatMap(policy -> policy.getRoles().stream())  
        .filter(role -> role.getIDs().contains(uID))  
        .map(Role::getName)  
        .collect(Collectors.toList());  
}
```

1 public List <String > getUserRoles (String uID , List <Po
2 return policies . stream ()
3 . flatMap (policy -> policy . getRoles () . stream ()
4 . filter (role -> role . getIDs () . contains (uID))
5 . map (role -> role . getName ())
6 . collect (Collectors . toList ()) ;
7 }

Fig. 3. Java Stream Program

[OOPSLA 2022]

< 2 / 2 >

Convert this Java program to use streams.



```
```
public List<String> getUserRoles(String uID, List<Policy> policies) {
 List<String> roles = new ArrayList<>();
 for (Policy policy : policies) {
 for (Role role : policy.getRoles()) {
 if (role.getIDs().contains(uID))
 roles.add(role.getName());
 }
 }
 return roles;
}
````
```



To convert the

code below:



java

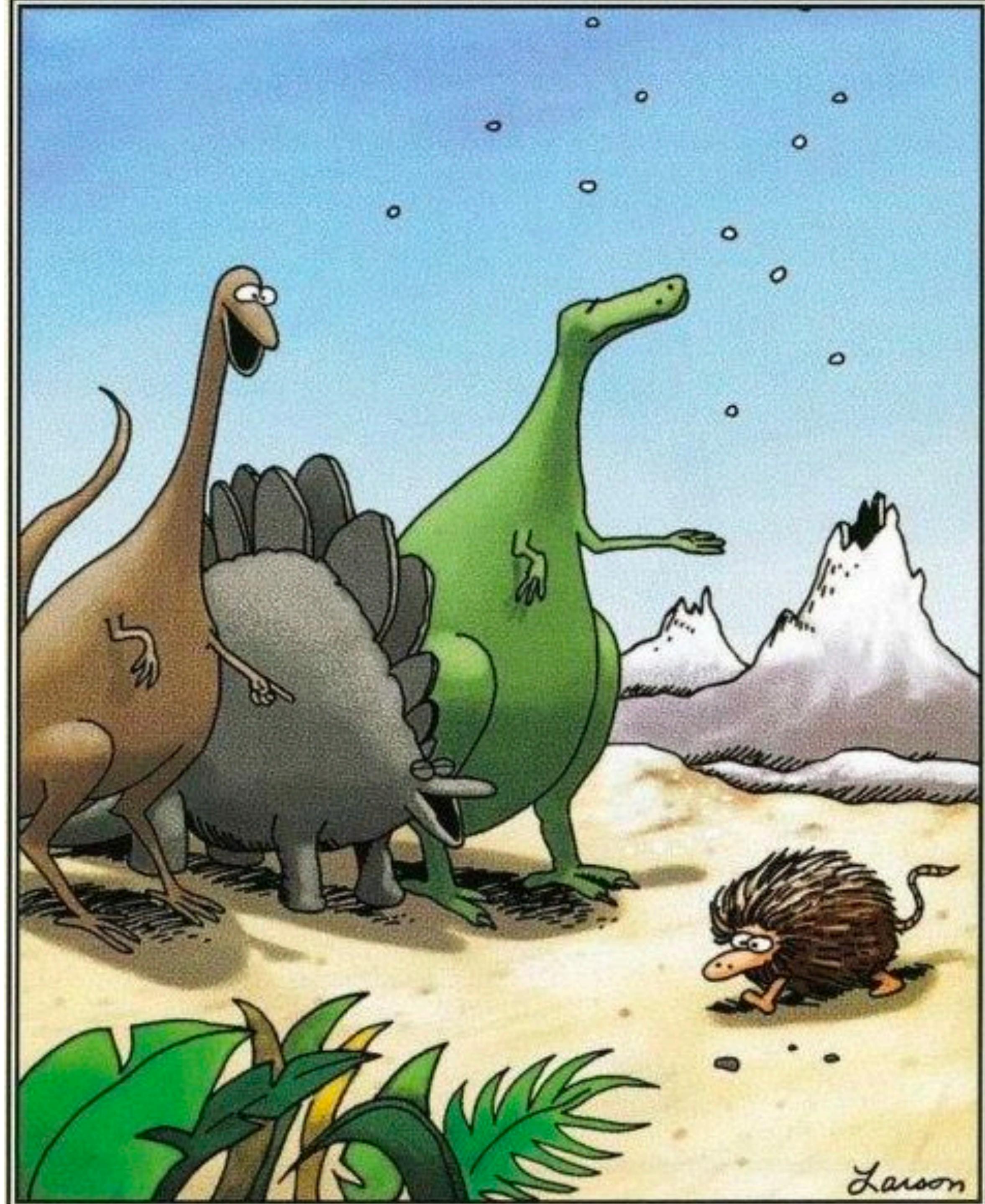
Copy code

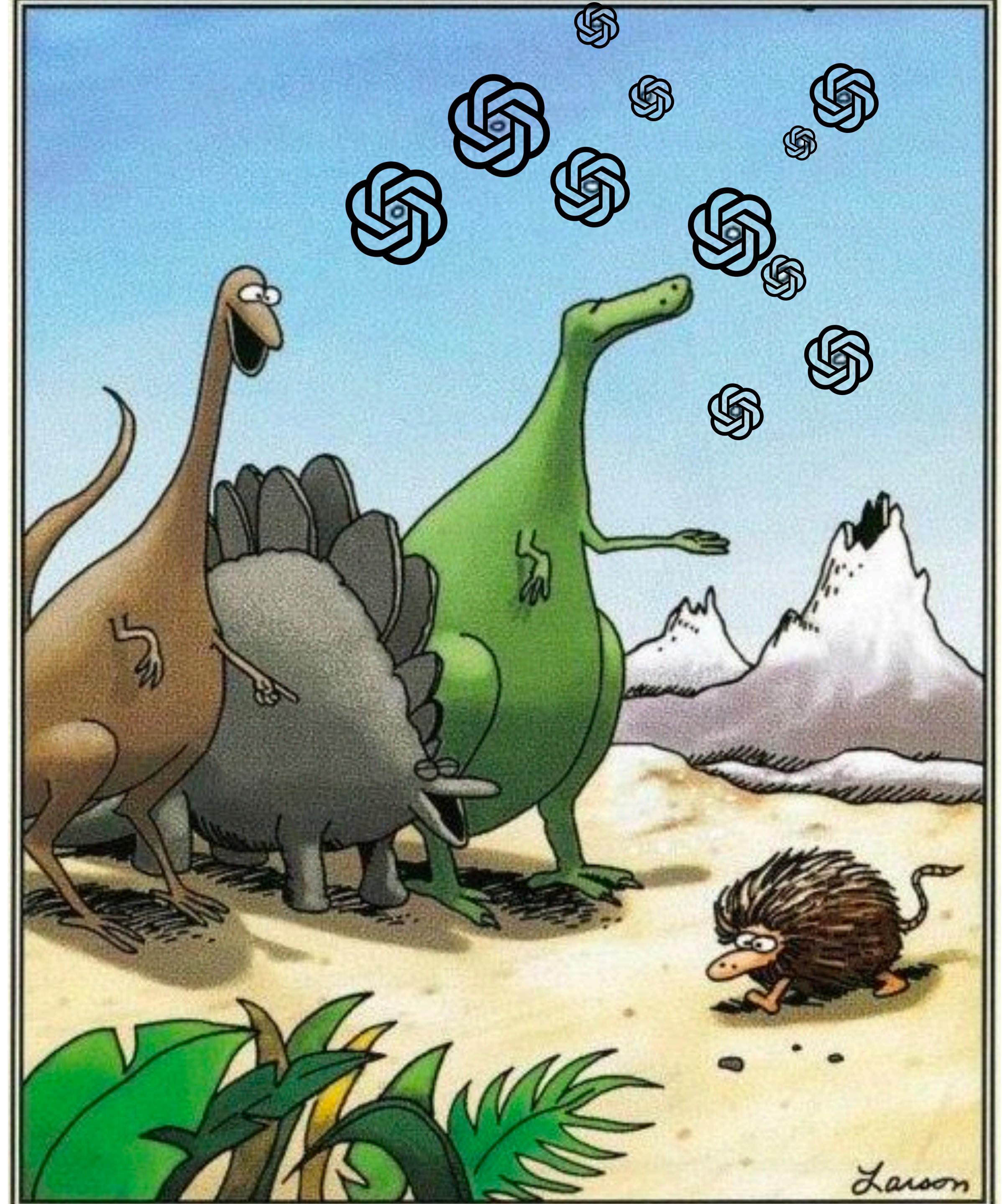
```
import java.util.List;
import java.util.stream.Collectors;

public List<String> getUserRoles(String uID, List<Policy> policies) {
    return policies.stream()                                // Convert the policies
        .flatMap(policy -> policy.getRoles().stream())   // Flatten the list of roles
        .filter(role -> role.getIDs().contains(uID))      // Filter the roles
        .map(Role::getName)                               // Map the roles to their names
        .collect(Collectors.toList());                    // Collect the results
}
```

```
1  public List<String> getUserRoles(String uID, List<Policy> policies) {
2      return policies.stream()
3          .flatMap(policy -> policy.getRoles().stream())
4          .filter(role -> role.getIDs().contains(uID))
5          .map(role -> role.getName())
6          .collect(Collectors.toList());
7  }
```

Fig. 3. Java Stream Program





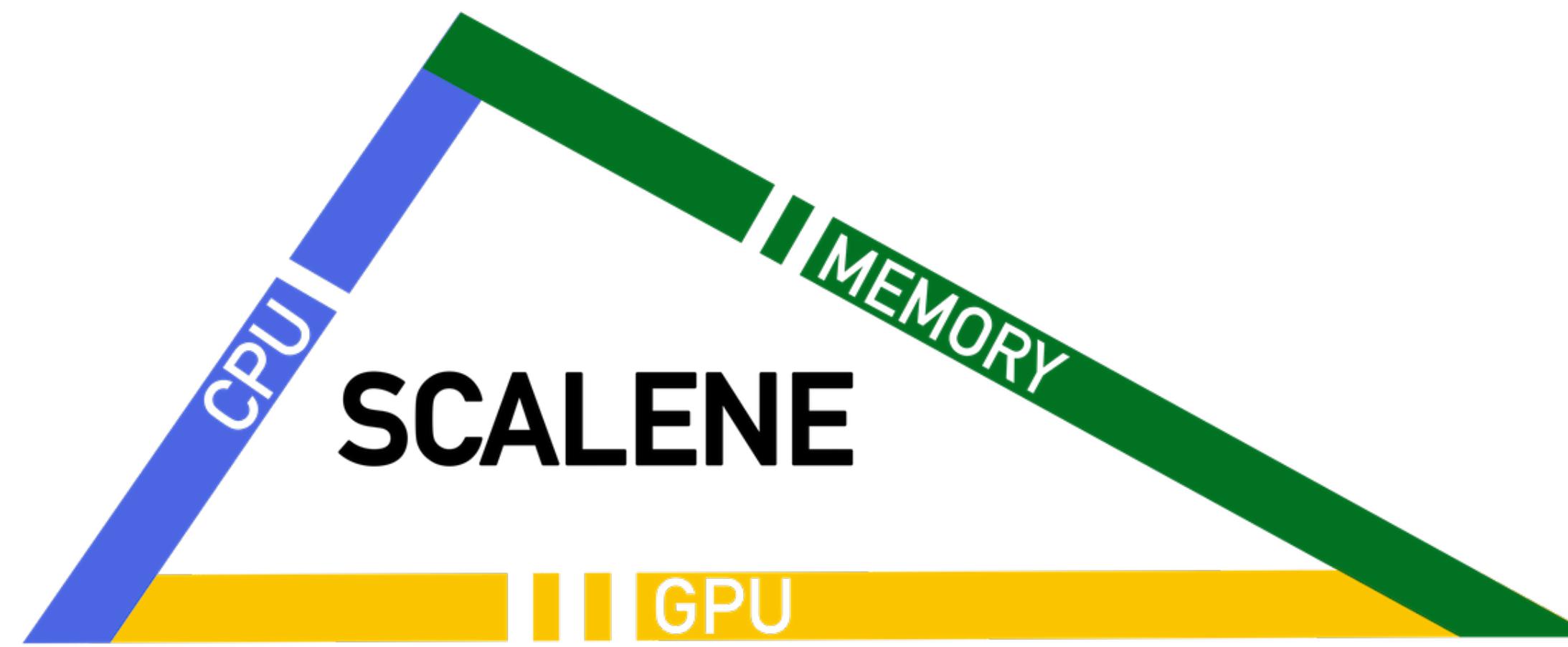
Larson



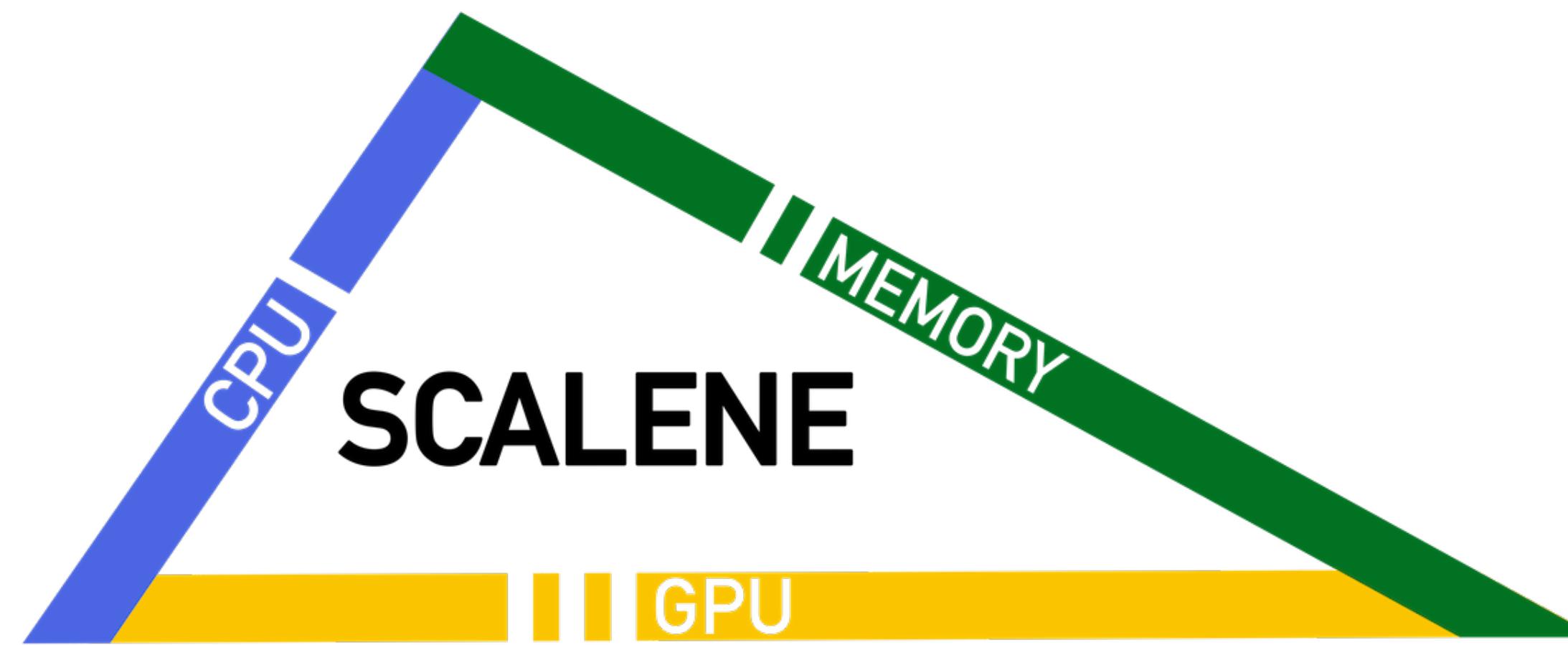
Larson



Larson



| | | | | |
|------------|-------------------|-----------------------|-------------|------------|
| <u>CPU</u> | <u>MEMORY</u> | <u>MEMORY</u> | <u>COPY</u> | <u>GPU</u> |
| PYTHON | PYTHON | USAGE | VOLUME | UTIL %, |
| NATIVE | NATIVE | OVER TIME, | (MB/s) | PEAK |
| SYS% | AVERAGE & PEAK | % OF MEM ALLOCATED | | MEMORY |



CPU
PYTHON
NATIVE
SYS%

MEMORY
PYTHON
NATIVE
AVERAGE &
PEAK

MEMORY COPY GPU
USAGE VOLUME UTIL %,
OVER TIME,
% OF MEM
ALLOCATED
(MB/s) PEAK
MEMORY

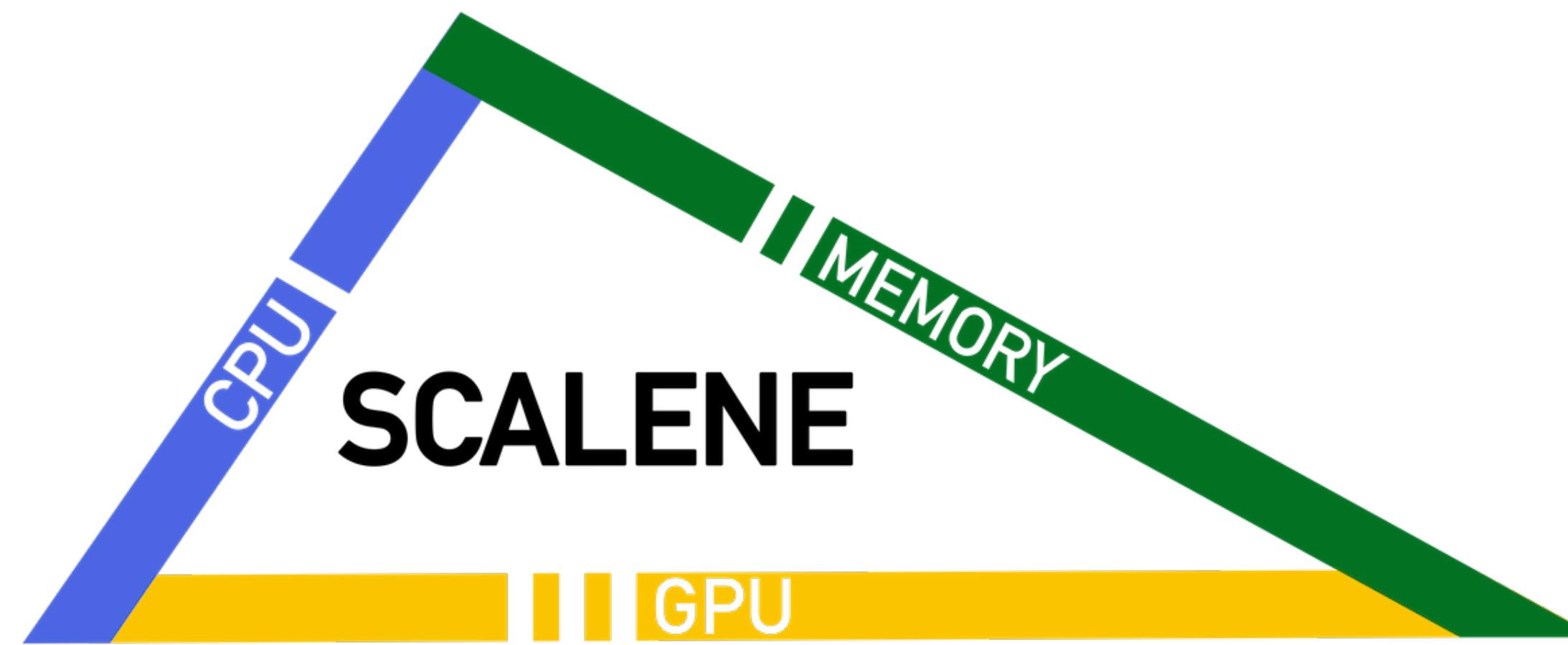
OSDI²³ ATTEND PROGRAM PARTICIPATE SPONSORS ABOUT

Triangulating Python Performance Issues with SCALENE

Authors:

Emery D. Berger, Sam Stern, and Juan Altmayer Pizzorno, *University of Massachusetts Amherst*

Awarded Best Paper!



CPU
PYTHON
NATIVE
SYS%

MEMORY
PYTHON
NATIVE
AVERAGE &
PEAK

MEMORY COPY GPU
USAGE VOLUME UTIL %,
OVER TIME, (MB/s) PEAK
% OF MEM MEMORY
ALLOCATED



ATTEND

PROGRAM

PARTICIPATE

SPONSORS

ABOUT

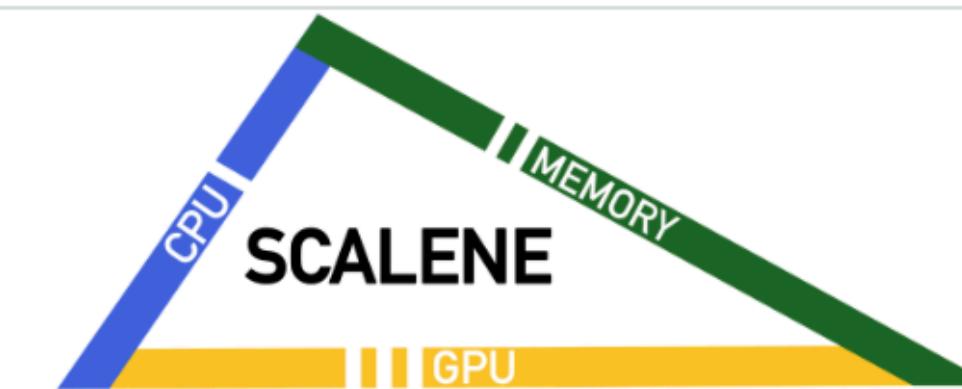
downloads 830k

Triangulating Python Performance Issues with SCALENE

Authors:

Emery D. Berger, Sam Stern, and Juan Altmayer Pizzorno, *University of Massachusetts Amherst*

Awarded Best Paper!



Select a profile (.json)

► advanced options

Time: Python | native | system

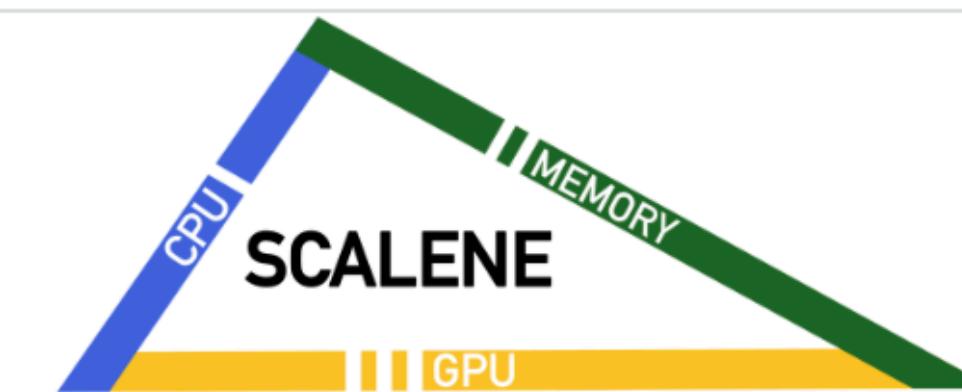
Memory: Python | native

Memory timeline: (max: 1.653 GB, growth: 21.0%)

hover over bars to see breakdowns; click on COLUMN HEADERS to sort.show all | hide all | only display profiled lines

▼test/issues/test-issue31.py: % of time = 100.0% (2.897s) out of 2.897s.

| TIME | MEMORY average | MEMORY peak | MEMORY timeline | MEMORY activity | COPY | LINE PROFILE (click to reset order) |
|------|-------------------|----------------|--------------------|--------------------|---|-------------------------------------|
| | | | | | 17 | 1 ⚡ import numpy as np |
| | | | | | 3 ⚡ def main1(): | |
| | | | | | 5 ⚡ x = np.array(range(10**7)) | |
| | | | | | 272 ⚡ y = np.array(np.random.uniform(0, 100, size=10**8)) | |
| | | | | | 8 ⚡ def main2(): | |
| | | | | | 10 ⚡ x = np.array(range(10**7)) | |



Select a profile (.json)

► advanced options

Time: Python | native | system

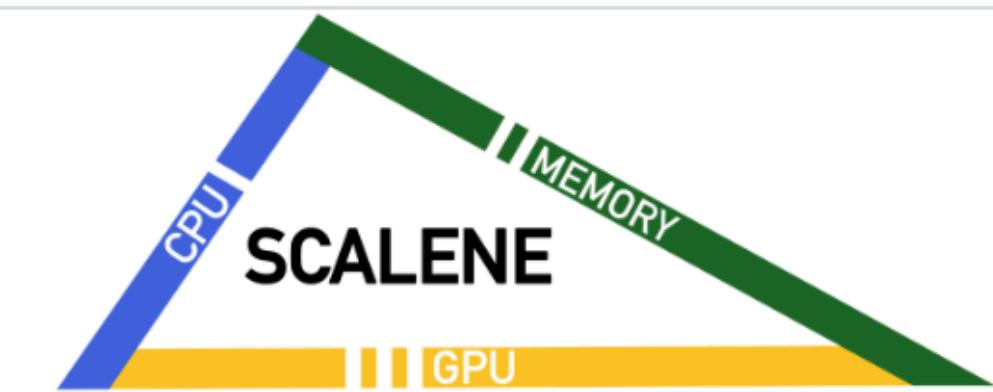
Memory: Python | native

Memory timeline: (max: 1.653 GB, growth: 21.0%)

hover over bars to see breakdowns; click on COLUMN HEADERS to sort.show all | hide all | only display profiled lines

▼test/issues/test-issue31.py: % of time = 100.0% (2.897s) out of 2.897s.

| TIME | MEMORY average | MEMORY peak | MEMORY timeline | MEMORY activity | COPY | LINE PROFILE (click to reset order) |
|------|-------------------|----------------|--------------------|--------------------|---|-------------------------------------|
| | | | | | 17 | 1 ⚡ import numpy as np |
| | | | | | 3 ⚡ def main1(): | |
| | | | | | 5 ⚡ x = np.array(range(10**7)) | |
| | | | | | 272 ⚡ y = np.array(np.random.uniform(0, 100, size=10**8)) | |
| | | | | | 8 ⚡ def main2(): | |
| | | | | | 10 ⚡ x = np.array(range(10**7)) | |



Select a profile (.json)

► advanced options

Time: Python | native | system

Memory: Python | native

Memory timeline: (max: 1.653 GB, growth: 21.0%)

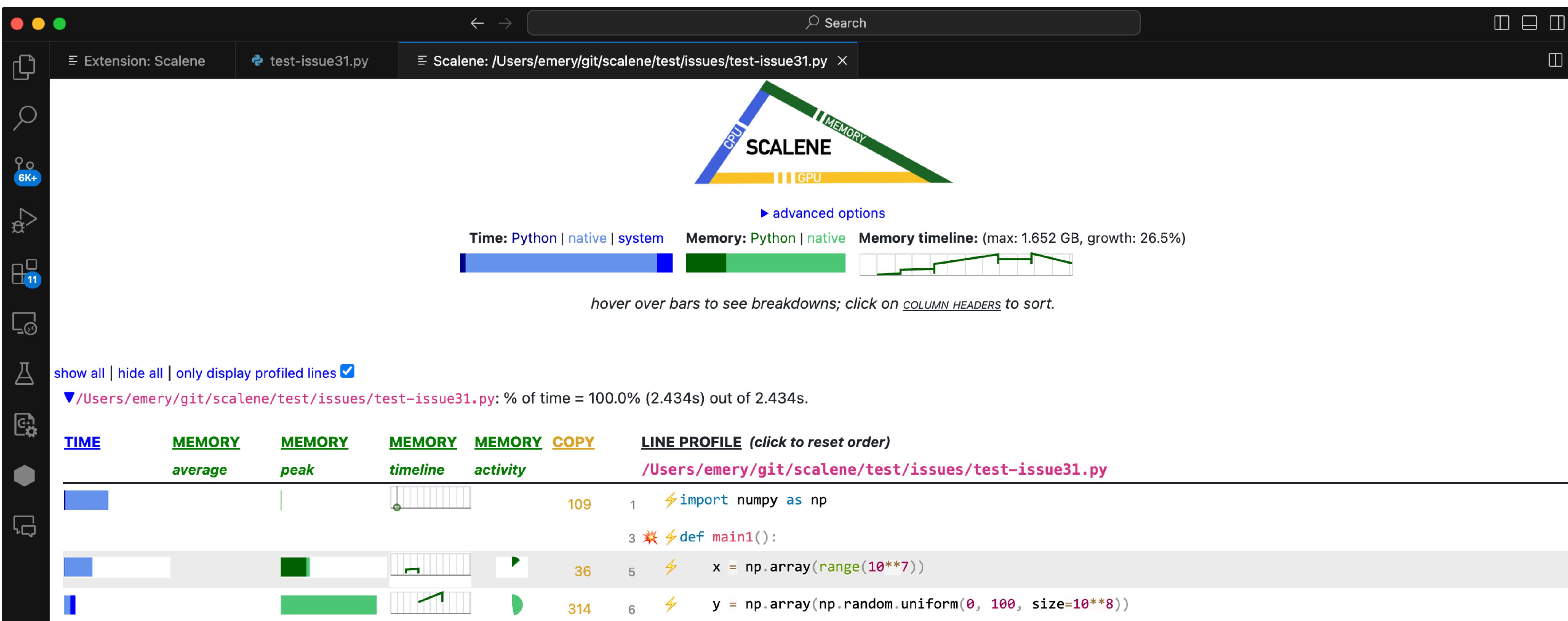


hover over bars to see breakdowns; click on COLUMN HEADERS to sort.

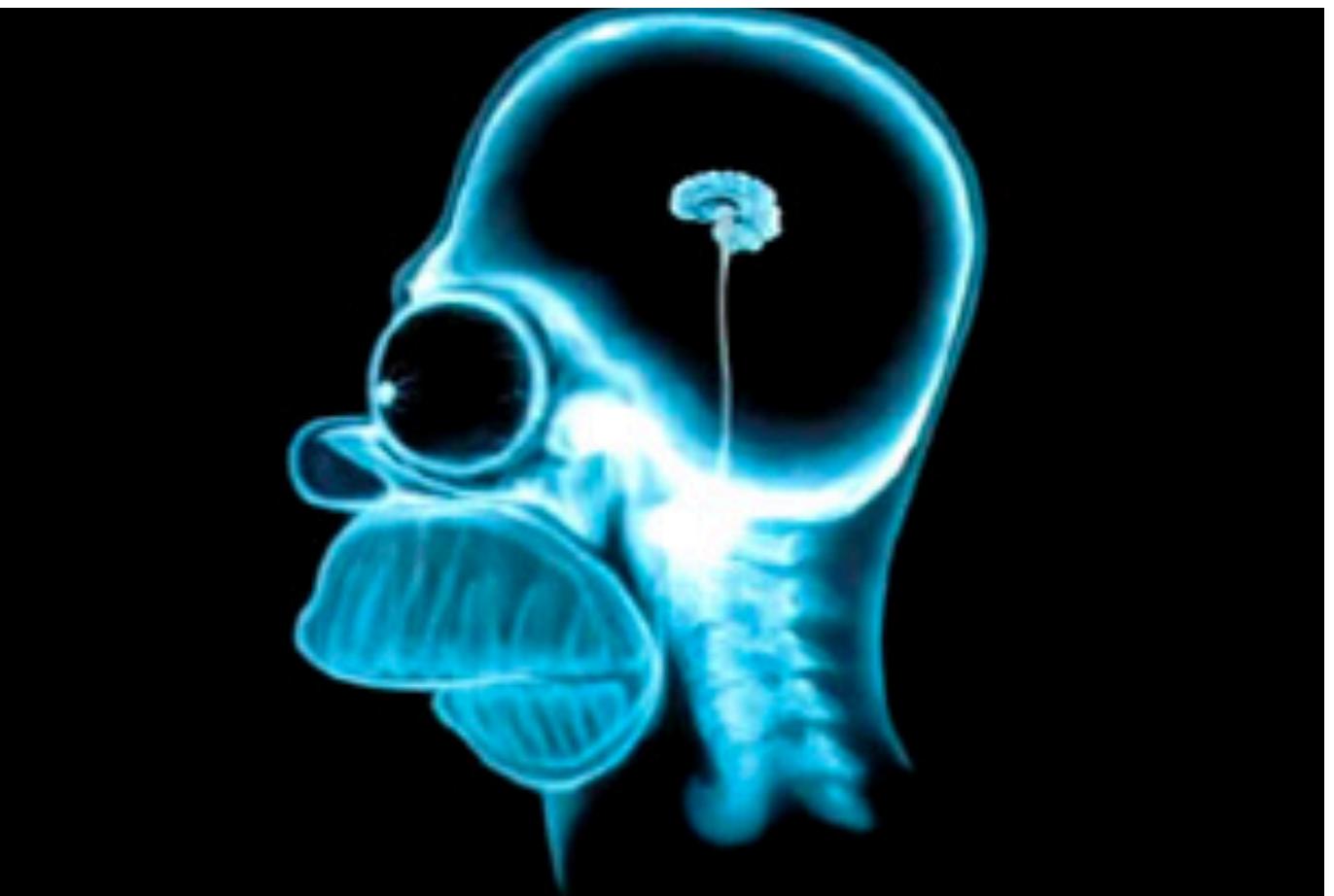
[show all](#) | [hide all](#) | [only display profiled lines](#)

▼ [test/issues/test-issue31.py](#): % of time = 100.0% (2.897s) out of 2.897s

| TIME | MEMORY average | MEMORY peak | MEMORY timeline | MEMORY activity | COPY | LINE PROFILE (click to reset order) |
|------|-------------------|----------------|--------------------|--------------------|------|---|
| 17 | 1 | 1 | 1 | 1 | 1 | import numpy as np |
| 3 | 1 | 1 | 1 | 1 | 1 | def main1(): |
| 5 | 1 | 1 | 1 | 1 | 1 | x = np.array(range(10**7)) |
| 272 | 6 | 1 | 1 | 1 | 1 | y = np.array(np.random.uniform(0, 100, size=10**8)) |
| 8 | 1 | 1 | 1 | 1 | 1 | def main2(): |
| 10 | 1 | 1 | 1 | 1 | 1 | x = np.array(range(10**7)) |



writing your code in Python



writing your code in Python

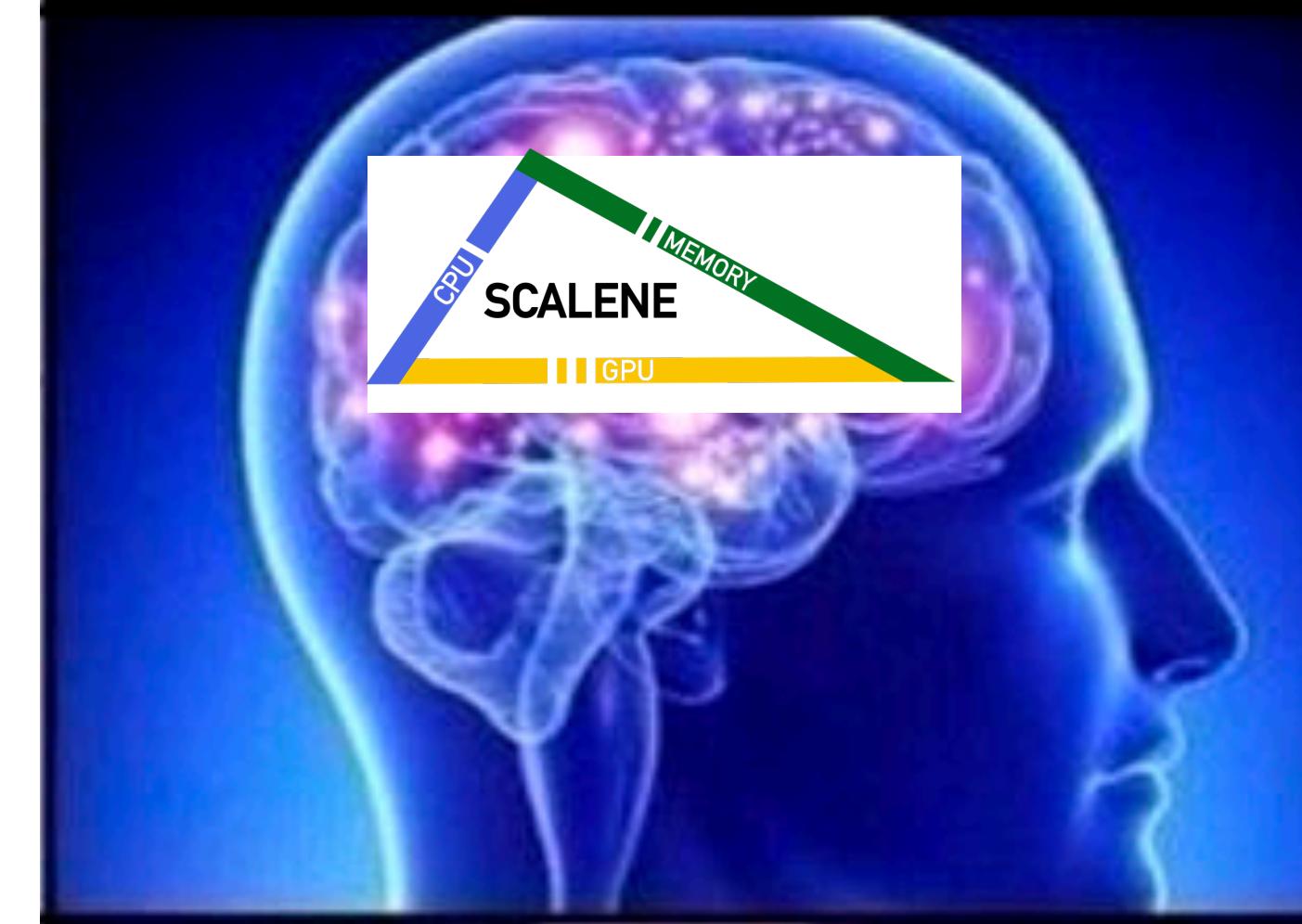
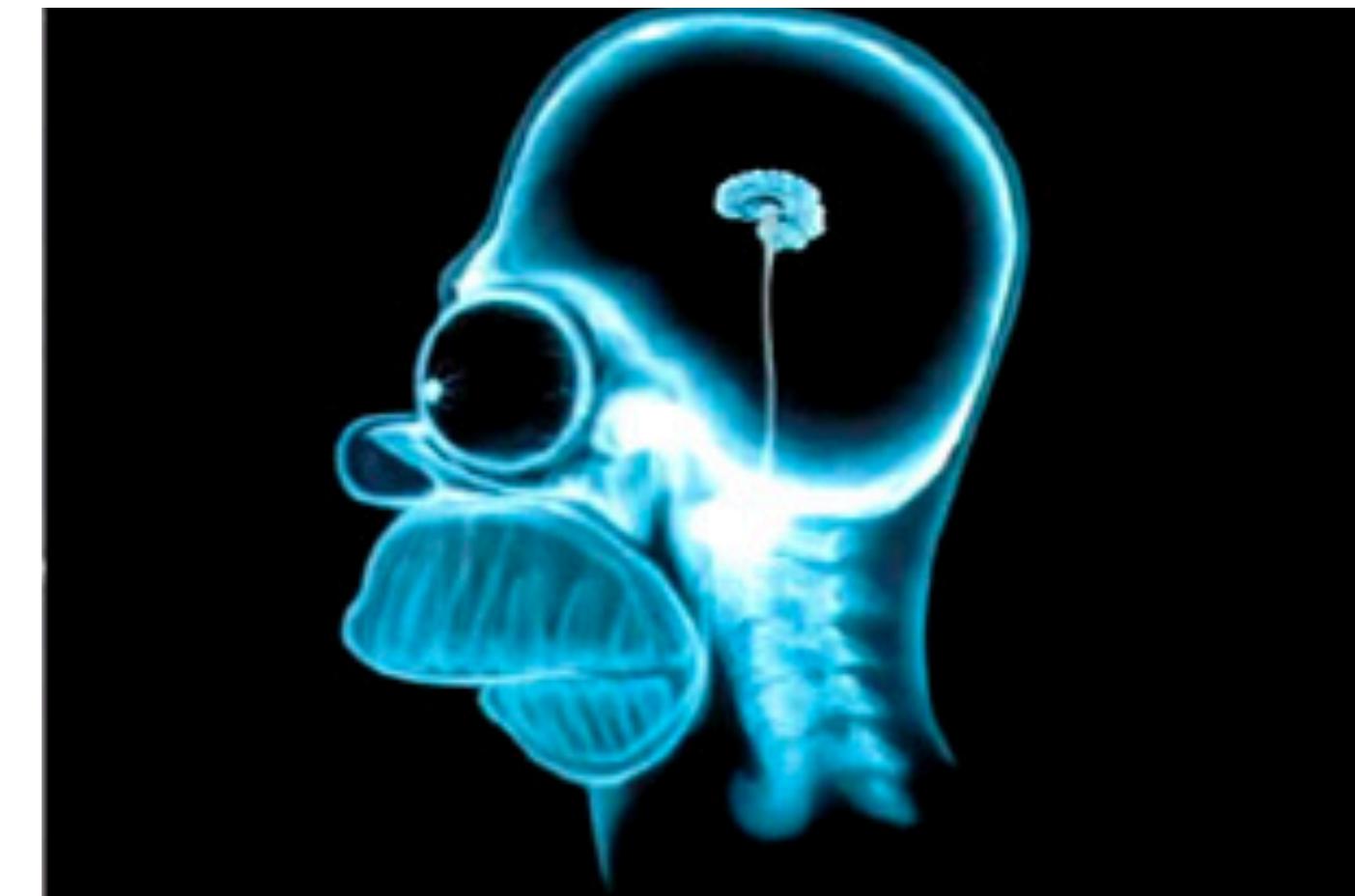
profiling your Python code with Scalene

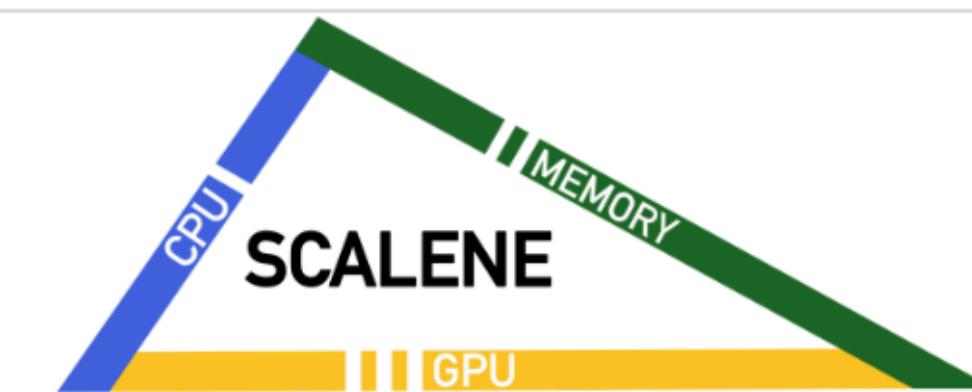


writing your code in Python

profiling your Python code with Scalene

getting Scalene to optimize your code!





Select a profile (.json)

▼ advanced options

Proposed optimizations

Enter an [OpenAI key](#) to enable: sk- [REDACTED] ✓

- Optimize runtime performance
- Optimize memory efficiency
- Include GPU optimizations

Click on an explosion (💥) to see proposed optimizations for a region of code, or on a lightning bolt (⚡) to propose optimizations for a specific line.

Click again to generate a different one.

Note that optimizations are AI-generated and may not be correct.

Time: Python | native | system

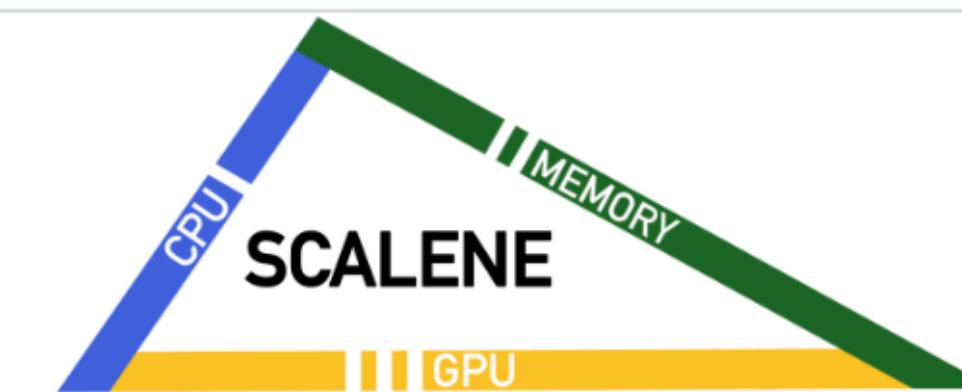
Memory: Python | native

Memory timeline: (max: 1.653 GB, growth: 21.0%)



hover over bars to see breakdowns; click on COLUMN HEADERS to sort.

[show all](#) | [hide all](#) | [only display profiled lines](#)



Select a profile (.json)

▼ advanced options

Proposed optimizations

Enter an [OpenAI key](#) to enable: sk-[REDACTED] ✓

Optimize runtime performance

Optimize memory efficiency

Include GPU optimizations

Click on an explosion (💥) to see proposed optimizations for a region of code, or on a lightning bolt (⚡) to propose optimizations for a specific line.

Click again to generate a different one.

Note that optimizations are AI-generated and may not be correct.

Time: Python | native | system

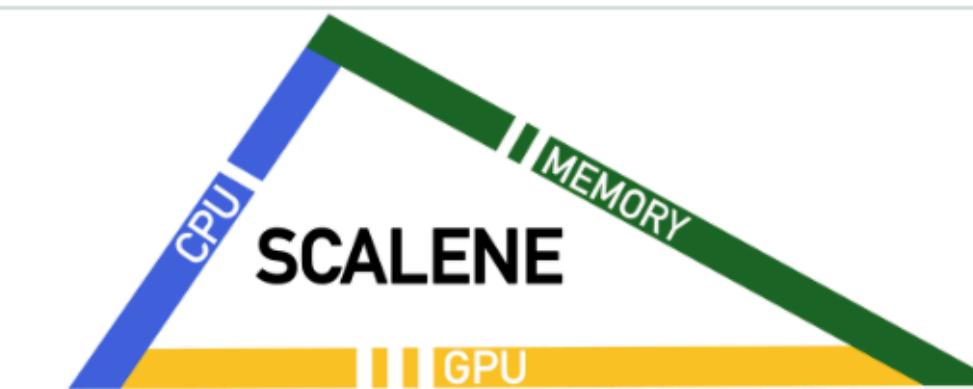
Memory: Python | native

Memory timeline: (max: 1.653 GB, growth: 21.0%)



hover over bars to see breakdowns; click on COLUMN HEADERS to sort.

[show all](#) | [hide all](#) | [only display profiled lines](#)



Select a profile (.json)

▼ advanced options

Proposed optimizations

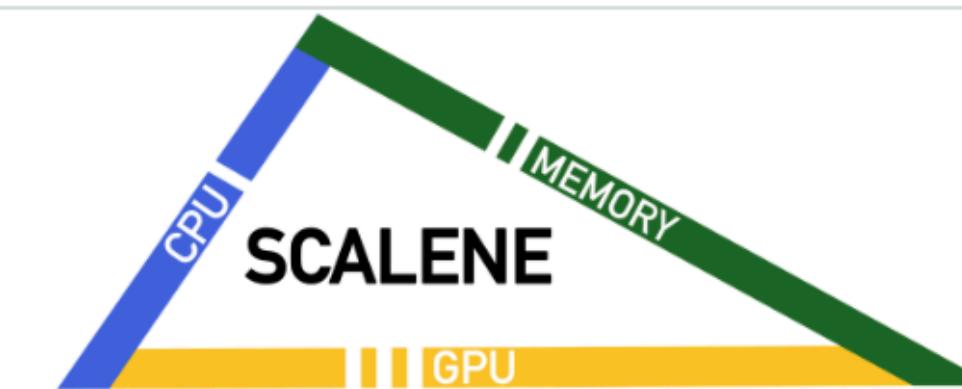
Enter an [OpenAI key](#) to enable: sk- [REDACTED] ✓

ⓘ Note

[REDACTED] needs to be connected to an [OpenAI account](#). Your account will need to have a positive balance for this to work ([check your balance](#)). If you have never purchased credits, you will need to purchase at least \$1 in credits (if your API account was created before August 13, 2023) or \$0.50 (if you have a newer API account) in order to have access to GPT-4, which [REDACTED] uses. [Get a key here](#).

Once you have an API key, set it as an environment variable called `OPENAI_API_KEY`.

```
export OPENAI_API_KEY=<your-api-key>
```



Select a profile (.json)

► advanced options

Time: Python | native | system

Memory: Python | native

Memory timeline: (max: 1.653 GB, growth: 21.0%)

hover over bars to see breakdowns; click on COLUMN HEADERS to sort.show all | hide all | only display profiled lines

▼test/issues/test-issue31.py: % of time = 100.0% (2.897s) out of 2.897s.

| TIME | MEMORY average | MEMORY peak | MEMORY timeline | MEMORY activity | COPY | LINE PROFILE (click to reset order) |
|------|-------------------|----------------|--------------------|--------------------|------|--|
| | | | | | 17 | 1 import numpy as np 2 3 def main1(): 4 5 x = np.array(range(10**7)) 6 7 y = np.array(np.random.uniform(0, 100, size=10**8)) 8 def main2(): 9 10 x = np.array(range(10**7)) |
| | | | | | 272 | |
| | | | | | | |
| | | | | | | |



Select a profile (.json)

► advanced options

Time: Python | native | system

Memory: Python | native

Memory timeline: (max: 1.653 GB, growth: 21.0%)

hover over bars to see breakdowns; click on COLUMN HEADERS to sort.show all | hide all | only display profiled lines

▼test/issues/test-issue31.py: % of time = 100.0% (2.897s) out of 2.897s.

TIMEMEMORY
averageMEMORY
peakMEMORY
timelineMEMORY COPY
activityLINE PROFILE (click to reset order)
test/issues/test-issue31.py

17

1

⚡ import numpy as np

💥⚡ def main1():

3

Proposed optimization:

Vectorize the code to reduce the number of loops and improve performance.

x = np.arange(10**7)

y = np.random.uniform(0, 100, size=10**8)



Scalene - test/issues/test-issue



Not Secure | http://plasma.umass.org/scalene-gui/



Select a profile (.json)

► advanced options

Time: Python | native | system

Memory: Python | native

Memory timeline: (max: 1.653 GB, growth: 21.0%)



hover over bars to see breakdowns; click on COLUMN HEADERS to sort.

show all | hide all | only display profiled lines

▼test/issues/test-issue31.py: % of time = 100.0% (2.897s) out of 2.897s.

TIMEMEMORYMEMORYMEMORYMEMORY COPYLINE PROFILE (click to reset order)

test/issues/test-issue31.py



5 ⚡ x = np.array(range(10**7))

272 ⚡ y = np.array(np.random.uniform(0, 100, size=10**8))



15 ⚡ x = np.arange(10**7)

16 ⚡ y = np.random.uniform(0, 100, size=10**8)



Scalene - test/issues/test-issue



Not Secure | http://plasma.umass.org/scalene-gui/



Select a profile (.json)

► advanced options

Time: Python | native | system

Memory: Python | native

Memory timeline: (max: 1.653 GB, growth: 21.0%)

hover over bars to see breakdowns; click on COLUMN HEADERS to sort.show all | hide all | only display profiled lines

▼test/issues/test-issue31.py: % of time = 100.0% (2.897s) out of 2.897s.

TIMEMEMORYMEMORYMEMORYMEMORY COPYLINE PROFILE (click to reset order)

test/issues/test-issue31.py



▶ 5 ⚡ x = np.array(range(10**7))

◀ 272 ⚡ y = np.array(np.random.uniform(0, 100, size=10**8))



▶ 15 ⚡ x = np.arange(10**7)

▶ 16 ⚡ y = np.random.uniform(0, 100, size=10**8)

90x speedup

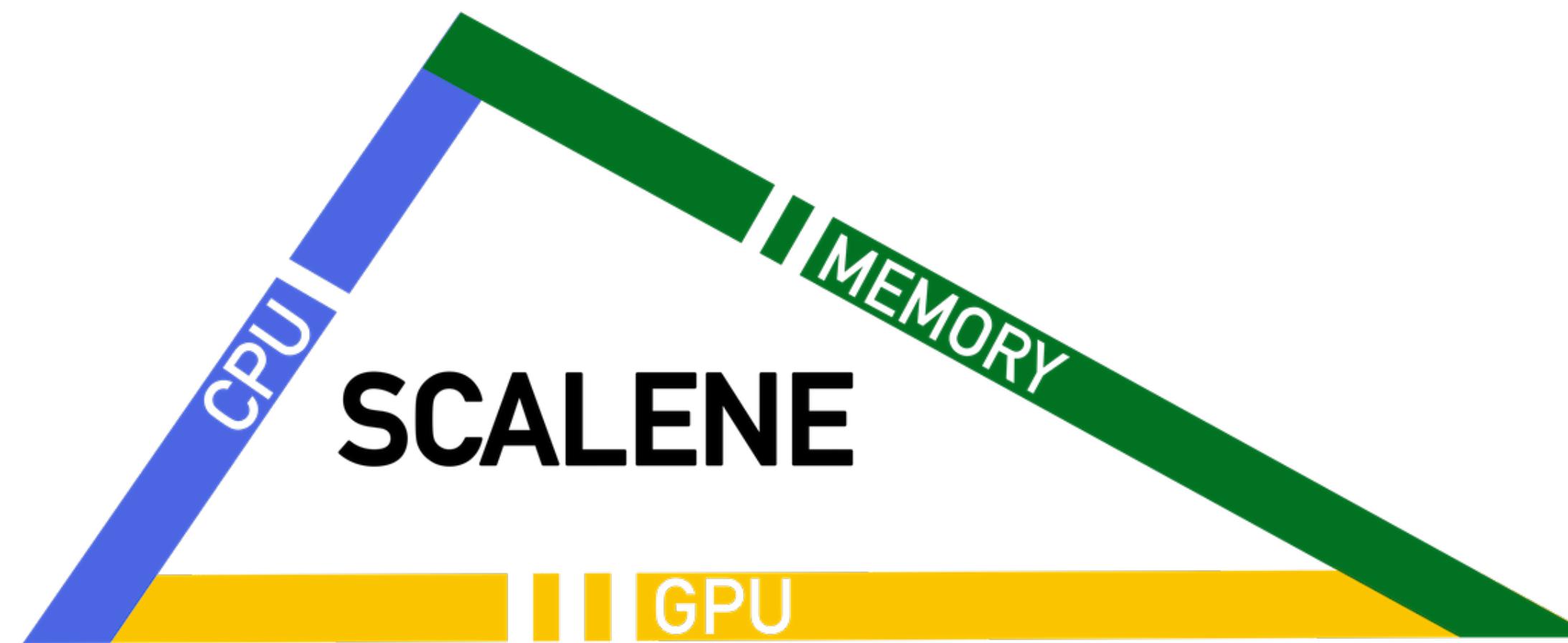
#58 (comment) presents the following code:

```
for i in range(n_features):
    for n in range(n_samples):
        subgrad[i] += (-y[n] * X[n][i]) if y[n] * (np.dot(X[n], w) + b) < 1 else 0
    subgrad[i] += self.lambda1 * (-1 if w[i] < 0 else 1) + 2 * self.lambda2 * w[i]
```

Scalene proposes the following optimization:

```
# Vectorized operations to replace for loops
subgrad[:-1] = np.sum(-y[:, None] * X * (y * (X.dot(w) + b) < 1)[:, None], axis=0)
subgrad[:-1] += self.lambda1 * np.sign(w) + 2 * self.lambda2 * w
subgrad[-1] = np.sum(-y * (y * (X.dot(w) + b) < 1))
```

Scalene's proposed optimization accelerates the original code by at least 90x (89 seconds to 1 second, when running 500 iterations), and takes full advantage of multiple cores.



AI POWERED OPTIMIZATIONS!

CPU

PYTHON

NATIVE

SYS%

MEMORY

PYTHON

NATIVE

AVERAGE &
PEAK

MEMORY
USAGE

OVER TIME,
% OF MEM
ALLOCATED

COPY
VOLUME

OVER TIME,
(MB/s)

GPU

UTIL %,

PEAK
MEMORY



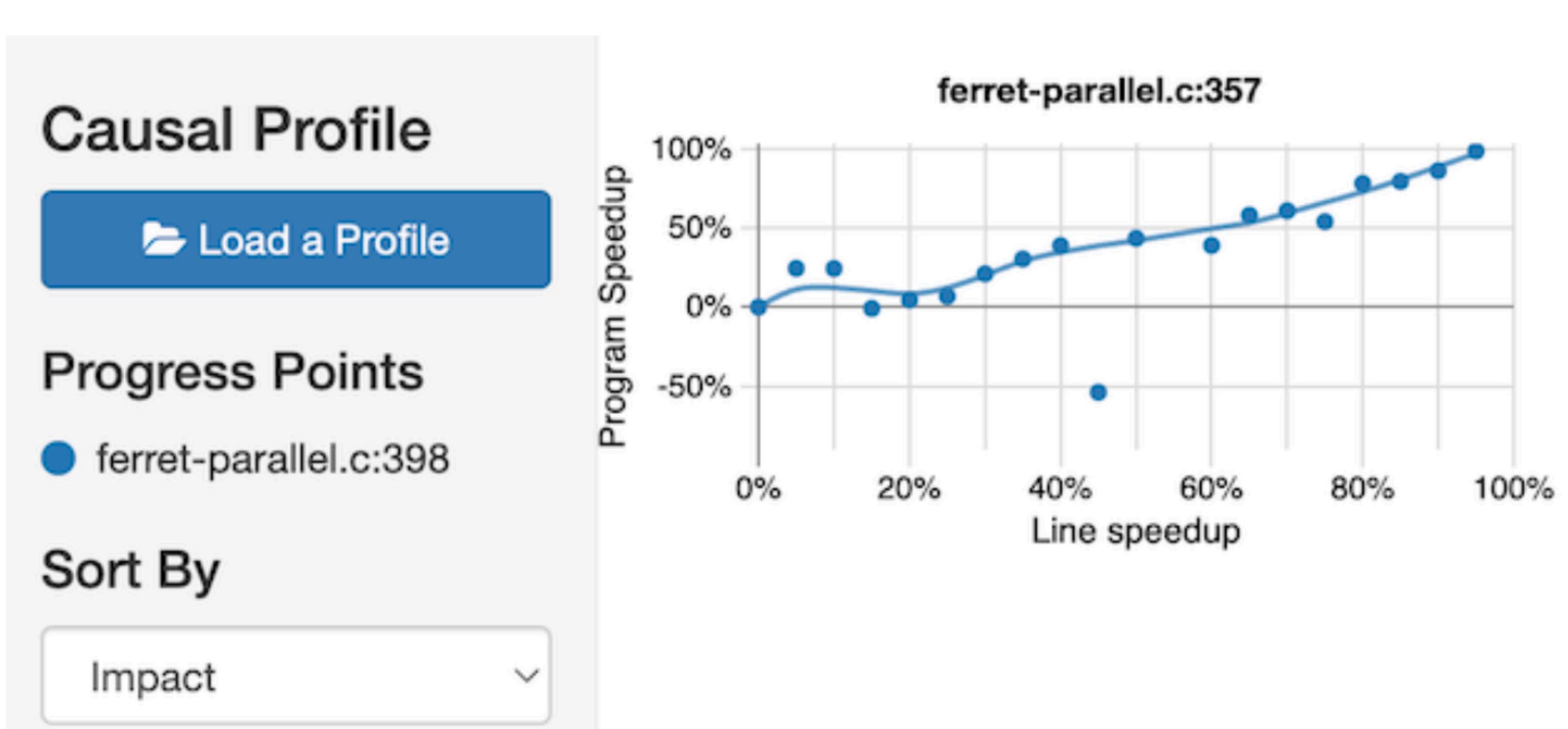
Coz: Finding Code that Counts with Causal Profiling [🔗](#)

by [Charlie Curtsinger](#) and [Emery Berger](#)

 downloads 444k

Coz is a profiler for native code (C/C++/Rust) that unlocks optimization opportunities missed by traditional profilers. Coz employs a novel technique called *causal profiling* that measures optimization potential. It predicts what the impact of optimizing code will have on overall throughput or latency.

Profiles generated by Coz show the "bang for buck" of optimizing a line of code in an application. In the below profile, almost every effort to optimize the performance of this line of code directly leads to an increase in overall performance, making it an excellent candidate for optimization efforts.



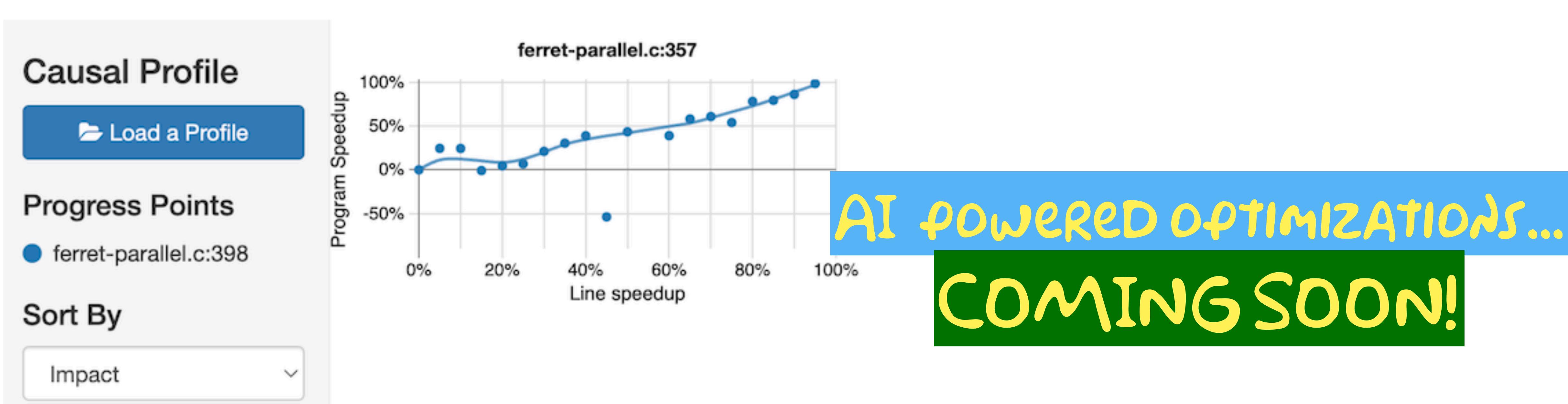
Coz: Finding Code that Counts with Causal Profiling [🔗](#)

by [Charlie Curtsinger](#) and [Emery Berger](#)

 downloads 444k

Coz is a profiler for native code (C/C++/Rust) that unlocks optimization opportunities missed by traditional profilers. Coz employs a novel technique called *causal profiling* that measures optimization potential. It predicts what the impact of optimizing code will have on overall throughput or latency.

Profiles generated by Coz show the "bang for buck" of optimizing a line of code in an application. In the below profile, almost every effort to optimize the performance of this line of code directly leads to an increase in overall performance, making it an excellent candidate for optimization efforts.





evolve

A cartoon illustration of an ostrich with its head buried in the sand. The ostrich has a large, fluffy brown and tan striped crest, a small white patch around its eye, and a long, thin beak. It is standing on a sandy ground with some sparse grass. The word "evolve" is overlaid in blue text.

evolve

profiler that suggests optimizations

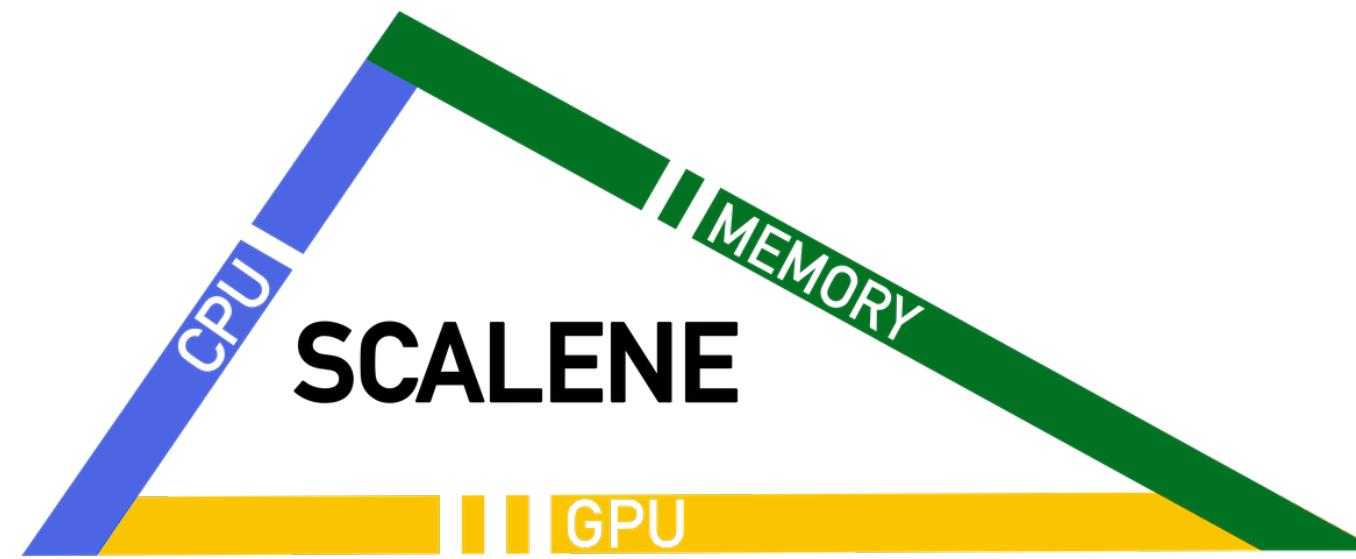
A cartoon illustration of an ostrich with its head buried in the sand. The ostrich has a large, fluffy brown and grey feathered head and neck, with a long, thin beak pointing upwards. Its body is dark brown with some light spots, and it has small legs and feet. The background is a simple, light-colored sandy ground.

evolve

profiler that suggests optimizations
"closes loop" between tool and human's goal



profiler that suggests optimizations
"closes loop" between tool and human's goal

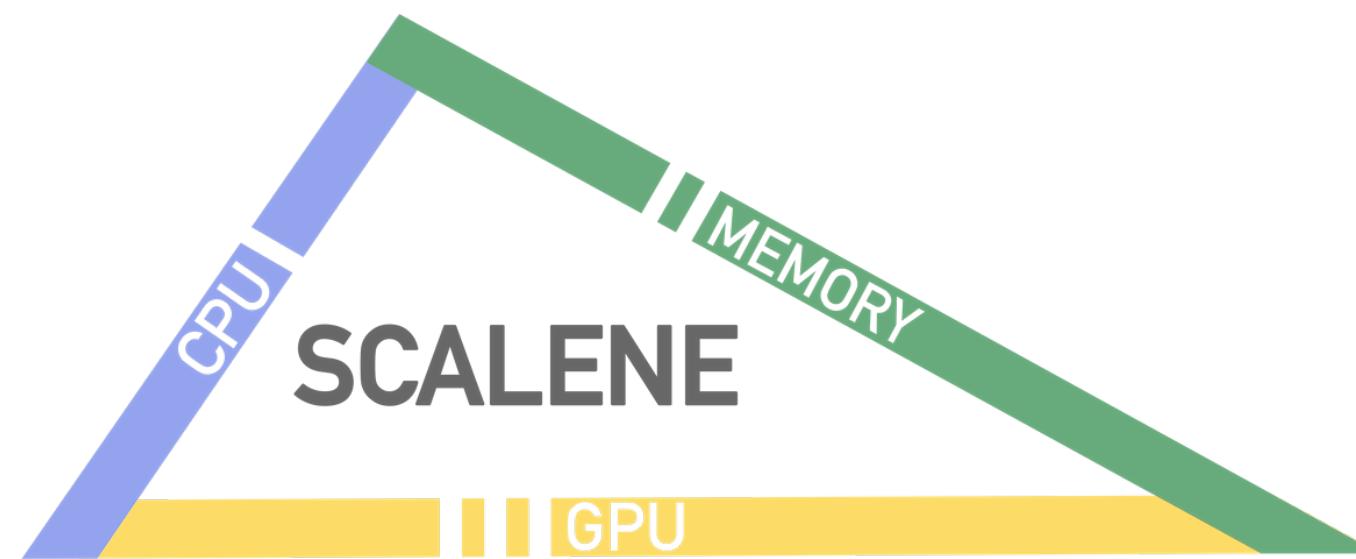


here's where your code
could be more efficient
maybe, you figure it out

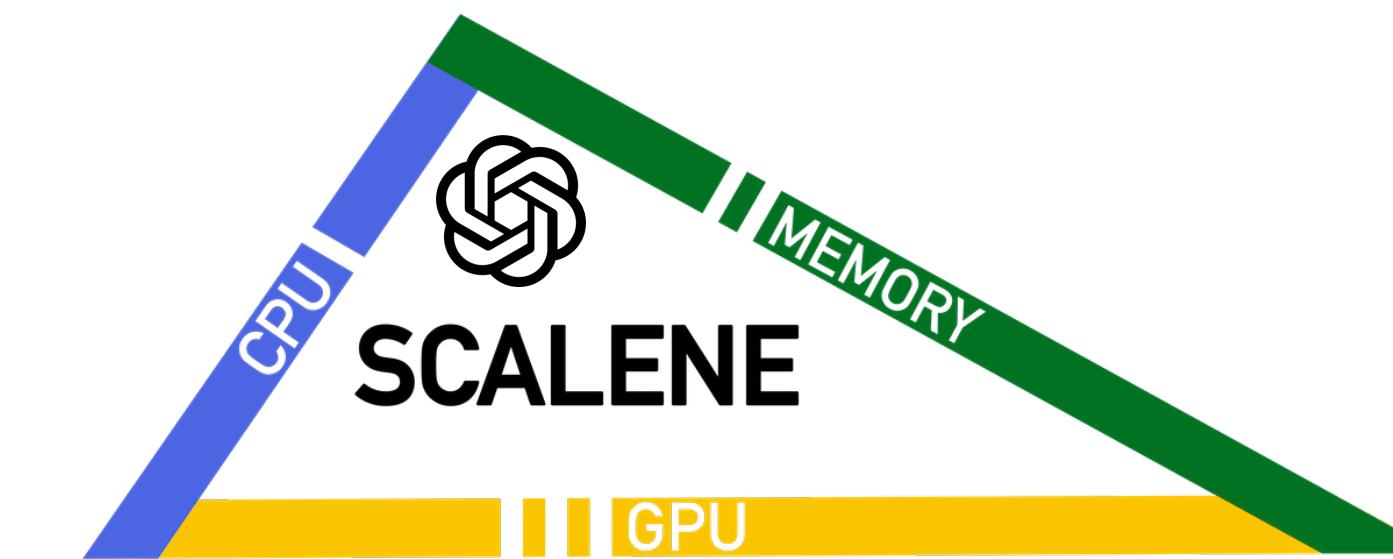




profiler that suggests optimizations
"closes loop" between tool and human's goal



here's where your code
could be more efficient
maybe, you figure it out



I figured out where your
code could be more efficient
so you go get some beer



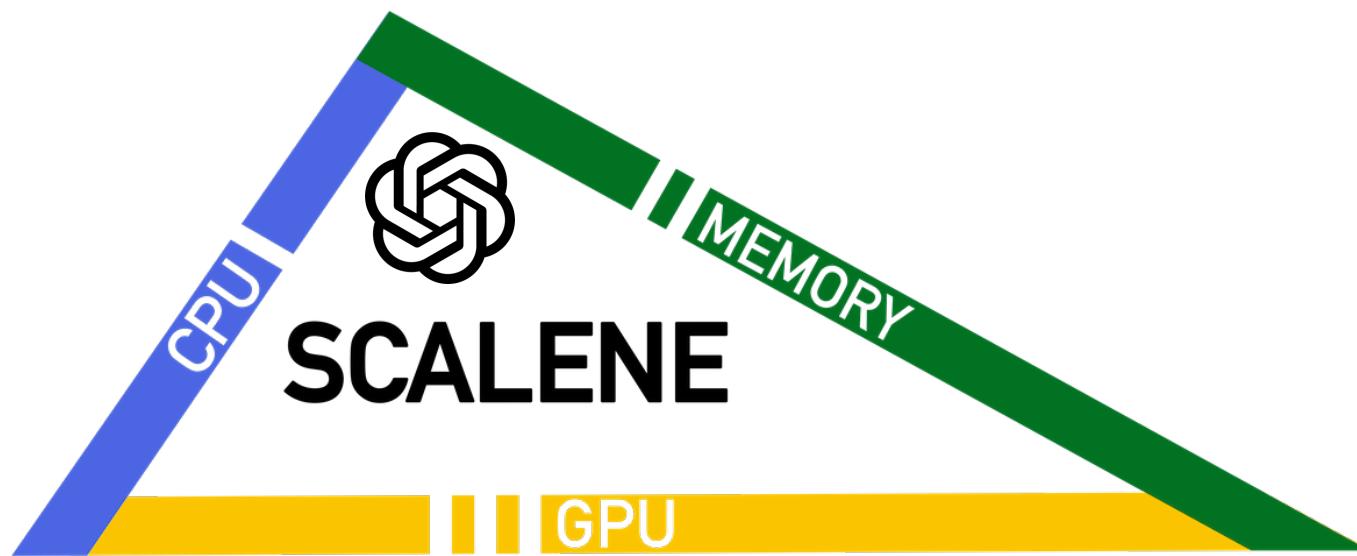


exploit
niche



exploit
niche

profiler knows where code is inefficient



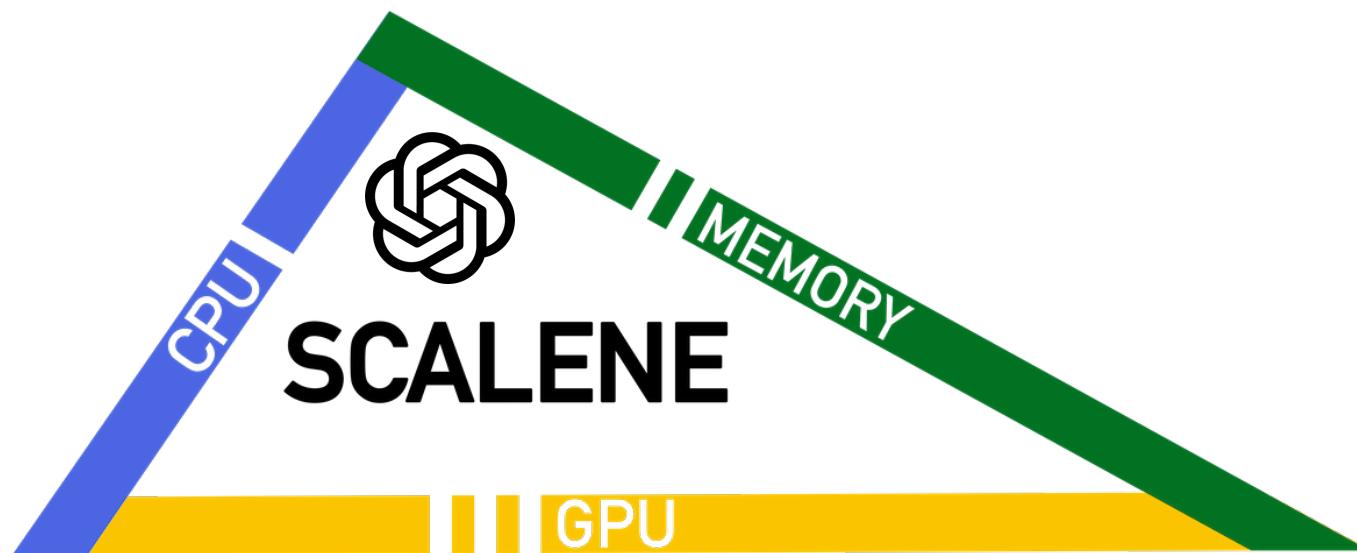
I figured out where your
code could be more efficient
so you go get some beer





exploit
niche

profiler knows where code is inefficient
profiler knows why code is inefficient



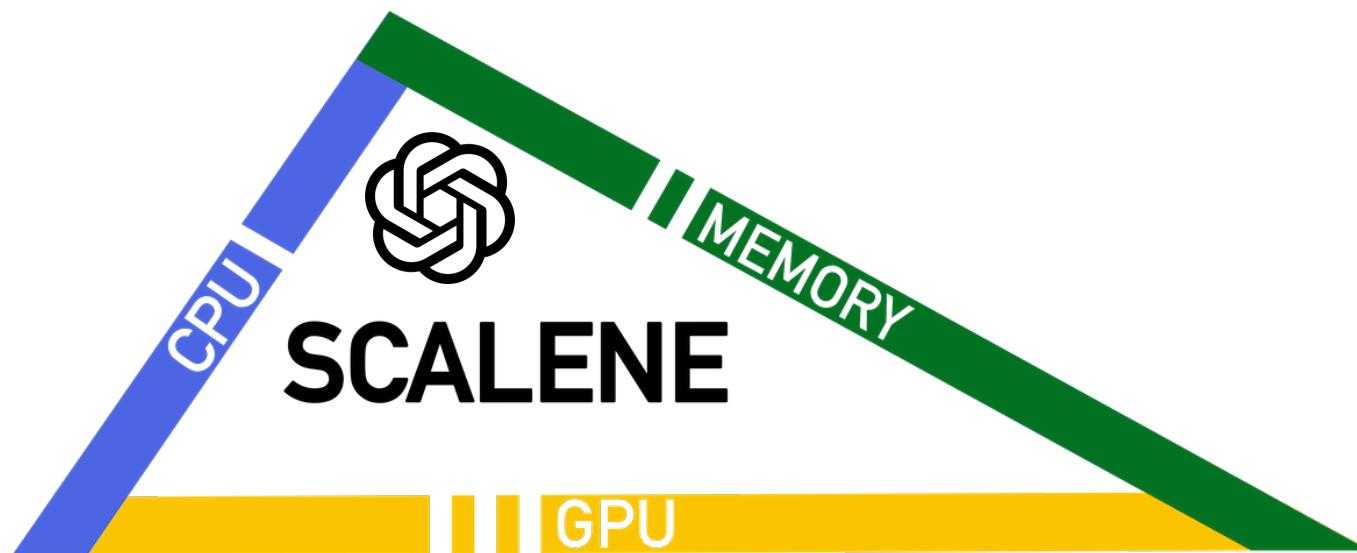
I figured out where your
code could be more efficient
so you go get some beer





exploit
niche

profiler knows where code is inefficient
profiler knows why code is inefficient
lots of time in interpreter



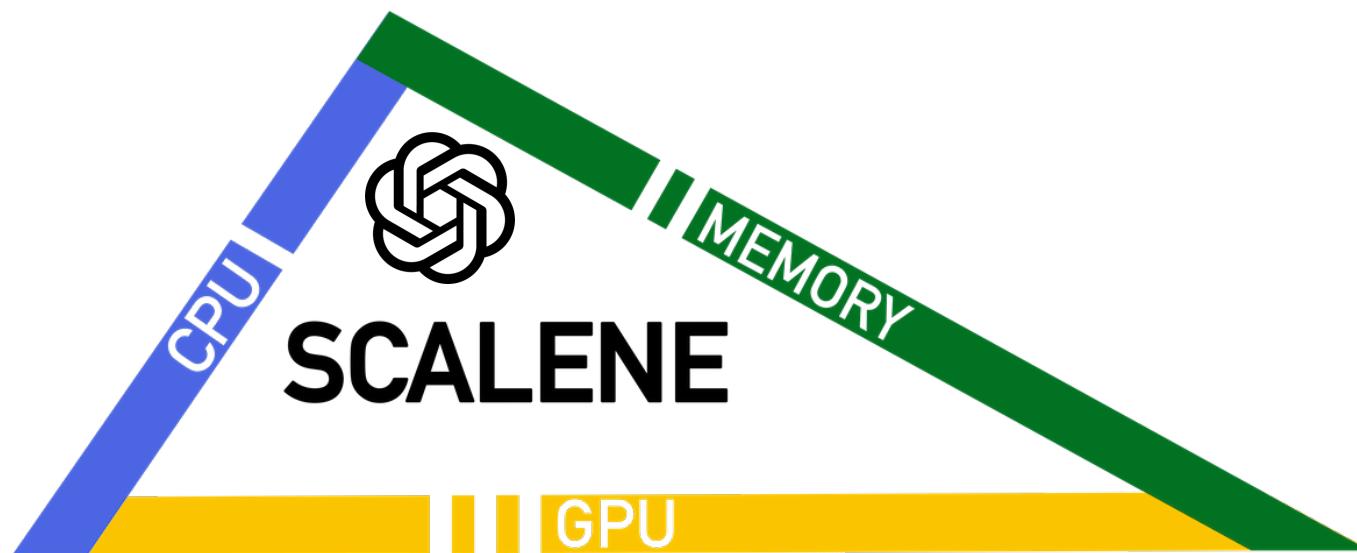
I figured out where your
code could be more efficient
so you go get some beer





exploit
niche

profiler knows where code is inefficient
profiler knows why code is inefficient
lots of time in interpreter ⇒ use native libraries



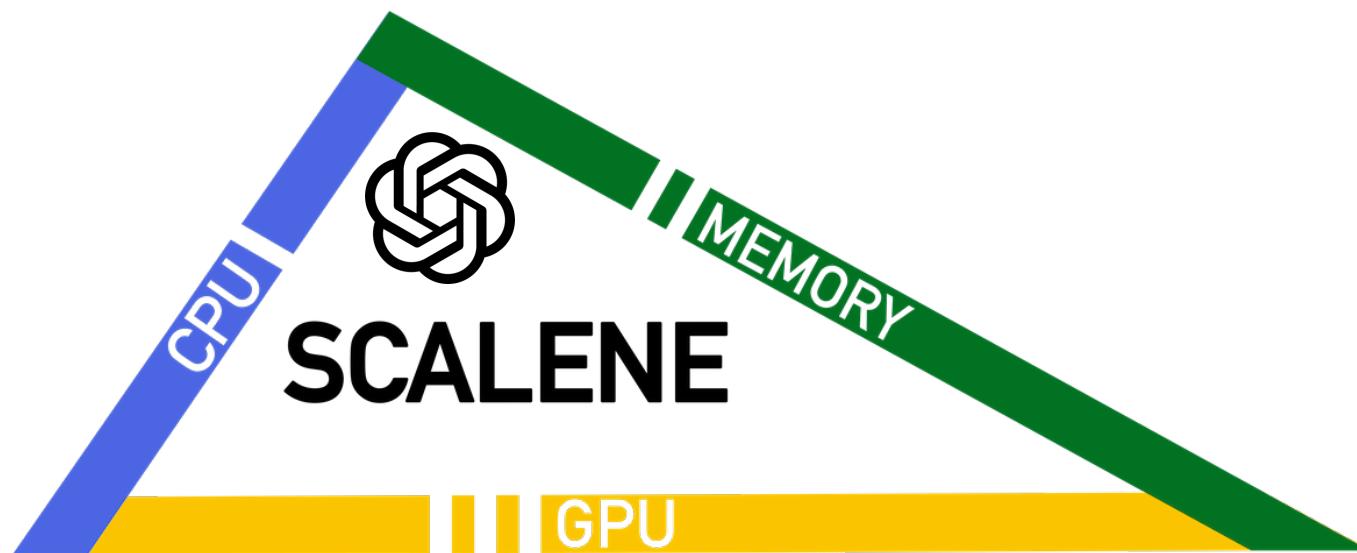
I figured out where your
code could be more efficient
so you go get some beer





exploit
niche

profiler knows where code is inefficient
profiler knows why code is inefficient
lots of time in interpreter ⇒ use native libraries
low core utilization



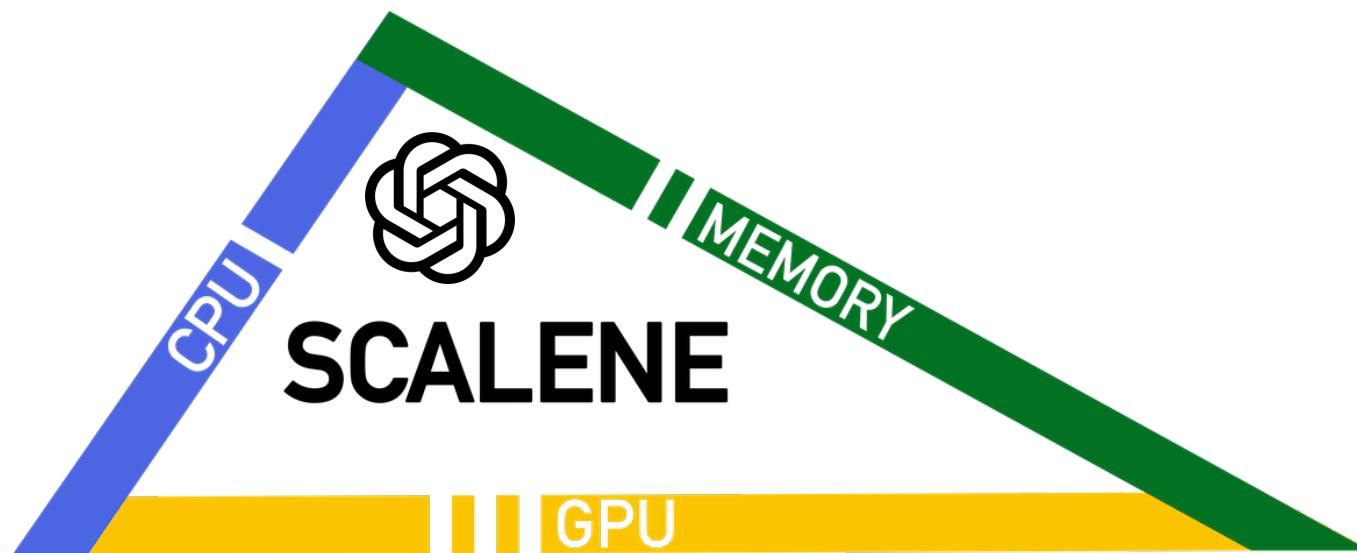
I figured out where your
code could be more efficient
so you go get some beer





exploit
niche

profiler knows where code is inefficient
profiler knows why code is inefficient
lots of time in interpreter ⇒ use native libraries
low core utilization ⇒ use vectorization / MT / MP

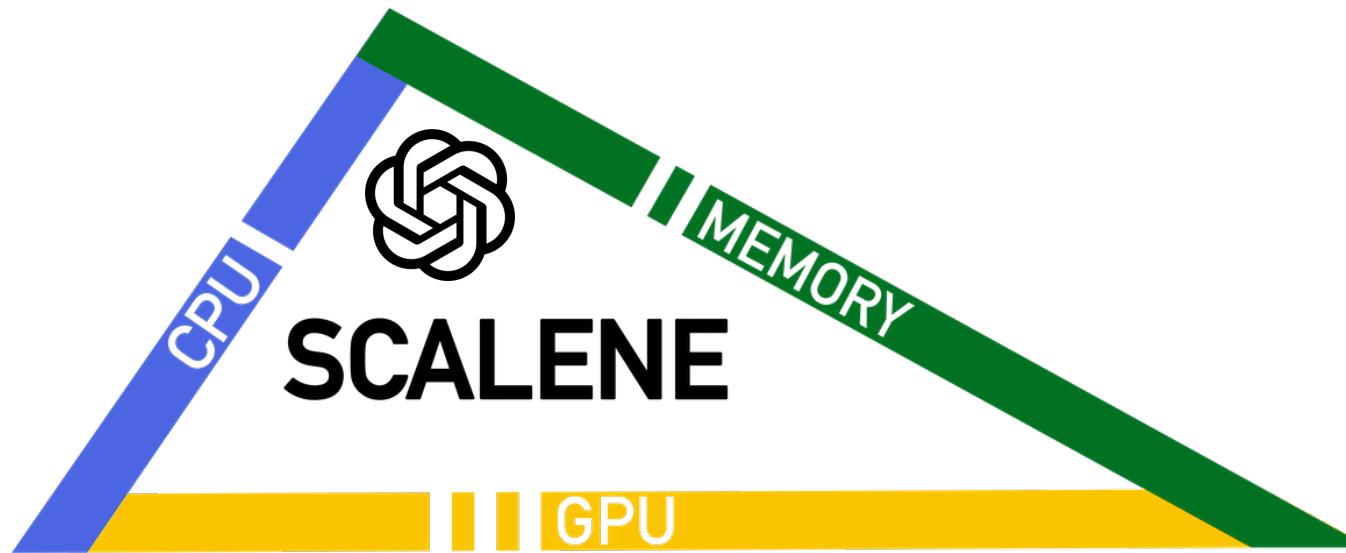


I figured out where your
code could be more efficient
so you go get some beer





exploit
niche



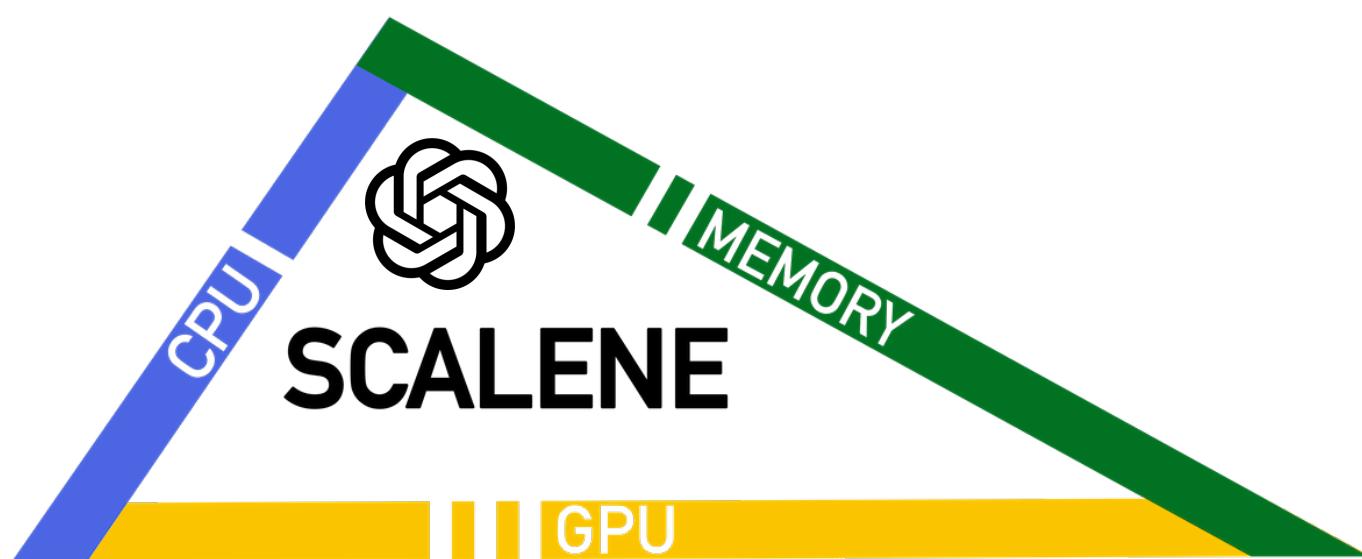
I figured out where your
code could be more efficient
so you go get some beer



profiler knows where code is inefficient
profiler knows why code is inefficient
lots of time in interpreter ⇒ use native libraries
low core utilization ⇒ use vectorization / MT / MP
no usage of GPU



exploit
niche



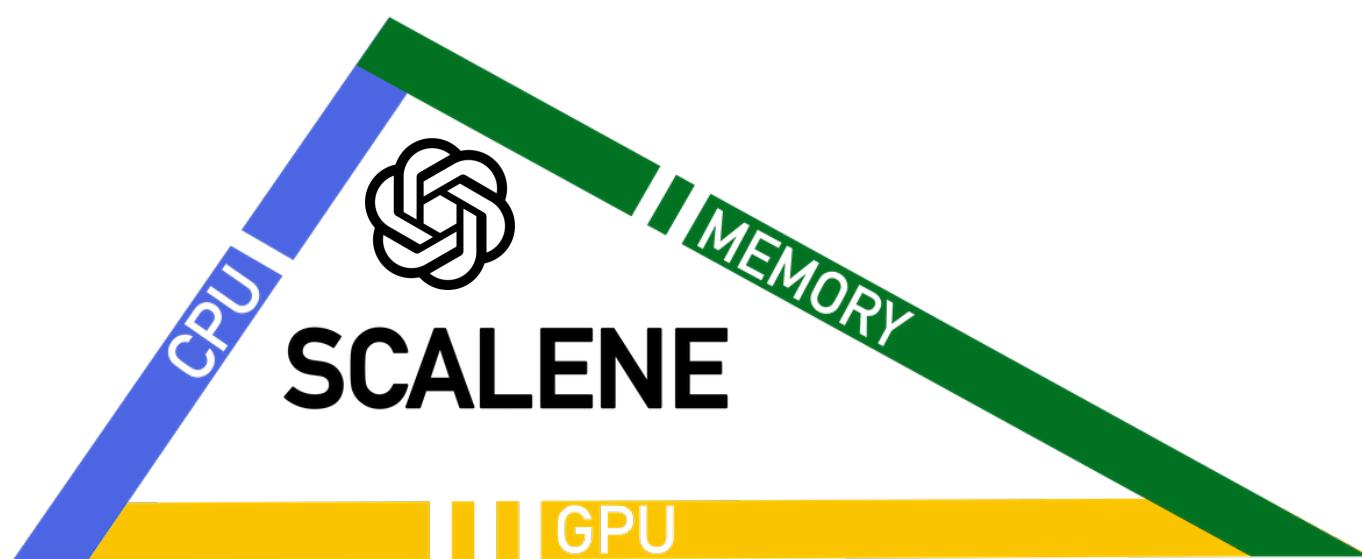
I figured out where your
code could be more efficient
so you go get some beer



profiler knows where code is inefficient
profiler knows why code is inefficient
lots of time in interpreter ⇒ use native libraries
low core utilization ⇒ use vectorization / MT / MP
no usage of GPU ⇒ use GPU-optimized libraries



exploit
niche



I figured out where your
code could be more efficient
so you go get some beer



profiler knows where code is inefficient
profiler knows why code is inefficient
lots of time in interpreter ⇒ use native libraries
low core utilization ⇒ use vectorization / MT / MP
no usage of GPU ⇒ use GPU-optimized libraries
profiler has original code

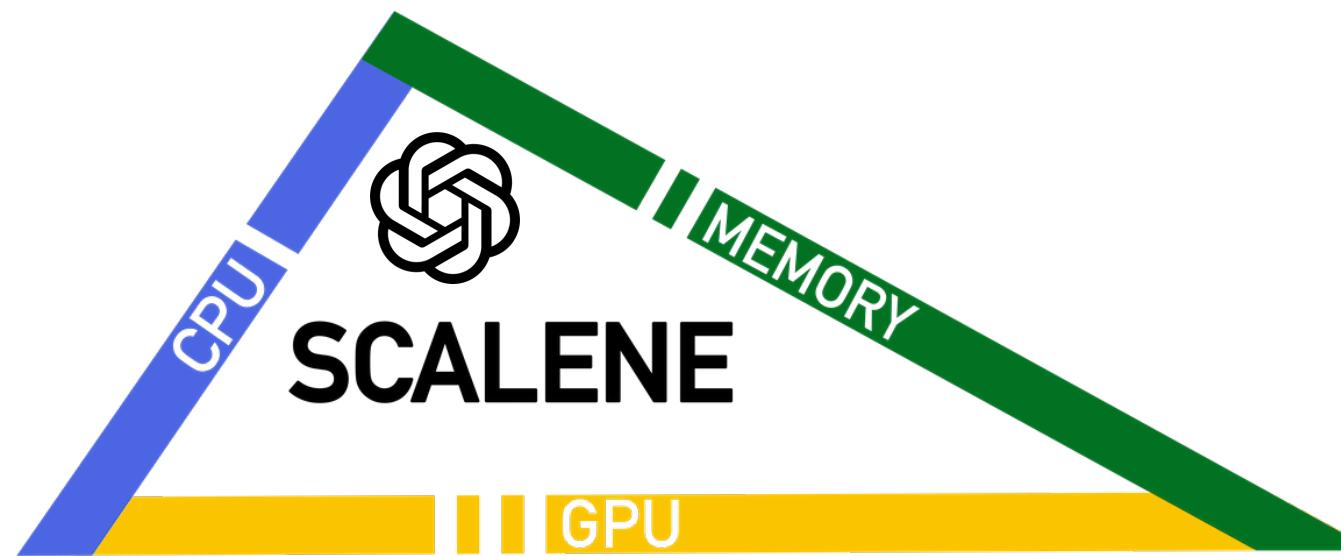


**ensure
fitness**



ensure
fitness

use original code as oracle



I figured out where your
code could be more efficient
so you go get some beer

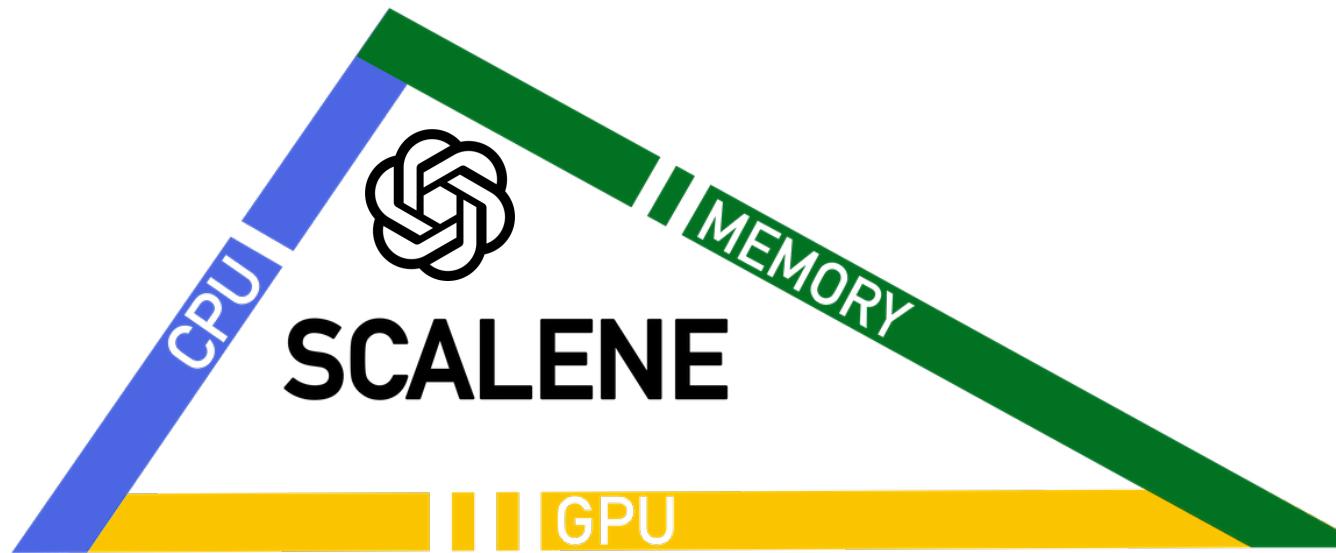




ensure
fitness

use original code as oracle

correctness: compare to original via testing



I figured out where your
code could be more efficient
so you go get some beer

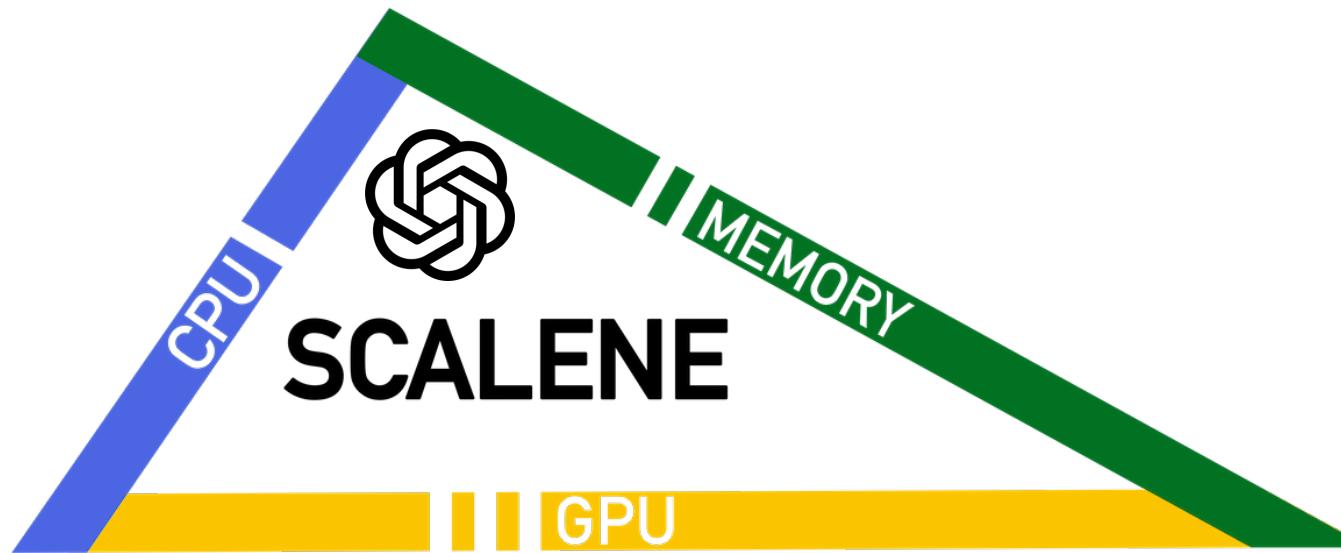




ensure
fitness

use original code as oracle

correctness: compare to original via testing



I figured out where your
code could be more efficient
so you go get some beer



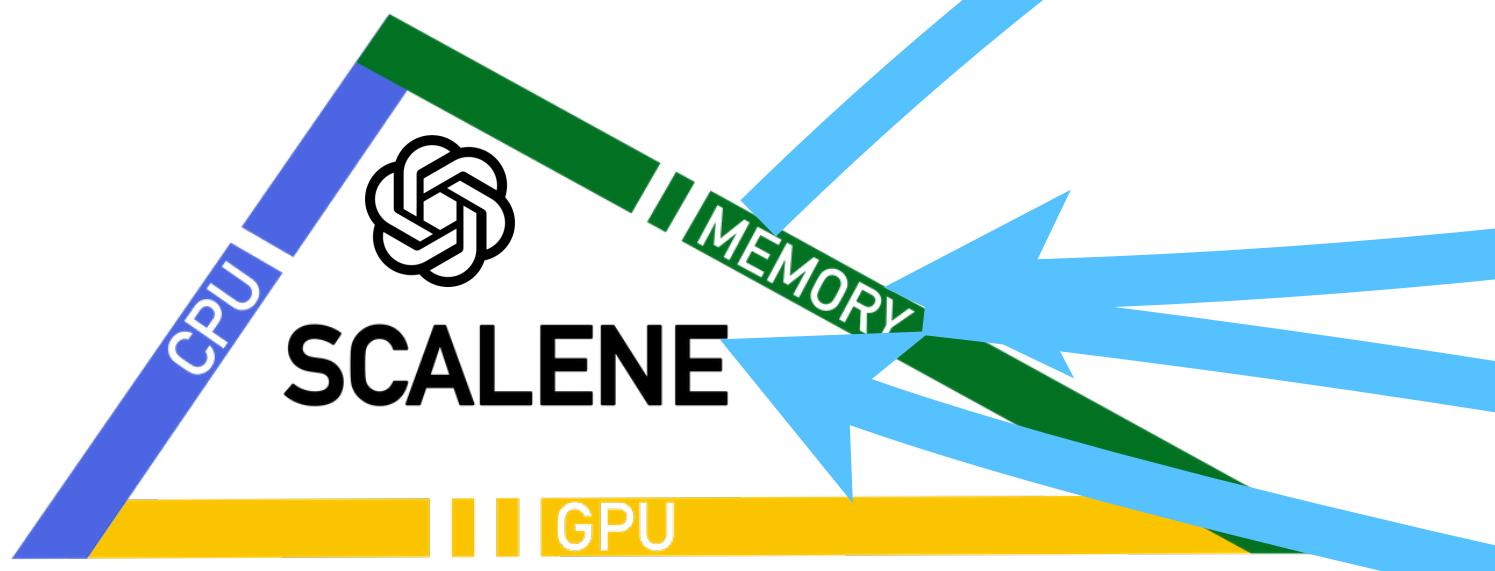
perf: compare to original via execution



ensure
fitness

use original code as oracle

correctness: compare to original via testing



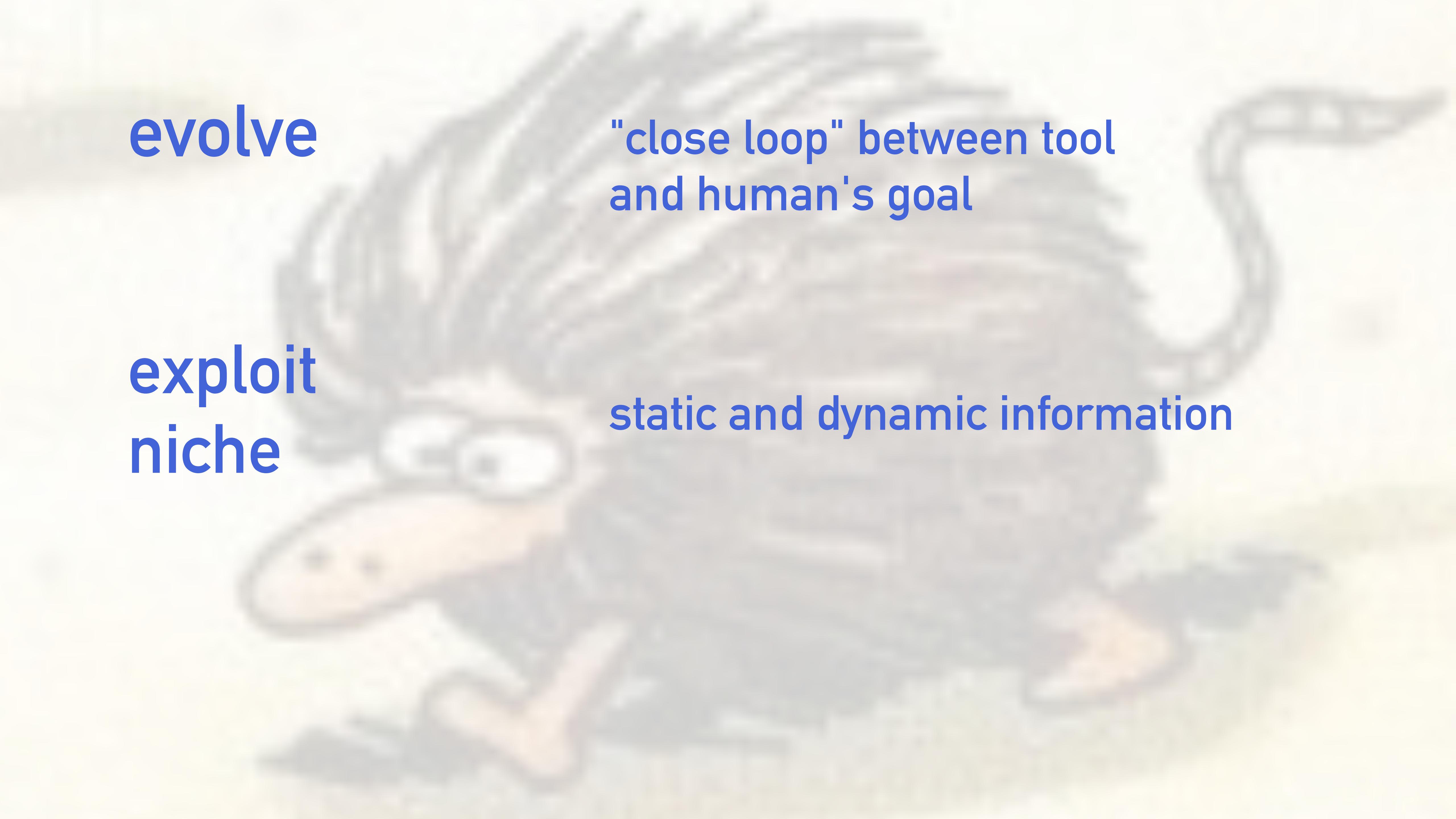
I figured out where your
code could be more efficient
so you go get some beer



perf: compare to original via execution

evolve

"close loop" between tool
and human's goal



evolve

"close loop" between tool
and human's goal

exploit
niche

static and dynamic information

evolve

"close loop" between tool
and human's goal

exploit
niche

static and dynamic information

ensure
fitness

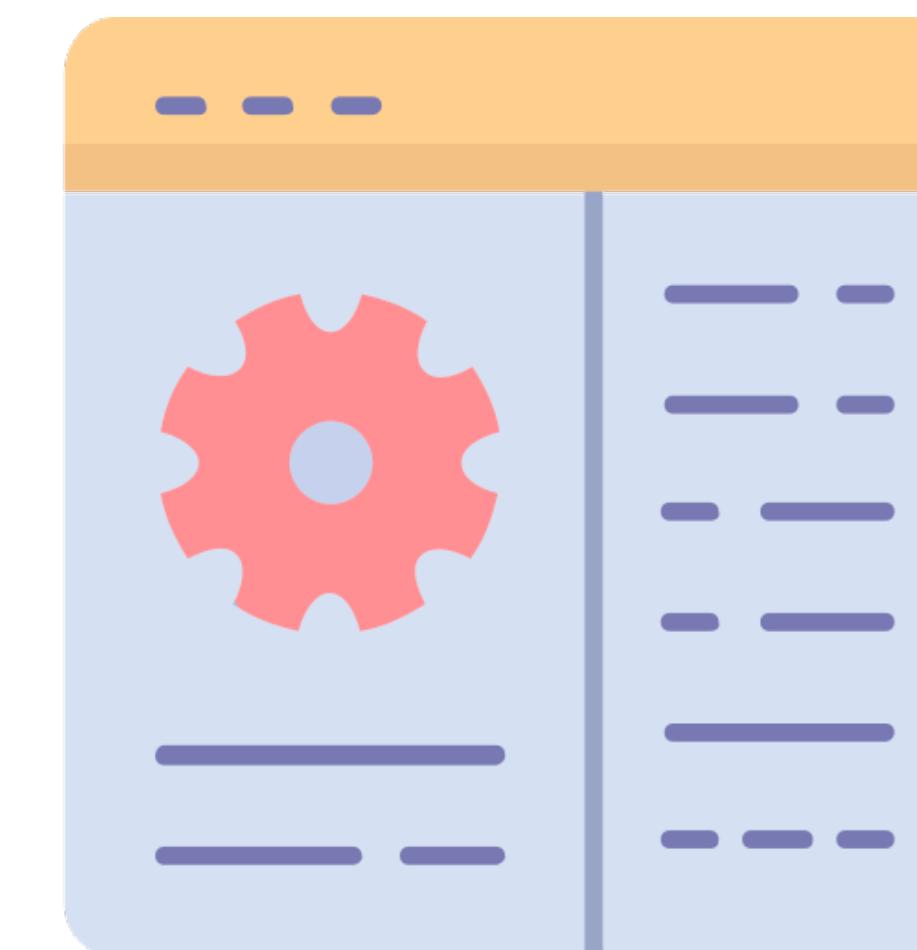
use to verify success / retry
/ refine prompts



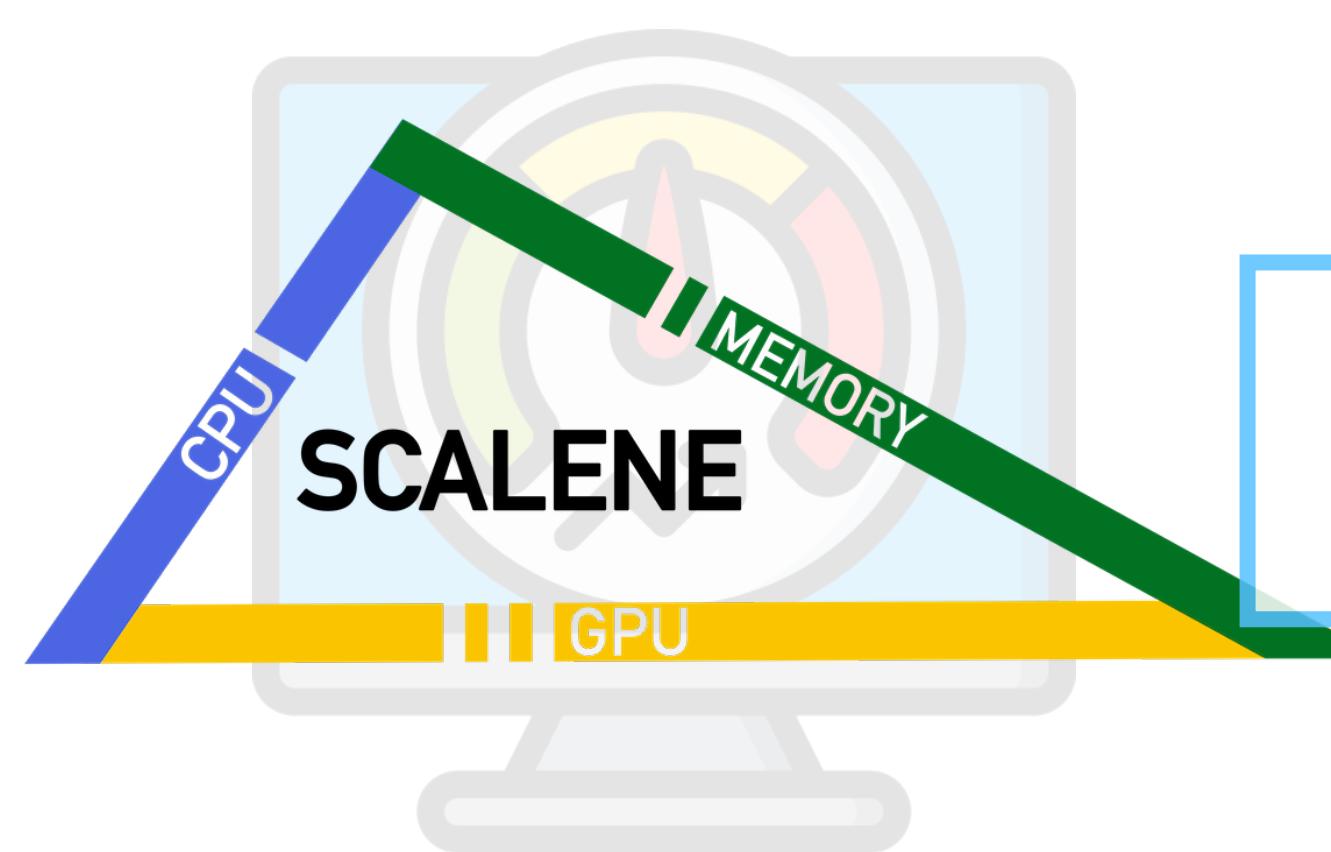
profiler



debugger



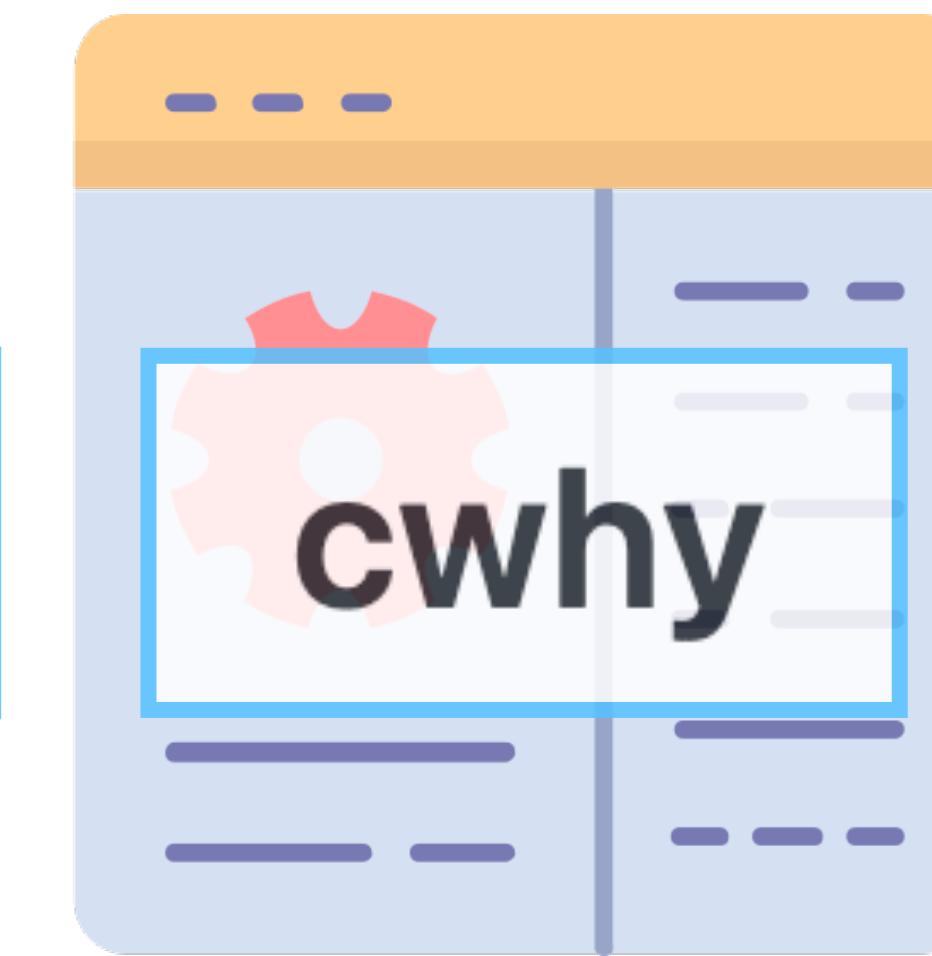
compiler



profiler



debugger



compiler



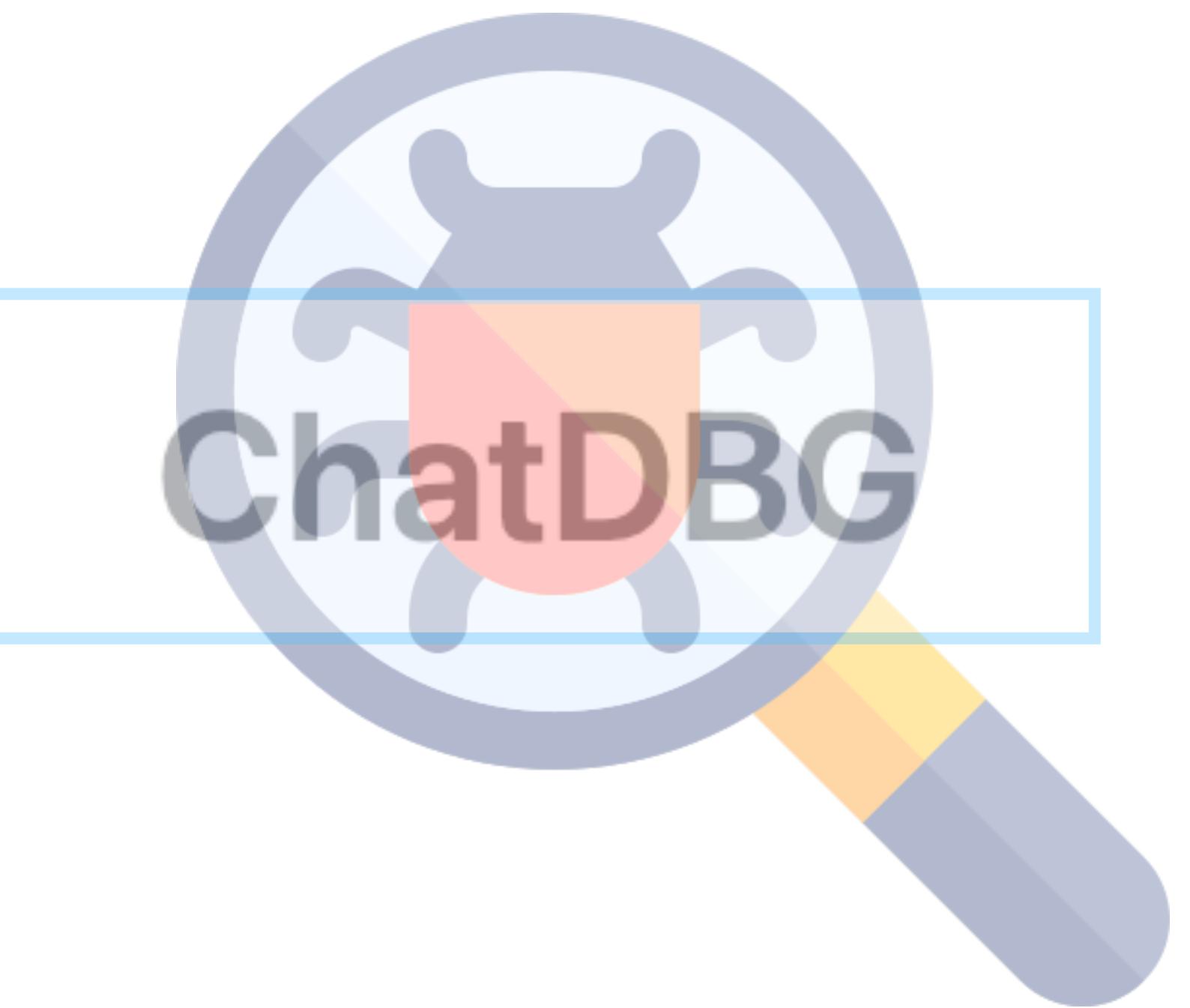
ChatDBG is an experimental debugger for C/C++/Python/Rust code that integrates large language models into a standard debugger (`pdb` , `lldb` , and `gdb`) to help debug your code. With ChatDBG, you can ask your debugger `why` your program failed, and it will provide a diagnosis and suggest a fix.

As far as we are aware, ChatDBG is the *first* debugger to automatically perform root cause analysis and to provide suggested fixes.

[pypi](#) v0.2.0 [downloads](#) 15k [downloads/month](#) 2k [python](#) 3.8 | 3.9 | 3.10 | 3.11



```
$ gdb ./a.out
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1
[...]
Reading symbols from ./a.out...
(gdb-ChatDBG) run
```



```
$ gdb ./a.out
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1
[...]
Reading symbols from ./a.out...
(gdb-ChatDBG) run
```

```
Starting program: /media/psf/Home/git/
ChatDBG/a.out
[Thread debugging using libthread_db
enabled]
Using host libthread_db library "/lib/
aarch64-linux-gnu/libthread_db.so.1".
a.out: ./test/test-failed-assert.cpp:9:
float fact(float): Assertion `x != 0.0'
failed.

Program received signal SIGABRT, Aborted.
__pthread_kill_implementation
(threadid=281474842450080,
signo=signo@entry=6, no_tid=no_tid@entry=0)
at ./nptl/pthread_kill.c:44
44          ./nptl/pthread_kill.c: No such
file or directory.
```



(gdb-ChatDBG) why

```
$ gdb ./a.out
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1
[...]
Reading symbols from ./a.out...
(gdb-ChatDBG) run
```

```
Starting program: /media/psf/Home/git/ChatDBG/
a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/aarch64-
linux-gnu/libthread_db.so.1".
a.out: ./test/test-failed-assert.cpp:9: float
fact(float): Assertion `x != 0.0' failed.

Program received signal SIGABRT, Aborted.
__pthread_kill_implementation
(threadid=281474842450080, signo=signo@entry=6,
no_tid=no_tid@entry=0) at ./nptl/
pthread_kill.c:44
44          ./nptl/pthread_kill.c: No such file
or directory.
```



```
$ gdb ./a.out
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1
[...]
Reading symbols from ./a.out...
(gdb-ChatDBG) run
```

```
Starting program: /media/psf/Home/git/ChatDBG/
a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/aarch64-
linux-gnu/libthread_db.so.1".
a.out: ./test/test-failed-assert.cpp:9: float
fact(float): Assertion `x != 0.0' failed.

Program received signal SIGABRT, Aborted.
__pthread_kill_implementation
(Threadid=281474842450080, signo=signo@entry=6,
no_tid=no_tid@entry=0) at ./nptl/
pthread_kill.c:44
44          ./nptl/pthread_kill.c: No such file
or directory.
```

(gdb-ChatDBG) why

The root cause of the error is in the `fact(float n)` function defined in the `./test/test-failed-assert.cpp` file. The `assert(x != 0.0);` is failing, causing the program to abort. **This is because the initial loop is including a zero in the multiplication.**

The erroneous line is:

```
...
for(auto i = 0.0; i < n; i++) {
    x *= i;
}
```

In this loop, `i` starts at `0.0` and `x` is initially `1.0`. On the first iteration, `x` becomes zero (`1.0 * 0.0`) and stays zero for all other iterations because multiplying by zero always results in zero.

A proposed fix would be to change the starting point of the loop to `1.0` instead of `0.0`. The corrected loop would read:

```
...
for(auto i = 1.0; i <= n; i++) {
    x *= i;
}
```

This way, `x` won't be prematurely zeroed out, enabling the computation to successfully calculate the factorial of `n` without causing an assertion error. **(Total cost: approximately \$0.03 USD.)**

```
$ gdb ./a.out
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1
[...]
Reading symbols from ./a.out...
(gdb-ChatDBG) run
```

```
Starting program: /media/psf/Home/git/ChatDBG/
a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/aarch64-
linux-gnu/libthread_db.so.1".
a.out: ./test/test-failed-assert.cpp:9: float
fact(float): Assertion `x != 0.0' failed.

Program received signal SIGABRT, Aborted.
__pthread_kill_implementation
(Threadid=281474842450080, signo=signo@entry=6,
no_tid=no_tid@entry=0) at ./nptl/
pthread_kill.c:44
44          ./nptl/pthread_kill.c: No such file
or directory.
```

(gdb-ChatDBG) why

The root cause of the error is in the `fact(float n)` function defined in the `./test/test-failed-assert.cpp` file. The `assert(x != 0.0);` is failing, causing the program to abort. This is because the initial loop is including a zero in the multiplication.

The erroneous line is:

```
...
for(auto i = 0.0; i < n; i++) {
    x *= i;
}
```

...

In this loop, `i` starts at `0.0` and `x` is initially `1.0`. On the first iteration, `x` becomes zero (`1.0 * 0.0`) and stays zero for all other iterations because multiplying by zero always results in zero.

A proposed fix would be to change the starting point of the loop to `1.0` instead of `0.0`. The corrected loop would read:

```
...
for(auto i = 1.0; i <= n; i++) {
    x *= i;
}
```

...

This way, `x` won't be prematurely zeroed out, enabling the computation to successfully calculate the factorial of `n` without causing an assertion error. (Total cost: approximately \$0.03 USD.)

```
$ gdb ./a.out
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1
[...]
Reading symbols from ./a.out...
(gdb-ChatDBG) run
```

```
Starting program: /media/psf/Home/git/ChatDBG/
a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/aarch64-
linux-gnu/libthread_db.so.1".
a.out: ./test/test-failed-assert.cpp:9: float
fact(float): Assertion `x != 0.0' failed.

Program received signal SIGABRT, Aborted.
__pthread_kill_implementation
(Threadid=281474842450080, signo=signo@entry=6,
no_tid=no_tid@entry=0) at ./nptl/
pthread_kill.c:44
44          ./nptl/pthread_kill.c: No such file
or directory.
```

(gdb-ChatDBG) why

The root cause of the error is in the `fact(float n)` function defined in the `./test/test-failed-assert.cpp` file. The `assert(x != 0.0);` is failing, causing the program to abort. **This is because the initial loop is including a zero in the multiplication.**

The erroneous line is:

```
...
for(auto i = 0.0; i < n; i++) {
    x *= i;
}
...
```

In this loop, `i` starts at `0.0` and `x` is initially `1.0`. On the first iteration, `x` becomes zero (`1.0 * 0.0`) and stays zero for all other iterations because multiplying by zero always results in zero.

A proposed fix would be to change the starting point of the loop to `1.0` instead of `0.0`. The corrected loop would read:

```
...
for(auto i = 1.0; i <= n; i++) {
    x *= i;
}
...
```

This way, `x` won't be prematurely zeroed out, enabling the computation to successfully calculate the factorial of `n` without causing an assertion error. **(Total cost: approximately \$0.03 USD.)**

```
$ gdb ./a.out
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1
[...]
Reading symbols from ./a.out...
(gdb-ChatDBG) run
```

```
Starting program: /media/psf/Home/git/ChatDBG/
a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/aarch64-
linux-gnu/libthread_db.so.1".
a.out: ./test/test-failed-assert.cpp:9: float
fact(float): Assertion `x != 0.0' failed.

Program received signal SIGABRT, Aborted.
__pthread_killImplementation
(ThreadID=281474842450080, signo=signo@entry=6,
no_tid=no_tid@entry=0) at ./nptl/
pthread_kill.c:44
44          ./nptl/pthread_kill.c: No such file
or directory.
```

(gdb-ChatDBG) why

The root cause of the error is in the `fact(float n)` function defined in the `./test/test-failed-assert.cpp` file. The `assert(x != 0.0);` is failing, causing the program to abort. **This is because the initial loop is including a zero in the multiplication.**

The erroneous line is:

```
...
for(auto i = 0.0; i < n; i++) {
    x *= i;
}
...
```

In this loop, `i` starts at `0.0` and `x` is initially `1.0`. On the first iteration, `x` becomes zero (`1.0 * 0.0`) and stays zero for all other iterations because multiplying by zero always results in zero.

A proposed fix would be to change the starting point of the loop to `1.0` instead of `0.0`. The corrected loop would read:

```
...
for(auto i = 1.0; i <= n; i++) {
    x *= i;
}
...
```

This way, `x` won't be prematurely zeroed out, enabling the computation to successfully calculate the factorial of `n` without causing an assertion error. **(Total cost: approximately \$0.03 USD.)**

```
$ gdb ./a.out
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1
[...]
Reading symbols from ./a.out...
(gdb-ChatDBG) run
```

```
Starting program: /media/psf/Home/git/ChatDBG/
a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/aarch64-
linux-gnu/libthread_db.so.1".
a.out: ./test/test-failed-assert.cpp:9: float
fact(float): Assertion `x != 0.0' failed.

Program received signal SIGABRT, Aborted.
__pthread_killImplementation
(Threadid=281474842450080, signo=signo@entry=6,
no_tid=no_tid@entry=0) at ./nptl/
pthread_kill.c:44
44          ./nptl/pthread_kill.c: No such file
or directory.
```

(gdb-ChatDBG) why

The root cause of the error is in the `fact(float n)` function defined in the `./test/test-failed-assert.cpp` file. The `assert(x != 0.0);` is failing, causing the program to abort. **This is because the initial loop is including a zero in the multiplication.**

The erroneous line is:

```
...
for(auto i = 0.0; i < n; i++) {
    x *= i;
}
...
```

In this loop, `i` starts at `0.0` and `x` is initially `1.0`. On the first iteration, `x` becomes zero (`1.0 * 0.0`) and stays zero for all other iterations because multiplying by zero always results in zero.

A proposed fix would be to change the starting point of the loop to `1.0` instead of `0.0`. The corrected loop would read:

```
...
for(auto i = 1.0; i <= n; i++) {
    x *= i;
}
...
```

This way, `x` won't be prematurely zeroed out, enabling the computation to successfully calculate the factorial of `n` without causing an assertion error. **(Total cost: approximately \$0.03 USD.)**

```
$ gdb ./a.out
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1
[...]
Reading symbols from ./a.out...
(gdb-ChatDBG) run
```

```
Starting program: /media/psf/Home/git/ChatDBG/
a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/aarch64-
linux-gnu/libthread_db.so.1".
a.out: ./test/test-failed-assert.cpp:9: float
fact(float): Assertion `x != 0.0' failed.

Program received signal SIGABRT, Aborted.
__pthread_kill_implementation
(Threadid=281474842450080, signo=signo@entry=6,
no_tid=no_tid@entry=0) at ./nptl/
pthread_kill.c:44
44          ./nptl/pthread_kill.c: No such file
or directory.
```

(gdb-ChatDBG) why

The root cause of the error is in the `fact(float n)` function defined in the `./test/test-failed-assert.cpp` file. The `assert(x != 0.0);` is failing, causing the program to abort. **This is because the initial loop is including a zero in the multiplication.**

The erroneous line is:

```
...
for(auto i = 0.0; i < n; i++) {
    x *= i;
}
...
```

In this loop, `i` starts at `0.0` and `x` is initially `1.0`. On the first iteration, `x` becomes zero (`1.0 * 0.0`) and stays zero for all other iterations because multiplying by zero always results in zero.

A proposed fix would be to change the starting point of the loop to `1.0` instead of `0.0`. The corrected loop would read:

```
...
for(auto i = 1.0; i <= n; i++) {
    x *= i;
}
...
```

This way, `x` won't be prematurely zeroed out, enabling the computation to successfully calculate the factorial of `n` without causing an assertion error. **(Total cost: approximately \$0.03 USD.)**

```
$ gdb ./a.out
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1
[...]
Reading symbols from ./a.out...
(gdb-ChatDBG) run
```

```
Starting program: /media/psf/Home/git/ChatDBG/
a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db version 2.18.1
"/lib/aarch64-
linux-gnu/libthread_db.so.1"
a.out: ./test-failed-assert.cpp:9: float
fact(float): Assertion `x != 0.0' failed.

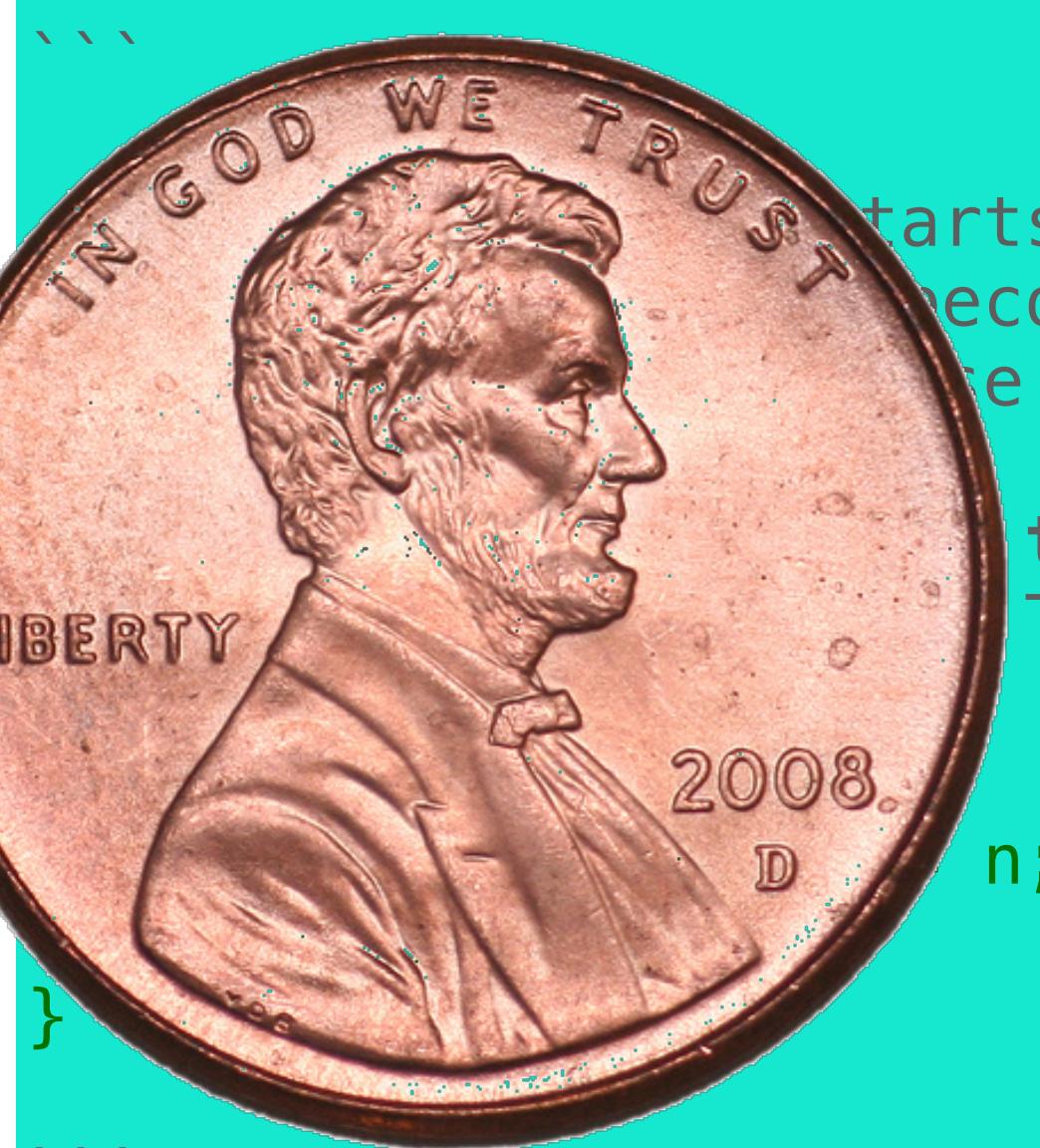
Program received signal SIGABRT, Aborted.
0x0000000000000000 in __libc_start_main@@GLIBC_2.27
(pthread_start, [tid=0x7f8d40000000], entry=0x400610, stack=0x7f8d40000000, argc=1, argv=0x7f8d40000000, env=0x7f8d40000000)
No such file or directory.
```



(gdb-ChatDBG) why
The root cause of the error is in the `fact(float n)` function defined in the `./test/test-failed-assert.cpp` file. The `assert(x != 0.0);` is failing, causing the program to abort. **This is because the initial loop is including a zero in the multiplication.**

The erroneous line is:

```
...
for(auto i = 0.0; i < n; i++) {
    x *= i;
}
```



... starts at `0.0` and becomes zero (`1.0 * 0.0 = 0.0`). On the next iteration, it's still zero for all subsequent multiplications in zero.

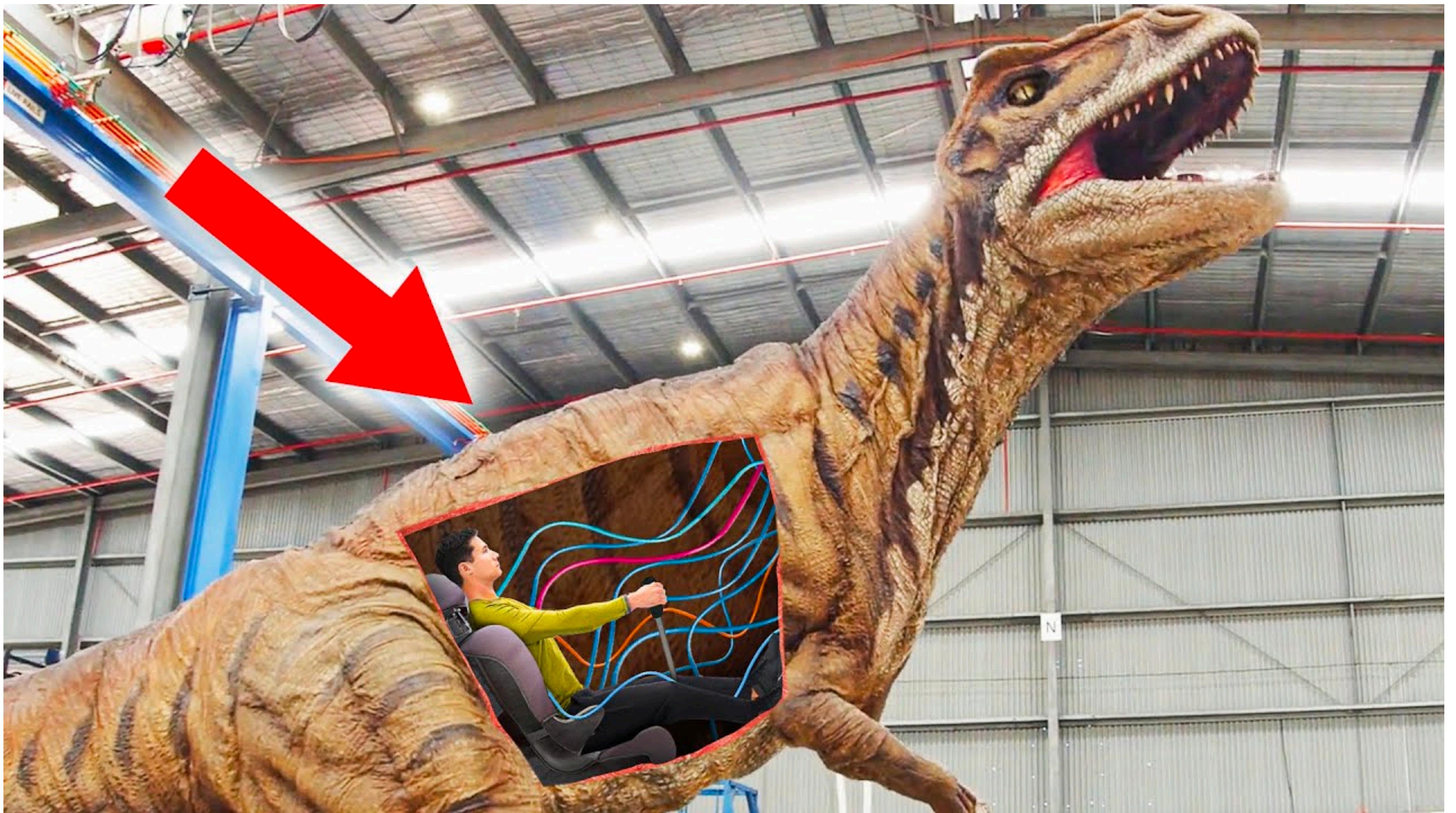
to change the value of x to something other than zero to avoid the assertion error.

The correct code is:

```
n; i++) {
```



This way, `x` won't be prematurely zeroed out, enabling the computation to successfully calculate the factorial of `n` without causing an assertion error. **(Total cost: approximately \$0.03 USD.)**





(ChatDBG llldb) why_prompt
Explain what the root cause of this error is, given the following source code context for each stack frame and a traceback, and propose a fix. In your response, never refer to the frames given below (as in, 'frame 0'). Instead, always refer only to specific lines and filenames of source code.

Source code for each stack frame:

```
```
/* frame 0 in test-failed-assert.cpp */
1 #include <assert.h>
2 #include <iostream>
3
4 float fact(float n) {
5 auto x = 1.0;
6 for (auto i = 0.0; i < n; i++) {
7 x *= i;
8 }
9 assert(x != 0.0);
-----^

/* frame 1 in test-failed-assert.cpp */
6 for (auto i = 0.0; i < n; i++) {
7 x *= i;
8 }
9 assert(x != 0.0);
10 return x;
11 }
12
13
14 int main()
15 {
16 std::cout << fact(100) << std::endl;
-----^
```

```
```
frame 0: fact((float) n = 100) at test-failed-
assert.cpp:9:3
```

Local variables:

frame 1: main() at test-failed-assert.cpp:16:16

Local variables:

stop reason = hit program assert

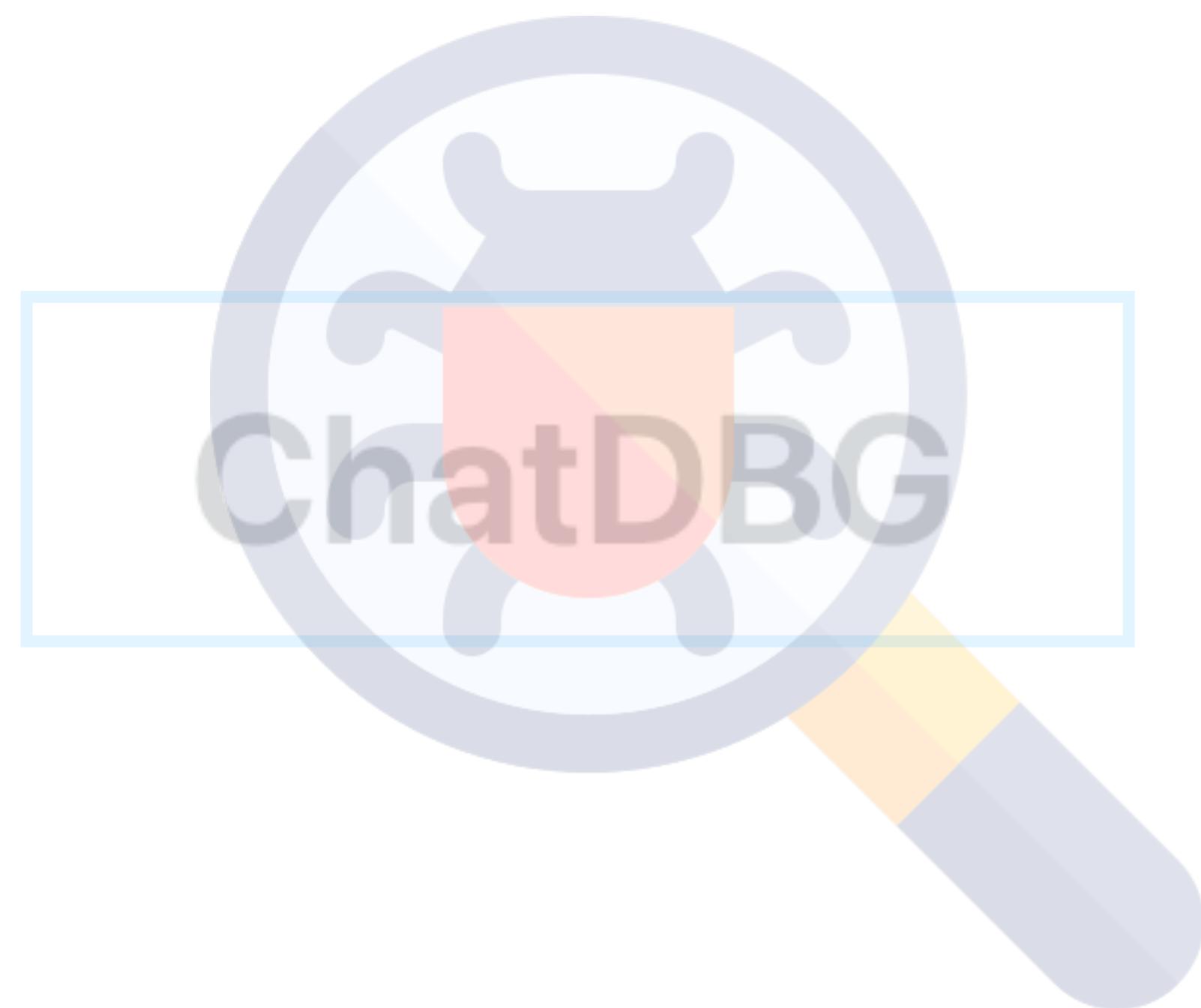


evolve



evolve

debugger that performs root cause
analysis & proposes fixes



evolve

debugger that performs root cause
analysis & proposes fixes

exploit
niche



evolve

debugger that performs root cause analysis & proposes fixes

exploit
niche

can produce stack trace "on steroids"



evolve

debugger that performs root cause analysis & proposes fixes

exploit
niche



can produce stack trace "on steroids"
include relevant source,
variable values, dynamic types, deref'd pointers

evolve

debugger that performs root cause analysis & proposes fixes

exploit
niche

can produce stack trace "on steroids"
include relevant source,
variable values, dynamic types, deref'd pointers

ensure
fitness



evolve

debugger that performs root cause analysis & proposes fixes

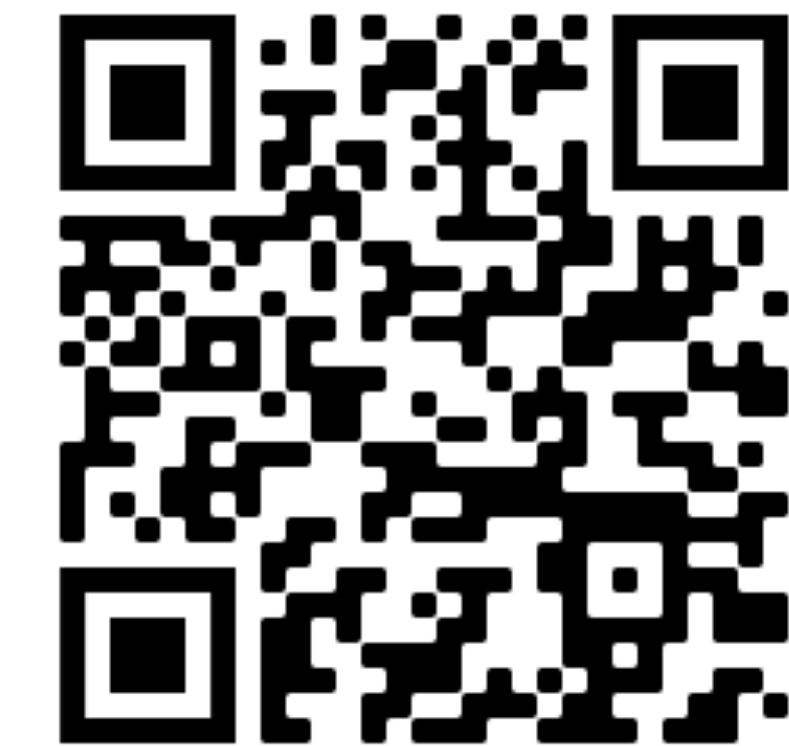
exploit
niche

can produce stack trace "on steroids"
include relevant source,
variable values, dynamic types, deref'd pointers

ensure
fitness

patch code, restart, verify that fix prevents crash / exception... (TBD)





CWhy

by [Emery Berger](#), [Nicolas van Kempen](#), and [Bryce Adelstein Lelbach](#).

[pypi v0.3.1](#) [downloads 4k](#) [downloads/month 731](#)

("See why")

Explains and suggests fixes for compiler error messages for a wide range of programming languages, including C, C++, C#, Go, Java, LaTeX, PHP, Python, Ruby, Rust, Swift, and TypeScript.



C++

This highlighted example is [missing-hash.cpp](#), which is one of the first cases we experimented with.

▼ Expand to see the original (pretty obscure) error message:

CW

CWhy

by Emery E

pypi v0.3.1

("See why"

Explains an
including C

C++

This highlighted example is [missing-hash.cpp](#), which is one of the first cases we experimented with.

▼ Expand to see the original (pretty obscure) error message:

```
% clang++ --std=c++20 -c missing-hash.cpp
missing-hash.cpp:13:45: error: call to implicitly-deleted default constructor of 'std::unordered_set<std::pair<int, int>>' visited;
    std::unordered_set<std::pair<int, int>> visited;
                                         ^
/usr/lib/gcc/x86_64-linux-gnu/10/../../../../../../../../include/c++/10/bits/unordered_set.h:135:7: note: explicitly
      unordered_set() = default;
      ^
/usr/lib/gcc/x86_64-linux-gnu/10/../../../../../../../../include/c++/10/bits/unordered_set.h:100:18: note: default co
      _Hashtable _M_h;
      ^
/usr/lib/gcc/x86_64-linux-gnu/10/../../../../../../../../include/c++/10/bits/hashtable.h:451:7: note: explicitly
      _Hashtable() = default;
      ^
/usr/lib/gcc/x86_64-linux-gnu/10/../../../../../../../../include/c++/10/bits/hashtable.h:174:7: note: default co
      : public __detail::_Hashtable_base<_Key, _Value, _ExtractKey, _Equal,
      ^
/usr/lib/gcc/x86_64-linux-gnu/10/../../../../../../../../include/c++/10/bits/hashtable_policy.h:1791:5: note: ex
      _Hashtable_base() = default;
      ^
/usr/lib/gcc/x86_64-linux-gnu/10/../../../../../../../../include/c++/10/bits/hashtable_policy.h:1726:5: note: de
      : public _Hash_code_base<_Key, _Value, _ExtractKey, _H1, _H2, _Hash,
      ^
/usr/lib/gcc/x86_64-linux-gnu/10/../../../../../../../../include/c++/10/bits/hashtable_policy.h:1368:7: note: ex
      _Hash_code_base() = default;
      ^
/usr/lib/gcc/x86_64-linux-gnu/10/../../../../../../../../include/c++/10/bits/hashtable_policy.h:1344:7: note: de
      private _Hashtable_ebo_helper<1, _H1>,
      ^
```

C++

This highlighted example is [missing-hash.cpp](#), which is one of the first cases we experimented with.

▼ Expand to see the original (pretty obscure) error message:

```
/usr/lib/gcc/x86_64-linux-gnu/10/../../../../include/c++/10/bits/hashtable_policy.h:1112:7: note: ex
      _Hashtable_ebo_helper() = default;
      ^
/usr/lib/gcc/x86_64-linux-gnu/10/../../../../include/c++/10/bits/hashtable_policy.h:1110:7: note: de
      : private _Tp
      ^
/usr/lib/gcc/x86_64-linux-gnu/10/../../../../include/c++/10/bits/functional_hash.h:101:19: note: de
      struct hash : __hash_enum<_Tp>
      ^
In file included from missing-hash.cpp:1:
In file included from /usr/lib/gcc/x86_64-linux-gnu/10/../../../../include/c++/10/functional:61:
In file included from /usr/lib/gcc/x86_64-linux-gnu/10/../../../../include/c++/10/unordered_map:46:
In file included from /usr/lib/gcc/x86_64-linux-gnu/10/../../../../include/c++/10/bits/hashtable.h:3:
/usr/lib/gcc/x86_64-linux-gnu/10/../../../../include/c++/10/bits/hashtable_policy.h:1377:2: error: s
      static_assert(__is_invocable<const _H1&, const _Key>{},
      ^
      ~~~~~
/usr/lib/gcc/x86_64-linux-gnu/10/../../../../include/c++/10/bits/hashtable.h:1675:29: note: in insta
      __hash_code __code = this->_M_hash_code(_k);
      ^
/usr/lib/gcc/x86_64-linux-gnu/10/../../../../include/c++/10/bits/hashtable.h:788:11: note: in instar
      { return _M_emplace(__unique_keys(), std::forward<_Args>(__args)...); }
      ^
/usr/lib/gcc/x86_64-linux-gnu/10/../../../../include/c++/10/bits/unordered_set.h:377:16: note: in ir
      { return _M_h.emplace(std::forward<_Args>(__args)...); }
      ^
missing-hash.cpp:20:44: note: in instantiation of function template specialization 'std::unordered_s
      const auto [_, inserted] = visited.emplace(n->position);
      ^
```

C++

This highlighted example is [missing-hash.cpp](#), which is one of the first cases we experimented with.

▼ Expand to see the original (pretty obscure) error message:

```
In file included from missing-hash.cpp:1:  
In file included from /usr/lib/gcc/x86_64-linux-gnu/10/../../../../include/c++/10/functional:61:  
In file included from /usr/lib/gcc/x86_64-linux-gnu/10/../../../../include/c++/10/unordered_map:46:  
In file included from /usr/lib/gcc/x86_64-linux-gnu/10/../../../../include/c++/10/bits/hashtable.h:3  
/usr/lib/gcc/x86_64-linux-gnu/10/../../../../include/c++/10/bits/hashtable_policy.h:1379:9: error: t  
    return _M_h1().__k);  
           ^~~~~~  
3 errors generated.
```

CWhy

by Emery E

pypi v0.3.1

("See why"

Explains an
including C

And here's the English-language explanation from `cwhy`:



```
% clang++ --std=c++20 -c missing-hash.cpp |& cwhy
The problem is that you are attempting to use `std::unordered_set` with `std::pair<int, int>` as the key type. However, the standard library does not provide a hash function specialization for `std::pair` out of the box, so the default constructor of the unordered set is deleted.
```

To resolve this, you'll need to provide a custom hash function for `std::pair<int, int>`. Here's an example of how you can define one:

```
```cpp
struct PairHash {
 template <typename T1, typename T2>
 std::size_t operator()(const std::pair<T1, T2>& pair) const {
 std::hash<T1> hash1;
 std::hash<T2> hash2;
 return hash1(pair.first) ^ (hash2(pair.second) << 1);
 }
};
```

Then, when instantiating the `std::unordered\_set`, you can specify the custom hash function:

```
```cpp
std::unordered_set<std::pair<int, int>, PairHash> visited;
```
```

With this change, the code should now compile and work as expected.

("See why")

Explains and suggests  
including C, C++, C#

```
void f(void*) {}
void f(char*) {}

template <typename T>
auto g(T t) -> decltype(f(t)) {
 return f(t);
}

int main() {
 g(nullptr);
}
```

```
void f(void*) {}
void f(char*) {}

template <typename T>
auto g(T t) -> decltype(f(t)) {
 return f(t);
}

int main() {
 g(nullptr);
}
```

```
% clang++ -std=c++20 tests/c++/sfinae-ambiguous.cpp
tests/c++/sfinae-ambiguous.cpp:10:5: error: no
matching function for call to 'g'
 g(nullptr);

tests/c++/sfinae-ambiguous.cpp:5:6: note: candidate
template ignored: substitution failure [with T =
std::nullptr_t]: call to 'f' is ambiguous
auto g(T t) -> decltype(f(t)) {

1 error generated.
```

```
void f(void*) {}
void f(char*) {}

template <typename T>
auto g(T t) -> decltype(f(t)) {
 return f(t);
}

int main() {
 g(nullptr);
}
```

```
% clang++ -std=c++20 tests/c++/sfinae-ambiguous.cpp
tests/c++/sfinae-ambiguous.cpp:10:5: error: no matching
function for call to 'g'
 g(nullptr);
^
tests/c++/sfinae-ambiguous.cpp:5:6: note: candidate template
ignored: substitution failure [with T = std::nullptr_t]:
call to 'f' is ambiguous
auto g(T t) -> decltype(f(t)) {
^
1 error generated.
```

```
% `cwhy --wrapper` -std=c++20 tests/c++/sfinae-ambiguous.cpp
```

```
void f(void*) {}
void f(char*) {}

template <typename T>
auto g(T t) -> decltype(f(t)) {
 return f(t);
}

int main() {
 g(nullptr);
}
```

```
% clang++ -std=c++20 tests/c++/sfinae-ambiguous.cpp
tests/c++/sfinae-ambiguous.cpp:10:5: error: no matching
function for call to 'g'
 g(nullptr);

tests/c++/sfinae-ambiguous.cpp:5:6: note: candidate template
ignored: substitution failure [with T = std::nullptr_t]:
call to 'f' is ambiguous
auto g(T t) -> decltype(f(t)) {

1 error generated.
```

```
% `cwhy --wrapper` -std=c++20 tests/c++/sfinae-ambiguous.cpp
```

The problem here lies on the call to function `f(t)` inside the `decltype` expression in your template function `g()`.

In C++, `nullptr` is of type `std::nullptr\_t` and it can be implicitly converted to any pointer type. Here `f` has overloads for `void\*` and `char\*`, which both are valid conversions for `nullptr`, so when you call `f(t)`, where `t` is `nullptr`, the compiler isn't sure which overload to choose, because both `void\*` and `char\*` are equally viable. Hence, the error: "call to 'f' is ambiguous"

The ambiguity of these two function calls is not something that the SFINAE (Substitution Failure Is Not An Error) mechanism can handle, so it triggers a full-blown compile error instead of simply discarding this instantiation.

To fix this, you could add a specific overload for `std::nullptr\_t`:

```
```cpp
void f(std::nullptr_t) {}

template <typename T>
auto g(T t) -> decltype(f(t)) {
    return f(t);
}

int main() {
    g(nullptr);
}
```

```

Or you could implement a solution that makes one of the overloads a better match for `nullptr`:

```
```cpp
void f(void*) {}
void f(char*&) {} // 'char&' makes this overload less preferred for
nullptr than the 'void*'

template <typename T>
auto g(T t) -> decltype(f(t)) {
    return f(t);
}

int main() {
    g(nullptr);
}
```

```

(Total cost: approximately \$0.03 USD.)

```
void f(void*) {}
void f(char*) {}

template <typename T>
auto g(T t) -> decltype(f(t)) {
 return f(t);
}

int main() {
 g(nullptr);
}
```

```
% clang++ -std=c++20 tests/c++/sfinae-ambiguous.cpp
tests/c++/sfinae-ambiguous.cpp:10:5: error: no matching
function for call to 'g'
 g(nullptr);

tests/c++/sfinae-ambiguous.cpp:5:6: note: candidate template
ignored: substitution failure [with T = std::nullptr_t]:
call to 'f' is ambiguous
auto g(T t) -> decltype(f(t)) {

1 error generated.
```

```
% `cwhy --wrapper` -std=c++20 tests/c++/sfinae-ambiguous.cpp
```

The problem here lies on the call to function `f(t)` inside the `decltype` expression in your template function `g()`.

In C++, `nullptr` is of type `std::nullptr\_t` and it can be implicitly converted to any pointer type. Here `f` has overloads for `void\*` and `char\*`, which both are valid conversions for `nullptr`, so when you call `f(t)`, where `t` is `nullptr`, the compiler isn't sure which overload to choose, because both `void\*` and `char\*` are equally viable. Hence, the error: "call to 'f' is ambiguous"

The ambiguity of these two function calls is not something that the SFINAE (Substitution Failure Is Not An Error) mechanism can handle, so it triggers a full-blown compile error instead of simply discarding this instantiation.

To fix this, you could add a specific overload for `std::nullptr\_t`:

```
```cpp
void f(std::nullptr_t) {}

template <typename T>
auto g(T t) -> decltype(f(t)) {
    return f(t);
}

int main() {
    g(nullptr);
}
```

```

Or you could implement a solution that makes one of the overloads a better match for `nullptr`:

```
```cpp
void f(void*) {}
void f(char*&) {} // 'char&' makes this overload less preferred for
nullptr than the 'void*'

template <typename T>
auto g(T t) -> decltype(f(t)) {
    return f(t);
}

int main() {
    g(nullptr);
}
```

```

(Total cost: approximately \$0.03 USD.)

```
void f(void*) {}
void f(char*) {}

template <typename T>
auto g(T t) -> decltype(f(t)) {
 return f(t);
}

int main() {
 g(nullptr);
}
```

```
% clang++ -std=c++20 tests/c++/sfinae-ambiguous.cpp
tests/c++/sfinae-ambiguous.cpp:10:5: error: no matching
function for call to 'g'
 g(nullptr);
^
tests/c++/sfinae-ambiguous.cpp:5:6: note: candidate template
ignored: substitution failure [with T = std::nullptr_t]:
call to 'f' is ambiguous
auto g(T t) -> decltype(f(t)) {
^
1 error generated.
```

```
% `cwhy --wrapper` -std=c++20 tests/c++/sfinae-ambiguous.cpp
```

The problem here lies on the call to function `f(t)` inside the `decltype` expression in your template function `g()`.

In C++, `nullptr` is of type `std::nullptr\_t` and it can be implicitly converted to any pointer type. Here `f` has overloads for `void\*` and `char\*`, which both are valid conversions for `nullptr`, so when you call `f(t)`, where `t` is `nullptr`, the compiler isn't sure which overload to choose, because both `void\*` and `char\*` are equally viable. Hence, the error: "call to 'f' is ambiguous"

The ambiguity of these two function calls is not something that the SFINAE (Substitution Failure Is Not An Error) mechanism can handle, so it triggers a full-blown compile error instead of simply discarding this instantiation.

To fix this, you could add a specific overload for `std::nullptr\_t`:

```
```cpp
void f(std::nullptr_t) {}

template <typename T>
auto g(T t) -> decltype(f(t)) {
    return f(t);
}

int main() {
    g(nullptr);
}
```

```

Or you could implement a solution that makes one of the overloads a better match for `nullptr`:

```
```cpp
void f(void*) {}
void f(char*&) {} // 'char&' makes this overload less preferred for
nullptr than the 'void*'

template <typename T>
auto g(T t) -> decltype(f(t)) {
    return f(t);
}

int main() {
    g(nullptr);
}
```

```

(Total cost: approximately \$0.03 USD.)

```
void f(void*) {}
void f(char*) {}

template <typename T>
auto g(T t) -> decltype(f(t)) {
 return f(t);
}

int main() {
 g(nullptr);
}
```

```
% clang++ -std=c++20 tests/c++/sfinae-ambiguous.cpp
tests/c++/sfinae-ambiguous.cpp:10:5: error: no matching
function for call to 'g'
 g(nullptr);
^

tests/c++/sfinae-ambiguous.cpp:5:6: note: candidate template
ignored: substitution failure [with T = std::nullptr_t]:
call to 'f' is ambiguous
auto g(T t) -> decltype(f(t)) {
^
1 error generated.
```

```
% `cwhy --wrapper` -std=c++20 tests/c++/sfinae-ambiguous.cpp
```

The problem here lies on the call to function `f(t)` inside the `decltype` expression in your template function `g()`.

In C++, `nullptr` is of type `std::nullptr\_t` and it can be implicitly converted to any pointer type. Here `f` has overloads for `void\*` and `char\*`, which both are valid conversions for `nullptr`, so when you call `f(t)`, where `t` is `nullptr`, the compiler isn't sure which overload to choose, because both `void\*` and `char\*` are equally viable. Hence, the error: "call to 'f' is ambiguous"

The ambiguity of these two function calls is not something that the SFINAE (Substitution Failure Is Not An Error) mechanism can handle, so it triggers a full-blown compile error instead of simply discarding this instantiation.

To fix this, you could add a specific overload for `std::nullptr\_t`:

```
```cpp
void f(std::nullptr_t) {}

template <typename T>
auto g(T t) -> decltype(f(t)) {
    return f(t);
}

int main() {
    g(nullptr);
}
```

```

Or you could implement a solution that makes one of the overloads a better match for `nullptr`:

```
```cpp
void f(void*) {}
void f(char*&) {} // 'char&' makes this overload less preferred for
nullptr than the 'void*'

template <typename T>
auto g(T t) -> decltype(f(t)) {
    return f(t);
}

int main() {
    g(nullptr);
}
```

```

```
void f(void*) {}
void f(char*) {}

template <typename T>
auto g(T t) -> decltype(f(t)) {
 return f(t);
}

int main() {
 g(nullptr);
}
```

```
% clang++ -std=c++20 tests/c++/sfinae-ambiguous.cpp
tests/c++/sfinae-ambiguous.cpp:10:5: error: no matching
function for call to 'g'
 g(nullptr);

tests/c++/sfinae-ambiguous.cpp:5:6: note: candidate template
ignored: substitution failure [with T = std::nullptr_t]:
call to 'f' is ambiguous
auto g(T t) -> decltype(f(t)) {

1 error generated.
```

```
% `cwhy --wrapper` -std=c++20 tests/c++/sfinae-ambiguous.cpp
```

The problem here lies on the call to function `f(t)` inside the `decltype` expression in your template function `g()`.

In C++, `nullptr` is of type `std::nullptr\_t` and it can be implicitly converted to any pointer type. Here `f` has overloads for `void\*` and `char\*`, which both are valid conversions for `nullptr`, so when you call `f(t)`, where `t` is `nullptr`, the compiler isn't sure which overload to choose, because both `void\*` and `char\*` are equally viable. Hence, the error: "call to 'f' is ambiguous"

The ambiguity of these two function calls is not something that the SFINAE (Substitution Failure Is Not An Error) mechanism can handle, so it triggers a full-blown compile error instead of simply discarding this instantiation.

To fix this, you could add a specific overload for `std::nullptr\_t`:

```
```cpp
void f(std::nullptr_t) {}

template <typename T>
auto g(T t) -> decltype(f(t)) {
    return f(t);
}

int main() {
    g(nullptr);
}
```

```

Or you could implement a solution that makes one of the overloads a better match for `nullptr`:

```
```cpp
void f(void*) {}
void f(char*&) {} // 'char&' makes this overload less preferred for
nullptr than the 'void*'

template <typename T>
auto g(T t) -> decltype(f(t)) {
    return f(t);
}

int main() {
    g(nullptr);
}
```

```

```
void f(void*) {}
void f(char*) {}

template <typename T>
auto g(T t) -> decltype(f(t)) {
 return f(t);
}

int main() {
 g(nullptr);
}
```

The problem here lies on the call to function `f(t)` inside the `decltype` expression in your template function `g()`.

In C++, `nullptr` is of type `std::nullptr\_t` and it can be implicitly converted to any pointer type. Here `f` has overloads for `void\*` and `char\*`, which both are valid conversions for `nullptr`, so when you call `f(t)`, where `t` is `nullptr`, the compiler isn't sure which overload to choose, because both `void\*` and `char\*` are equally viable. Hence, the error: "call to 'f' is ambiguous"

The ambiguity of these two function calls is not something that the SFINAE (Substitution Failure Is Not An Error) mechanism can handle, so it triggers a full-blown compile error instead of simply discarding this instantiation.

To fix this, you could add a specific overload for `std::nullptr\_t`:

```
% clang++ -std=c++14 tests/c++/sfinae-ambiguous.cpp
error: no matching function for call to 'f'
 g(nullptr)
          ~~~~~^
tests/c++/sfinae-ambiguous.cpp:10:10: note: candidate is not viable: requires
      argument of type 'std::nullptr_t' but none was provided
    void f(std::nullptr_t) {}
             ^~~~~~
```



```
% `cwhy --wrapper` -std=c++20 tests/c++/sfinae-ambiguous.cpp
```

```
void f(void*) {}
void f(char*&) // 'char&' makes this overload less preferred for
                // nullptr than the 'void*'

template <typename T>
auto g(T t) -> decltype(f(t)) {
    return f(t);
}

int main() {
    g(nullptr);
}
```


```
IN GOD WE TRUST
LIBERTY
2008. D
IN GOD WE TRUST
LIBERTY
2008. D
IN GOD WE TRUST
LIBERTY
2008. D
```


```

(Total cost: approximately \$0.03 USD.)

HED

CppCon 2023

30 MDT

 [Expressive Compile-time Parsers](#)  
[Alon Wolf](#)

 [Give Feedback](#)

rsday, October

30 MDT

[Add to My Schedule](#)

 [Link](#)

#### Expressive Compile-time Parsers

Modern C++ has many language features that can be leveraged to design an expressive API or a domain-specific language. However, these features are limited by the syntax rules of C++. To overcome this limitation, we will discuss the use of compile-time parsers to write more expressive code with zero overhead.

We will analyze open-source compile-time parsing libraries from C++11 to C++23 and compare their APIs in terms of flexibility and expressiveness. Additionally, we will delve into the techniques used in the implementation of these libraries to handle compile-time parsing under the constraints of compile-time execution.

Finally, we will highlight how these libraries can be used to

#### Speakers



[Alon Wolf](#)

Software Engineer, Medtronic



C++ source #1 X

A - + v 🔍 🎨

C++ C++

```
1 #include <ctre.hpp>
2 #include <optional>
3
4 std::optional<std::string_view> extract_number(std::string s) {
5     using namespace ctre::literals;
6     if (auto m = ctre::match<"^[-z]+([0-9]+)$">(s)) {
7         return m.get<1>().to_view();
8     } else {
9         return std::nullopt;
10    }
11 }
```

12



<https://gcc.godbolt.org/z/a5Kavxjab>

C++ source #1 C++ x86-64 gcc (trunk) (Editor #1) Output of x86-64 gcc (trunk) (Compiler #1)

A + C++ x86-64 gcc (trunk)

```
1 #include <ctre.hpp>
2 #include <optional>
3
4 std::optional<std::string_view> extract_number(std::string s)
5     using namespace ctre::literals;
6     if (auto m = ctre::match<"^[-z]+([0-9]++)$">(s)) {
7         return m.get<1>().to_view();
8     } else {
9         return std::nullopt;
10    }
11 }
```

-std=c++2a -O3 -g0

A

1 <Compilation failed>
2
3 # For more information, run with --diag-detail=1

C Output (0/102) x86-64 gcc  
(trunk) i - cached

Compiler License

In file included from /opt/compiler-explorer/libs/ctre/main/include/ctre/literals.hpp:7,
 from /opt/compiler-explorer/libs/ctre/main/include/ctre.hpp:4,
 from <source>:1:
/opt/compiler-explorer/libs/ctre/main/include/ctre/wrapper.hpp: In instantiation of 'struct ctre::problem\_at\_position<size\_t>':
<source>:6:24: required from here
/opt/compiler-explorer/libs/ctre/main/include/ctre/wrapper.hpp:306:100: required from 'const ctre::problem\_at\_position<size\_t>' used by 'ctre::return\_type<size\_t>::operator()'
<source>:6:24: required from here
/opt/compiler-explorer/libs/ctre/main/include/ctre/wrapper.hpp:299:42: error: invalid use of incomplete type 'ctre::problem\_at\_position<size\_t>'; did you mean 'ctre::problem\_at\_position<const char\*>'?
299 | static\_assert(result::is\_correct && problem\_at\_position<n>{}, "Regular Expression Match Failed");
 |
<source>: In function 'std::optional<std::basic\_string\_view<char> > extract\_number(std::string s)'
<source>:7:24: error: no matching function for call to 'ctre::regex\_results<const char\*>::get<1>()
7 | return m.get<1>().to\_view();
 |
In file included from /opt/compiler-explorer/libs/ctre/main/include/ctre/evaluation.hpp:9,
 from /opt/compiler-explorer/libs/ctre/main/include/ctre/literals.hpp:6:
<source>: In function 'ctre::return\_type<size\_t> extract\_number(std::string s)'
<source>:305:136: note: candidate: 'template <size\_t Id, typename = std::enable\_if\_t<decltype(\_captures)>>::template return\_type<size\_t> ctre::return\_type<size\_t>::operator()<Id, \_captures>() const'
305 | template <size\_t Id, typename = std::enable\_if\_t<decltype(\_captures)>>::template
 |
<source>: In function 'ctre::return\_type<size\_t> extract\_number(std::string s)'
<source>:305:136: note: template argument deduction/substitution failed
In file included from /opt/compiler-explorer/gcc-trunk-20231002/include/c++/14.0.0/bits/stl\_pair.h:42,
 from /opt/compiler-explorer/gcc-trunk-20231002/include/c++/14.0.0/utility:69,
 from /opt/compiler-explorer/libs/ctre/main/include/ctll/fixed\_string.hpp:4,
 from /opt/compiler-explorer/libs/ctre/main/include/ctll/parser.hpp:4,
 from /opt/compiler-explorer/libs/ctre/main/include/ctll.hpp:4,
 from /opt/compiler-explorer/libs/ctre/main/include/ctre/literals.hpp:4:
<source>: In file included from /opt/compiler-explorer/gcc-trunk-20231002/include/c++/14.0.0/type\_traits: In substitution of 'std::enable\_if<true>':
<source>: In file included from /opt/compiler-explorer/libs/ctre/main/include/ctre/return\_type.hpp:305:23: required from here
<source>: In file included from /opt/compiler-explorer/gcc-trunk-20231002/include/c++/14.0.0/type\_traits:2623:11: error: no type named 'enable\_if\_t'
2623 | using enable\_if\_t = typename enable\_if<\_Cond, \_Tp>::type;
 |
<source>: In file included from /opt/compiler-explorer/libs/ctre/main/include/ctre/return\_type.hpp:308:142: note: candidate: 'template <typename Name, typename = std::enable\_if\_t<decltype(\_captures)>>::template return\_type<size\_t> ctre::return\_type<size\_t>::operator()<Name, \_captures>() const'
308 | template <typename Name, typename = std::enable\_if\_t<decltype(\_captures)>>::template
 |
<source>: In file included from /opt/compiler-explorer/libs/ctre/main/include/ctre/return\_type.hpp:308:142: note: template argument deduction/substitution failed
<source>:7:24: error: type/value mismatch at argument 1 in template parameter list for 'template <typename Name, typename = std::enable\_if\_t<decltype(\_captures)>>::template return\_type<size\_t> ctre::return\_type<size\_t>::operator()<Name, \_captures>() const'
7 | return m.get<1>().to\_view();
 |
<source>:7:24: note: expected a type, got '1'
<source>: In file included from /opt/compiler-explorer/libs/ctre/main/include/ctre/return\_type.hpp:312:152: note: candidate: 'template <ctll::fixed\_string Name, typename = std::enable\_if\_t<decltype(\_captures)>>::template return\_type<size\_t> ctre::return\_type<size\_t>::operator()<Name, \_captures>() const'
312 | template <ctll::fixed\_string Name, typename = std::enable\_if\_t<decltype(\_captures)>>::template
 |



C++ source #1 X

A - + v

C++ CWhy

```
1 #include <ctre.hpp>
2 #include <optional>
3
4 std::optional<std::string_view> extract_number(std::string_view s) noexcept {
5     using namespace ctre::literals;
6     if (auto m = ctre::match<"^a--z++([0-9]++)$">(s)) {
7         return m.get<1>().to_view();
8     } else {
9         return std::nullopt;
10    }
11 }
```

Your regular expression `"<sup>1</sup>^a--z++([0-9]++)\$` is invalid.

Your intention here appears to be to match any string which starts with one or more lowercase letters and ends with one or more digits. However, your character class range `a--z` seems to be incorrectly typed. In regular expressions, a single `--` is used to represent a range, so this should be `a-z` instead:

So the corrected regular expression should look like this:

```
```cpp
"^[a-z]++([0-9]++)$"
```

```

The corrected code should then be:

```
```cpp
1 #include <ctre.hpp>
2 #include <optional>
3
4 std::optional<std::string_view> extract_number(std::string_view s) noexcept {
5     using namespace ctre::literals;
6     if (auto m = ctre::match<"^a-z++([0-9]++)$">(s)) {
7         return m.get<1>().to_view();
8     } else {
9         return std::nullopt;
10    }
11 }
```

```

When your original code tried to compile, the compiler tried to compile an instantiation of `problem\_at\_position<4>`, in other words it was trying to tell you there was a problem at the 4th character in your regular expression, which is where your `--` is located. Once you fix your regular expression to `"<sup>1</sup>^a-z++([0-9]++)\$` the code should compile without issues.

This kind of messages can help you debug your regular expressions in the future by helping you locate problematic parts of your regular expression. So when you see `problem\_at\_position<n>`, you can look at the nth character in your regular expression for potential issues.

(Total cost: approximately \$0.06 USD.)

e/main/include/ctre/literals.hpp:7,  
e/main/include/ctre.hpp:4,  
  
upper.hpp: In instantiation of 'struct ctre::problem\_at\_position<n>':  
upper.hpp:306:100: required from 'const ctre::problem\_at\_position<n> &operator<<(...)'  
  
upper.hpp:299:42: error: invalid use of incomplete type 'ctre::problem\_at\_position<n>'; did you mean 'problem\_at\_position<n>{}'? "Regular Expression Error"  
  
upper.hpp:277:26: note: declaration of 'std::string &operator<<(...)'  
// do not define!  
  
g\_view<char> > extract\_number(std::string\_view s) const { auto m = ctre::regex\_results<const char\*>::get<1>(s);  
  
e/main/include/ctre/evaluation.hpp:9,  
e/main/include/ctre/literals.hpp:6:  
turn\_type.hpp:305:136: note: candidate: 'std::enable\_if\_t<decltype(\_captures)>::template turn\_type<...>'  
turn\_type.hpp:305:136: note: template alias defined in /usr/lib/gcc/x86\_64-pc-linux-gnu/10.2.0/include/c++/v1/utility  
k-20231002/include/c++/14.0.0/bits/stl\_parallel.h  
k-20231002/include/c++/14.0.0/utility:69,  
e/main/include/ctll/fixed\_string.hpp:4,  
e/main/include/ctll/parser.hpp:4,  
e/main/include/ctll.hpp:4,  
e/main/include/ctre/literals.hpp:4:  
/14.0.0/type\_traits: In substitution of 'std::enable\_if\_t<decltype(\_captures)>::template turn\_type<...>':  
turn\_type.hpp:305:23: required from here  
/14.0.0/type\_traits:2623:11: error: no type named '\_Tp' in 'std::enable\_if\_t<decltype(\_captures)>::template turn\_type<...>'  
  
turn\_type.hpp:308:142: note: candidate: 'std::enable\_if\_t<decltype(\_captures)>::template turn\_type<...>'  
turn\_type.hpp:308:142: note: template alias defined in /usr/lib/gcc/x86\_64-pc-linux-gnu/10.2.0/include/c++/v1/utility  
1 in template parameter list for 'template class std::enable\_if\_t<decltype(\_captures)>::template turn\_type<...>'  
  
turn\_type.hpp:312:152: note: candidate: 'std::enable\_if\_t<decltype(\_captures)>::template turn\_type<...>'  
pename = std::enable\_if\_t<decltype(\_captures)>::template turn\_type<...>::pename



C++ source #1

```
1 #include <ctre.hpp>
2 #include <optional>
3
4 std::optional<std::string> Ctre::GetDefaultName() {
5     using namespace std::literals::string_literals;
6     if (auto m = ctre::GetDefaultName()) {
7         return m.get();
8     } else {
9         return std::nullopt;
10    }
11 }
12 }
```



<https://gcc.godbolt.org/z/a5Kavxjab>

C++ source #1 X

A - + v S C++ CWhy

```
1 #include <ctre.hpp>
2 #include <optional>
3
4 std::optional<std::string_view> extract_number(std::string_view s) noexcept {
5     using namespace ctre::literals;
6     if (auto m = ctre::match<"^a--z++([0-9]++)$">(s)) {
7         return m.get<1>().to_view();
8     } else {
9         return std::nullopt;
10    }
11 }
```

Your regular expression `"<sup>1</sup>a--z++([0-9]++)\$"` is invalid.

Your intention here appears to be to match any string which starts with one or more lowercase letters and ends with one or more digits. However, your character class range `a--z` seems to be incorrectly typed. In regular expressions, a single `--` is used to represent a range, so this should be `a-z` instead:

So the corrected regular expression should look like this:

```
```cpp
"^[a-z]++([0-9]++)$"
```

```

The corrected code should then be:

```
```cpp
1 #include <ctre.hpp>
2 #include <optional>
3
4 std::optional<std::string_view> extract_number(std::string_view s) noexcept {
5     using namespace ctre::literals;
6     if (auto m = ctre::match<"^a-z++([0-9]++)$">(s)) {
7         return m.get<1>().to_view();
8     } else {
9         return std::nullopt;
10    }
```

```

When your original code tried to compile, the compiler tried to compile an instantiation of `problem\_at\_position<4>`, in other words it was trying to tell you there was a problem at the 4th character in your regular expression, which is where your `--` is located. Once you fix your regular expression to `"<sup>1</sup>^a-z++([0-9]++)\$"` the code should compile without issues.

This kind of messages can help you debug your regular expressions in the future by helping you locate problematic parts of your regular expression. So when you see `problem\_at\_position<n>`, you can look at the nth character in your regular expression for potential issues.

(Total cost: approximately \$0.06 USD.)

<https://gcc.godbolt.org/z/a5Kavxjab>



C++ source #1 X

A - + v S C++

```
1 #include <ctre.hpp>
2 #include <optional>
3
4 std::optional<std::string_view> extract_number(std::string_view s) noexcept {
5     using namespace ctre::literals;
6     if (auto m = ctre::match<"^a--z++([0-9]++)$>(s)) {
7         return m.get<1>().to_view();
8     } else {
9         return std::nullopt;
10    }
11 }
```

=====  
CWhy  
=====

Your regular expression `"<sup>2</sup>^a--z++([0-9]++)\$` is invalid.

Your intention here appears to be to match any string which starts with one or more lowercase letters and ends with one or more digits. However, your character class range `a--z` seems to be incorrectly typed. In regular expressions, a single `--` is used to represent a range, so this should be `a-z` instead:

So the corrected regular expression should look like this:  
```cpp  
"^[a-z]++([0-9]++)\$"  
```

The corrected code should then be:

```
```cpp
1 #include <ctre.hpp>
2 #include <optional>
3
4 std::optional<std::string_view> extract_number(std::string_view s) noexcept {
5     using namespace ctre::literals;
6     if (auto m = ctre::match<"^a-z++([0-9]++)$>(s)) {
7         return m.get<1>().to_view();
8     } else {
9         return std::nullopt;
10    }
11 }
```

When your original code tried to compile, the compiler tried to compile an instantiation of `problem\_at\_position<4>`, in other words it was trying to tell you there was a problem at the 4th character in your regular expression, which is where your `--` is located. Once you fix your regular expression to `"<sup>2</sup>^a-z++([0-9]++)\$` the code should compile without issues.

This kind of messages can help you debug your regular expressions in the future by helping you locate problematic parts of your regular expression. So when you see `problem\_at\_position<n>`, you can look at the nth character in your regular expression for potential issues.

(Total cost: approximately \$0.06 USD.)

=====  
e/main/include/ctre/literals.hpp:7,  
e/main/include/ctre.hpp:4,  
  
upper.hpp: In instantiation of 'struct ctre::problem\_at\_position<n>':  
upper.hpp:306:100: required from 'const ctre::problem\_at\_position<n> &operator<<(...)'  
  
upper.hpp:299:42: error: invalid use of incomplete type 'ctre::problem\_at\_position<n>'; did you mean 'problem\_at\_position<n>{}'? "Regular Expression Problem At Position n"  
~~~~~  
upper.hpp:277:26: note: declaration of 'std::problem_at_position<n>'
// do not define!

g_view<char> > extract_number(std::string_view s) noexcept {
 auto m = ctre::regex_results<const char*>::get<1>(s);
 if (m) {
 return m.get<1>().to_view();
 } else {
 return std::nullopt;
 }
1 in template parameter list for 'template<class T> struct ctre::problem_at_position<T>'

e/main/include/ctre/evaluation.hpp:9,
e/main/include/ctre/literals.hpp:6:
turn_type.hpp:305:136: note: candidate: 'std::enable_if_t<decltype(_captures)>::template enable_if_t<decltype(_captures)> turn_type<T>::turn_type<T>() const' [with T = ctre::problem_at_position<n>]
turn_type.hpp:305:136: note: template argument deduction/substitution failed: 'std::enable_if_t<decltype(_captures)> turn_type<T>::turn_type<T>()' does not match 'const ctre::problem_at_position<n> &operator<<(...)' [with T = ctre::problem_at_position<n>]

k-20231002/include/c++/14.0.0/bits/stl_parallel.h:10: note: candidate: 'std::enable_if_t<decltype(_captures)> turn_type<T>::turn_type<T>() const' [with T = ctre::problem_at_position<n>]
k-20231002/include/c++/14.0.0/utility:69, note: candidate: 'std::enable_if_t<decltype(_captures)> turn_type<T>::turn_type<T>() const' [with T = ctre::problem_at_position<n>]
e/main/include/ctll/fixed_string.hpp:4, note: candidate: 'std::enable_if_t<decltype(_captures)> turn_type<T>::turn_type<T>() const' [with T = ctre::problem_at_position<n>]
e/main/include/ctll/parser.hpp:4, note: candidate: 'std::enable_if_t<decltype(_captures)> turn_type<T>::turn_type<T>() const' [with T = ctre::problem_at_position<n>]
e/main/include/ctll.hpp:4, note: candidate: 'std::enable_if_t<decltype(_captures)> turn_type<T>::turn_type<T>() const' [with T = ctre::problem_at_position<n>]
e/main/include/ctre/literals.hpp:4: note: candidate: 'std::enable_if_t<decltype(_captures)> turn_type<T>::turn_type<T>() const' [with T = ctre::problem_at_position<n>]
/14.0.0/type_traits: In substitution of 'std::enable_if_t<decltype(_captures)> turn_type<T>::turn_type<T>() const' [with T = ctre::problem_at_position<n>]
turn_type.hpp:305:136: note: template argument deduction/substitution failed: 'std::enable_if_t<decltype(_captures)> turn_type<T>::turn_type<T>()' does not match 'const ctre::problem_at_position<n> &operator<<(...)' [with T = ctre::problem_at_position<n>]

turn_type.hpp:308:142: note: candidate: 'std::enable_if_t<decltype(_captures)> turn_type<T>::turn_type<T>() const' [with T = ctre::problem_at_position<n>]
turn_type.hpp:308:142: note: candidate: 'std::enable_if_t<decltype(_captures)> turn_type<T>::turn_type<T>() const' [with T = ctre::problem_at_position<n>]
turn_type.hpp:308:142: note: candidate: 'std::enable_if_t<decltype(_captures)> turn_type<T>::turn_type<T>() const' [with T = ctre::problem_at_position<n>]
1 in template parameter list for 'template<class T> struct ctre::problem_at_position<T>'

turn_type.hpp:312:152: note: candidate: 'std::enable_if_t<decltype(_captures)> turn_type<T>::turn_type<T>() const' [with T = ctre::problem_at_position<n>]
turn_type.hpp:312:152: note: candidate: 'std::enable_if_t<decltype(_captures)> turn_type<T>::turn_type<T>() const' [with T = ctre::problem_at_position<n>]
turn_type.hpp:312:152: note: candidate: 'std::enable_if_t<decltype(_captures)> turn_type<T>::turn_type<T>() const' [with T = ctre::problem_at_position<n>]



```

1 #include <ctre.hpp>
2 #include <optional>
3
4 std::optional<std::string_view> extract_number(std::string_view s) {
5     using namespace ctre::literals;
6     if (auto m = ctre::match<"^a--z++([0-9]++)$">(s)) {
7         return m.get<1>().to_view();
8     } else {
9         return std::nullopt;
10    }
11 }
12

```



<https://gcc.godbolt.org/z/a5Kavxjab>

C++

CWhy

Your regular expression `"[a--z]++([0-9]++)\$"` is invalid.

Your intention here appears to be to match any string which starts with one or more lowercase letters and ends with one or more digits. However, your character class range `a--z` seems to be incorrectly typed. In regular expressions, a single ``-`` is used to represent a range, so this should be `a-z` instead:

So the corrected regular expression should look like this:

```

```cpp
"^[a-z]++([0-9]++)$"
```

```

The corrected code should then be:

```

```cpp
1 #include <ctre.hpp>
2 #include <optional>
3
4 std::optional<std::string_view> extract_number(std::string_view s) noexcept {
5 using namespace ctre::literals;
6 if (auto m = ctre::match<"^a-z++([0-9]++)$">(s)) {
7 return m.get<1>().to_view();
8 } else {
9 return std::nullopt;
10 }
```

```

When your original code tried to compile, the compiler tried to compile an instantiation of `problem_at_position<4>`, in other words it was trying to tell you there was a problem at the 4th character in your regular expression, which is where your `--` is located. Once you fix your regular expression to `"[a-z]++([0-9]++)\$"` the code should compile without issues.

This kind of messages can help you debug your regular expressions in the future by helping you locate problematic parts of your regular expression. So when you see `problem_at_position<n>`, you can look at the nth character in your regular expression for potential issues.

(Total cost: approximately \$0.06 USD.)

```

e/main/include/ctre/literals.hpp:7,
e/main/include/ctre.hpp:4,
apper.hpp: In instantiation of 'struct ctre::problem_at_position<n>':
apper.hpp:306:100:   required from 'const ctre::problem_at_position<n> &problem_at_position<n>::operator=(const ctre::problem_at_position<n> &)' [with n = 4]
apper.hpp:299:42: error: invalid use of incomplete type 'class ctre::problem_at_position<n>' [-fpermissive]
apper.hpp:277:26: note: declaration of 'class ctre::problem_at_position<n>' was never completed
g_view<char> > extract_number(std::string_view s) noexcept {
o 'ctre::regex_results<const char*>::get<4>' [with
e/main/include/ctre/evaluation.hpp:9,
e/main/include/ctre/literals.hpp:6:
turn_type.hpp:305:136: note: candidate: 'std::enable_if_t<decltype(_captures)>::template turn_type<std::enable_if_t<decltype(_captures)>::value_type>::turn_type'
turn_type.hpp:305:136: note: template argument deduction/substitution failed:
k-20231002/include/c++/14.0.0/bits/stl_parallel.h: In instantiation of 'void std::transform<std::function<T(T, U)>, std::vector<T>, std::vector<U>, std::function<T(U)>>::operator()<T, U>(<brace-enclosed initializer list>)' [with T = int, U = int]:
k-20231002/include/c++/14.0.0/utility:69, at top level:
e/main/include/ctll/fixed_string.hpp:4,
e/main/include/ctll/parser.hpp:4,
e/main/include/ctll.hpp:4,
e/main/include/ctre/literals.hpp:4:
/14.0.0/type_traits: In substitution of 'std::enable_if_t<decltype(_captures)>::template turn_type<std::enable_if_t<decltype(_captures)>::value_type>::turn_type'
turn_type.hpp:305:136: note: template argument deduction/substitution failed:
/14.0.0/type_traits:2623:11: error: no type named '_Tp' [with _Tp = std::enable_if_t<decltype(_captures)>::value_type]
turn_type.hpp:308:142: note: candidate: 'std::enable_if_t<decltype(_captures)>::template turn_type<std::enable_if_t<decltype(_captures)>::value_type>::turn_type'
turn_type.hpp:308:142: note: template argument deduction/substitution failed:
1 in template parameter list for 'template<class _Tp> void std::transform<std::function<_Tp(_Tp, _Tp)>, std::vector<_Tp>, std::vector<_Tp>, std::function<_Tp(_Tp)>>::operator()<_Tp>(<brace-enclosed initializer list>)' [with _Tp = int]
turn_type.hpp:312:152: note: candidate: 'std::enable_if_t<decltype(_captures)>::template turn_type<std::enable_if_t<decltype(_captures)>::value_type>::turn_type'
turn_type.hpp:312:152: note: template argument deduction/substitution failed:

```

C++ source #1 X

A - + v 🔍 🌐

C++ CWhy

```
1 #include <ctre.hpp>
2 #include <optional>
3
4 std::optional<std::string_view> extract_number(std::string_view s) {
5     using namespace ctre::literals;
6     if (auto m = ctre::match^a--z++([0-9]++)$(s)) {
7         return m.get<1>().to_view();
8     } else {
9         return std::nullopt;
10    }
11 }
```

Your regular expression `"^{^a--z++([0-9]++)\$}"` is invalid.

Your intention here appears to be to match any string which starts with one or more lowercase letters and ends with one or more digits. However, your character class range `a--z` seems to be incorrectly typed. In regular expressions, a single `--` is used to represent a range, so this should be `a-z` instead:

So the corrected regular expression should look like this:
`^{^a-z++([0-9]++)\$}`

e/main/include/ctre/literals.hpp:7,
e/main/include/ctre.hpp:4,

upper.hpp: In instantiation of 'struct ctre::problem_at_position<n>':
upper.hpp:306:100: required from 'const std::optional<std::string_view> ctre::extract_number(std::string_view, _results<const char*>::get<n>)'
upper.hpp:299:42: error: invalid use of incomplete type 'ctre::problem_at_position<n>'; did you mean 'problem_at_position'?

upper.hpp:277:26: note: declaration of 'std::optional<std::string_view> ctre::extract_number(std::string_view, _results<const char*>::get<n>)'
// do not define!

/ctre/evaluation.hpp:9,
/ctre/literals.hpp:6:
305:136: note: candidate: 'ctre::problem_at_position<n>'
cltype(_captures)::template problem_at_position<n>

305:136: note: template ali
lude/c++/14.0.0/bits/stl_pa
lude/c++/14.0.0/utility:69,
/ctll/fixed_string.hpp:4,
/ctll/parser.hpp:4,
/ctll.hpp:4,
/ctre/literals.hpp:4:
raits: In substitution of 't
urn_type.hpp:305:23: required from here
/14.0.0/type_traits:2623:11: error: no typ
ond, _Tp>::type;

turn_type.hpp:308:142: note: candidate: 'std::enable_if_t<decltype(_captures)>::temp
turn_type.hpp:308:142: note: template ali
1 in template parameter list for 'tempat
turn_type.hpp:312:152: note: candidate: 't
pename = std::enable_if_t<decltype(_captures)>::tempat'

Alon Wolf to me ▾

Wed, Oct 4, 8:13PM (11 hours ago)

Hi,
Yes this is correct.
It provided an amazing answer, way better than I expected.

When your original code tried to compile, the compiler tried to compile an instantiation of `problem_at_position<4>`, in other words it was trying to tell you there was a problem at the 4th character in your regular expression, which is where your `--` is located. Once you fix your regular expression to `"^{^a-z++([0-9]++)\$}"` the code should compile without issues.

This kind of messages can help you debug your regular expressions in the future by helping you locate problematic parts of your regular expression. So when you see `problem_at_position<n>`, you can look at the nth character in your regular expression for potential issues.

(Total cost: approximately \$0.06 USD.)



<https://gcc.godbolt.org/z/a5Kavxjab>

C++ source #1 X

A - + v 🔍 🌐

C++ CWhy

```
1 #include <ctre.hpp>
2 #include <optional>
3
4 std::optional<std::string_view> extract_number(std::string_view s) {
5     using namespace ctre::literals;
6     if (auto m = ctre::match<"^a--z]+([0-9]++)$">(s)) {
7         return m.get<1>().to_view();
8     } else {
9         return std::nullopt;
10    }
11 }
```

Alon Wolf



to me ▾

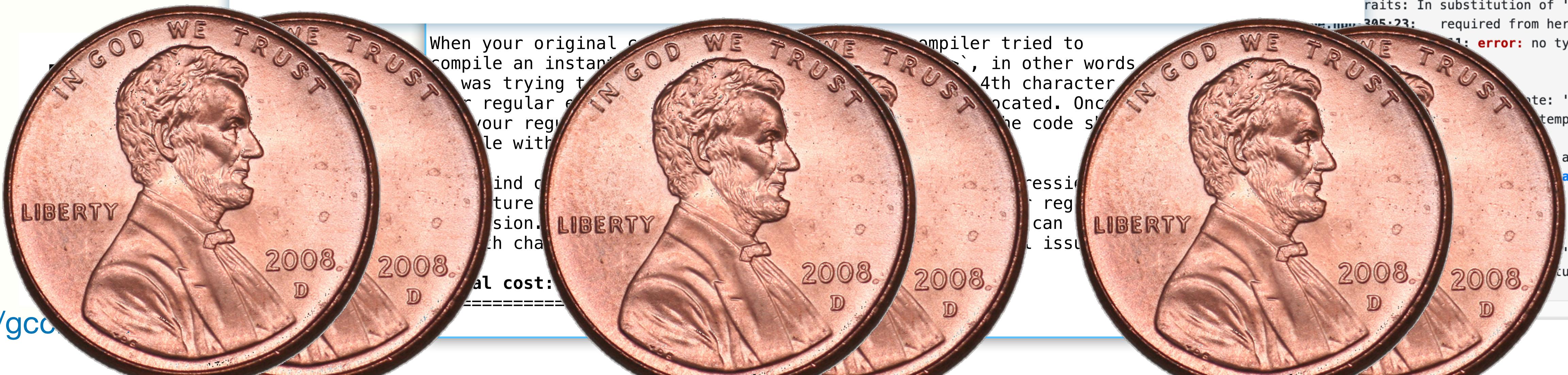
Wed, Oct 4, 8:13PM (11 hours ago)



Hi,

Yes this is correct.

It provided an amazing answer, way better than I expected.



<https://gcc.io/>



```
=====
CWhy
=====
===== Prompt =====
This is my code:
File `ctre-test.cpp`:
```
1 #include <ctre.hpp>
2 #include <optional>
3
4 std::optional<std::string_view> extract_number(std::string_view s) noexcept {
5 using namespace ctre::literals;
6 if (auto m = ctre::match<"^a--z++([0-9]++)$">(s)) {
7 return m.get<1>().to_view();
8 } else {
9 return std::nullopt;
10 }
```

This is my error:
```
ctre-test.cpp:6:28: error: reference to string literal is not allowed in a template argument
 if (auto m = ctre::match<"^a--z++([0-9]++)$">(s)) {
 ^
ctre-test.cpp:7:14: error: missing 'template' keyword prior to dependent template name 'get'
[...]
 return m.get<1>().to_view();
 ^ ~~~
2 errors generated.
```
=====
```

What's the problem?

evolve



evolve

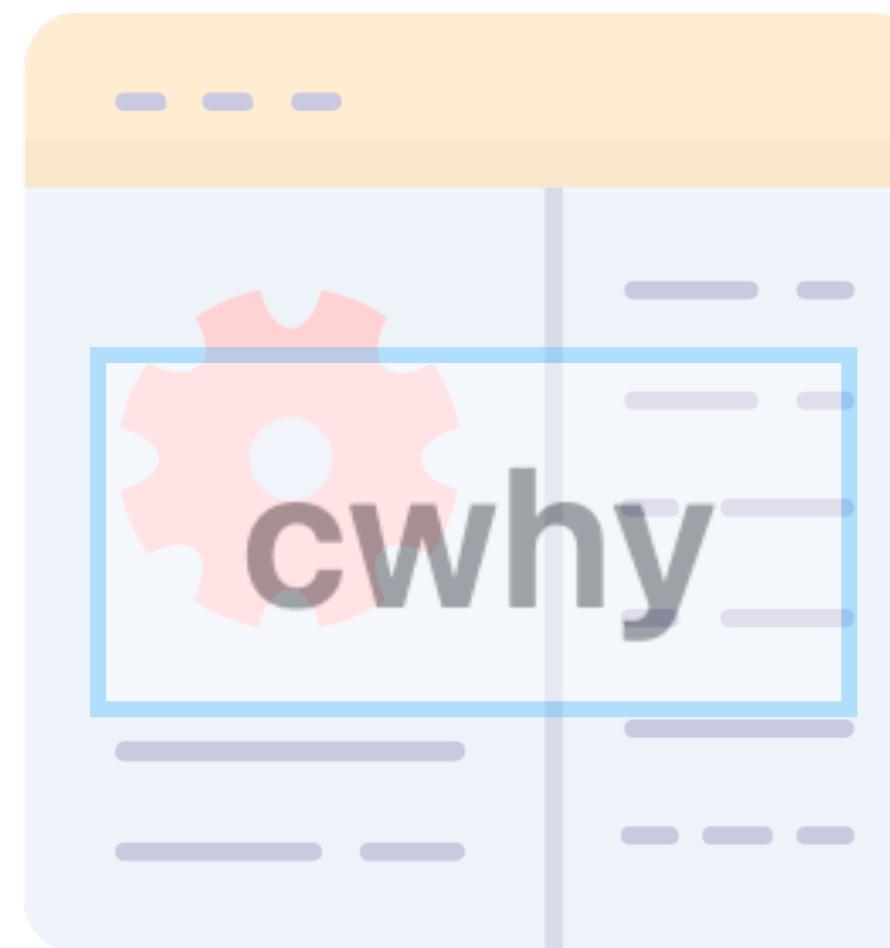
compiler that provides descriptive
error messages & proposes fixes



evolve

compiler that provides descriptive
error messages & proposes fixes

exploit niche



evolve

compiler that provides descriptive
error messages & proposes fixes

exploit niche



has access to all code & system incl.
files, compiler command line,
original error messages

evolve

compiler that provides descriptive
error messages & proposes fixes

exploit
niche



ensure
fitness

has access to all code & system incl.
files, compiler command line,
original error messages

evolve

compiler that provides descriptive
error messages & proposes fixes

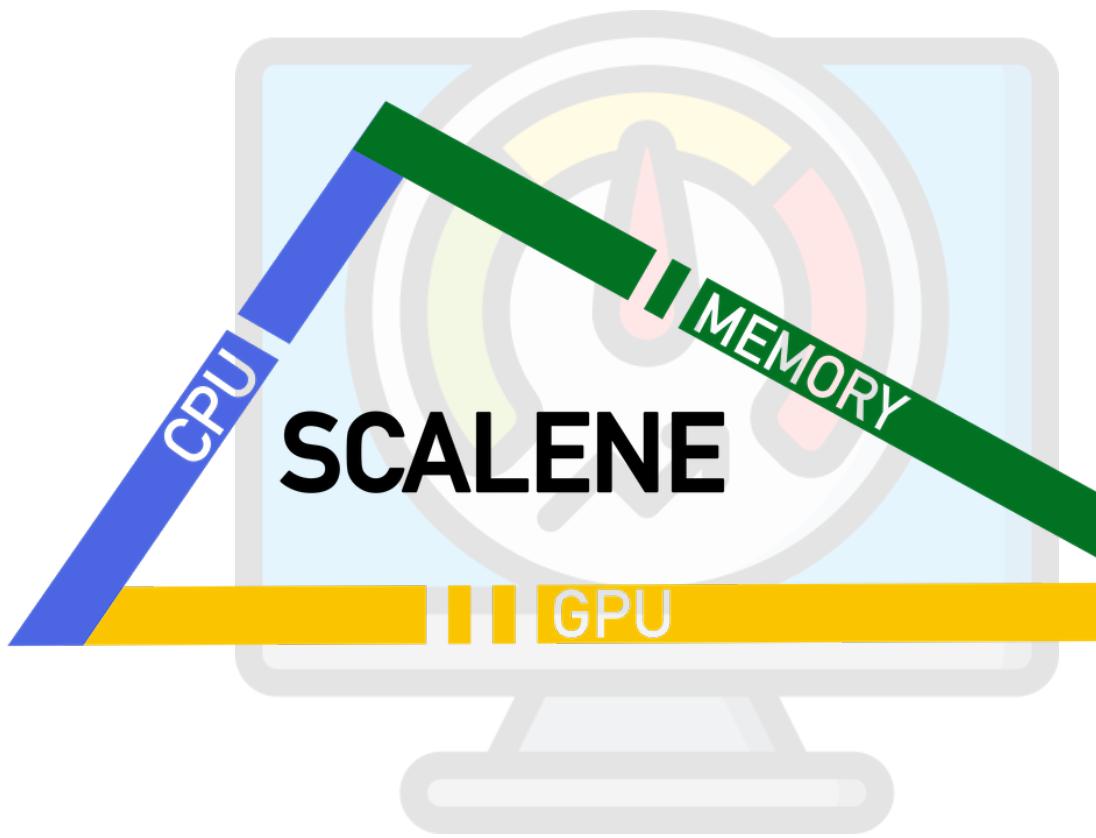
exploit
niche



has access to all code & system incl.
files, compiler command line,
original error messages

ensure
fitness

re-compile & verify that fix prevents
compiler error (TBD)



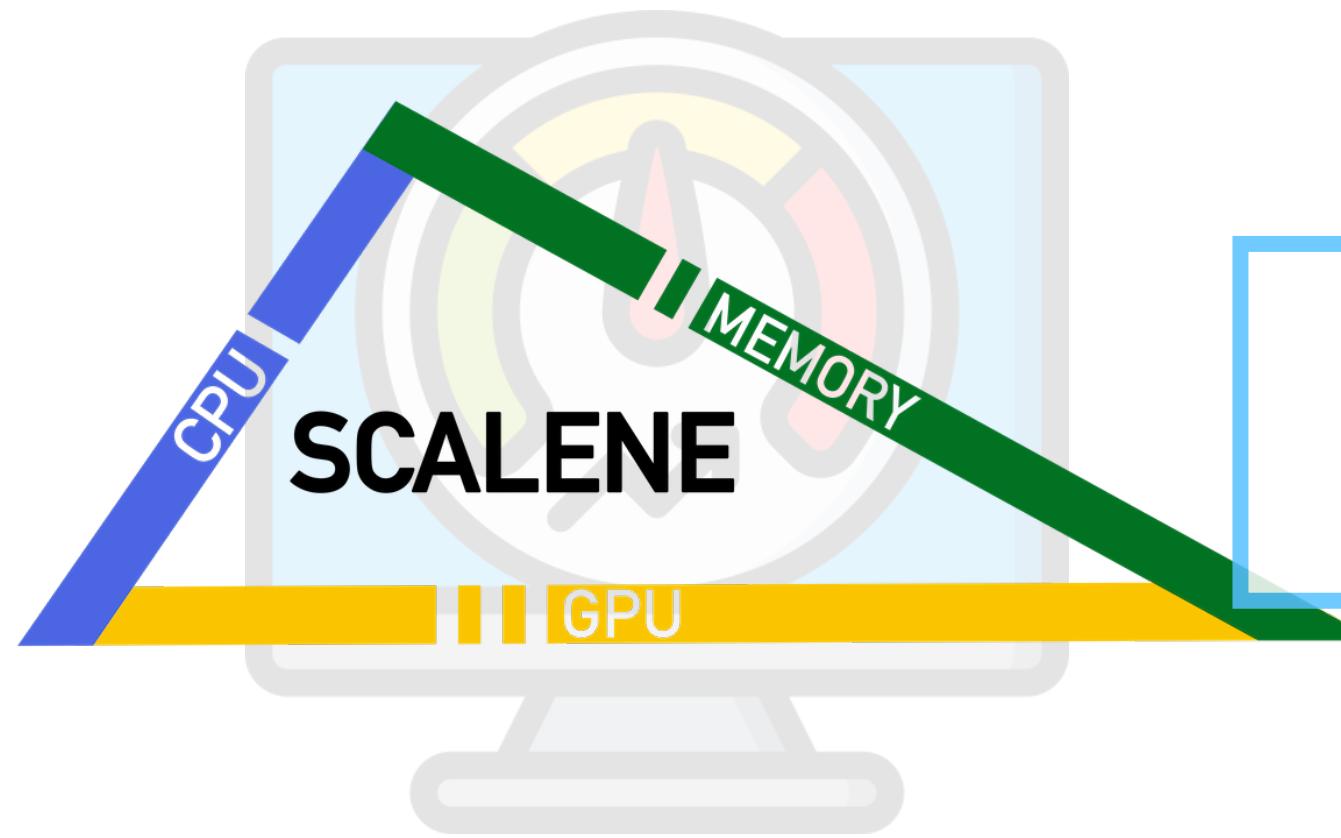
profiler



debugger



compiler



profiler



debugger



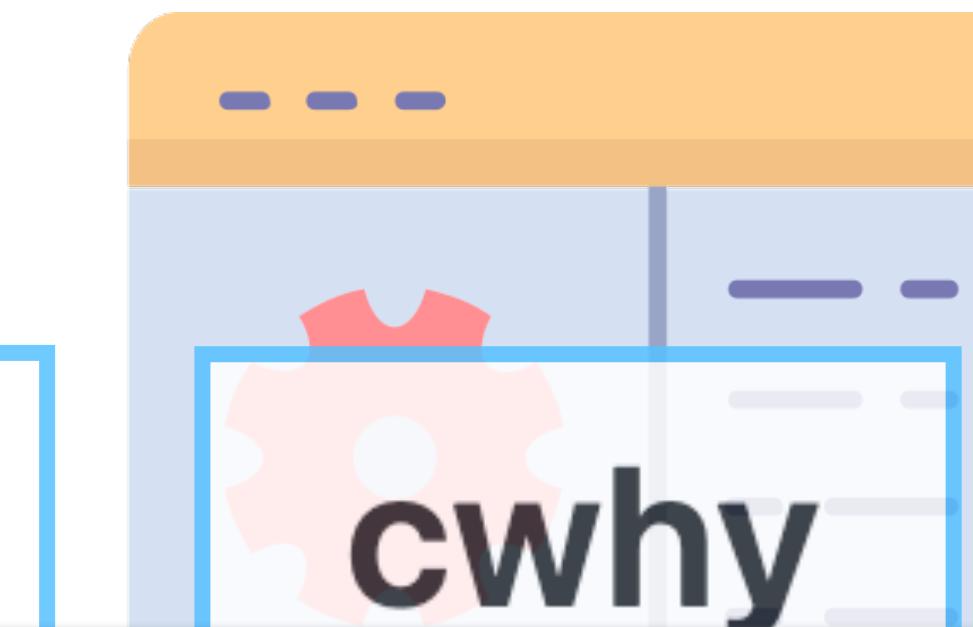
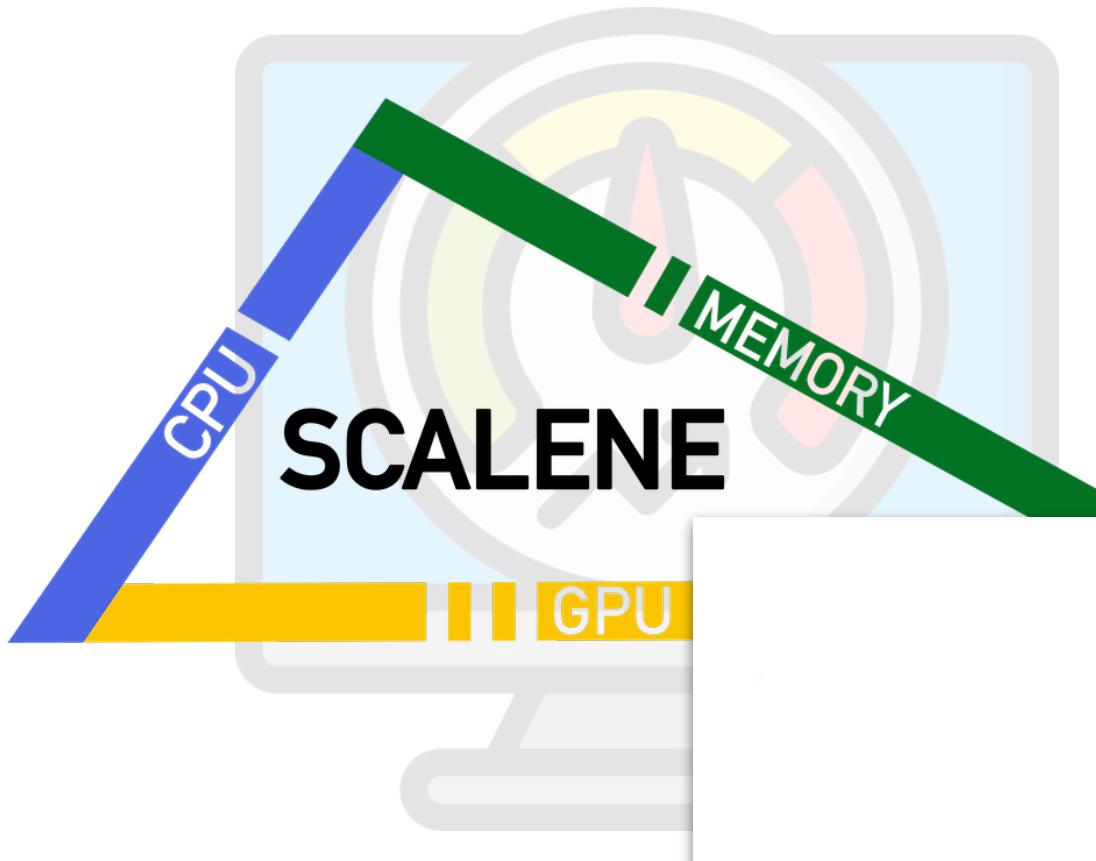
compiler

evolve

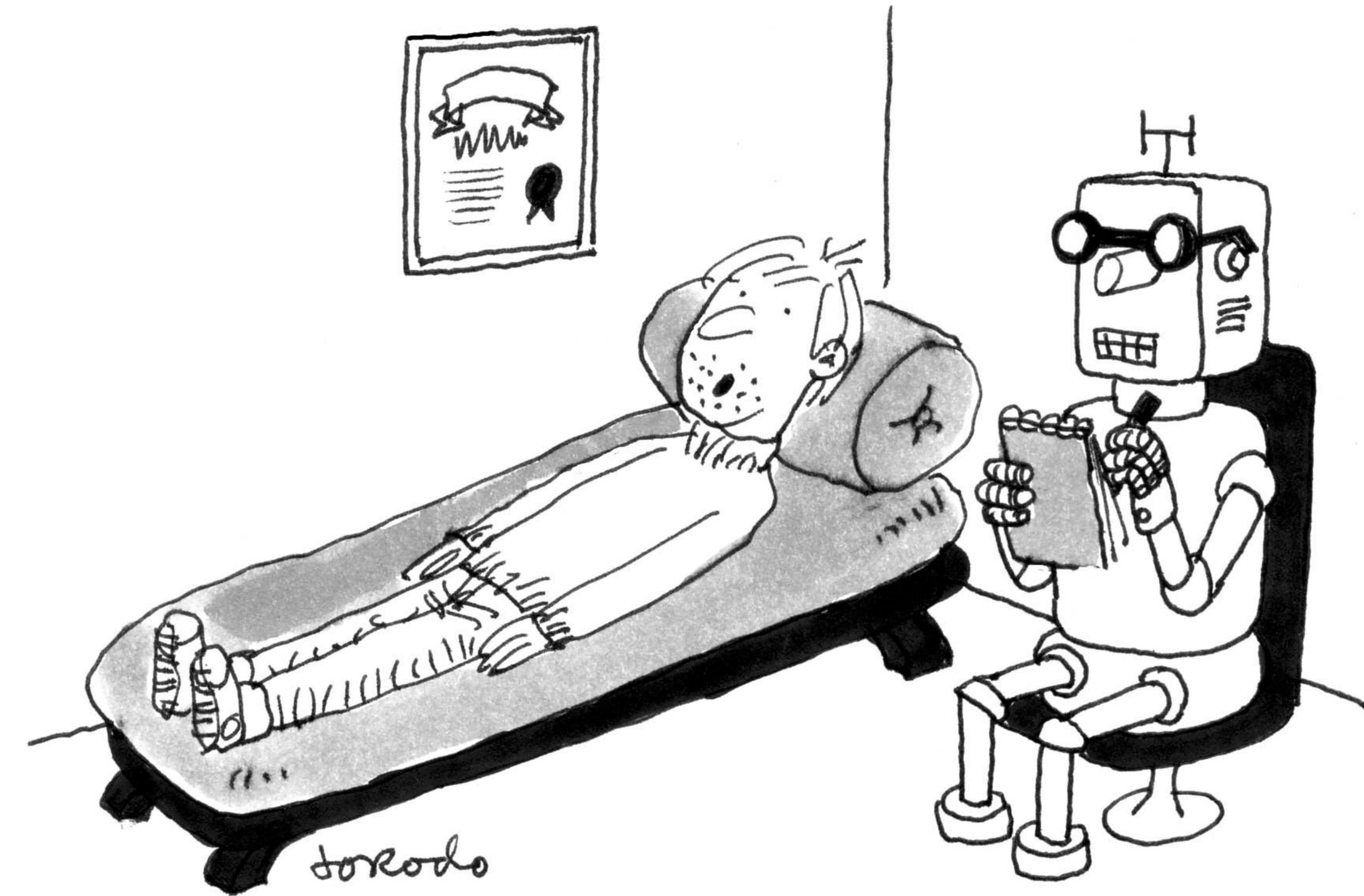
exploit
niche

ensure
fitness

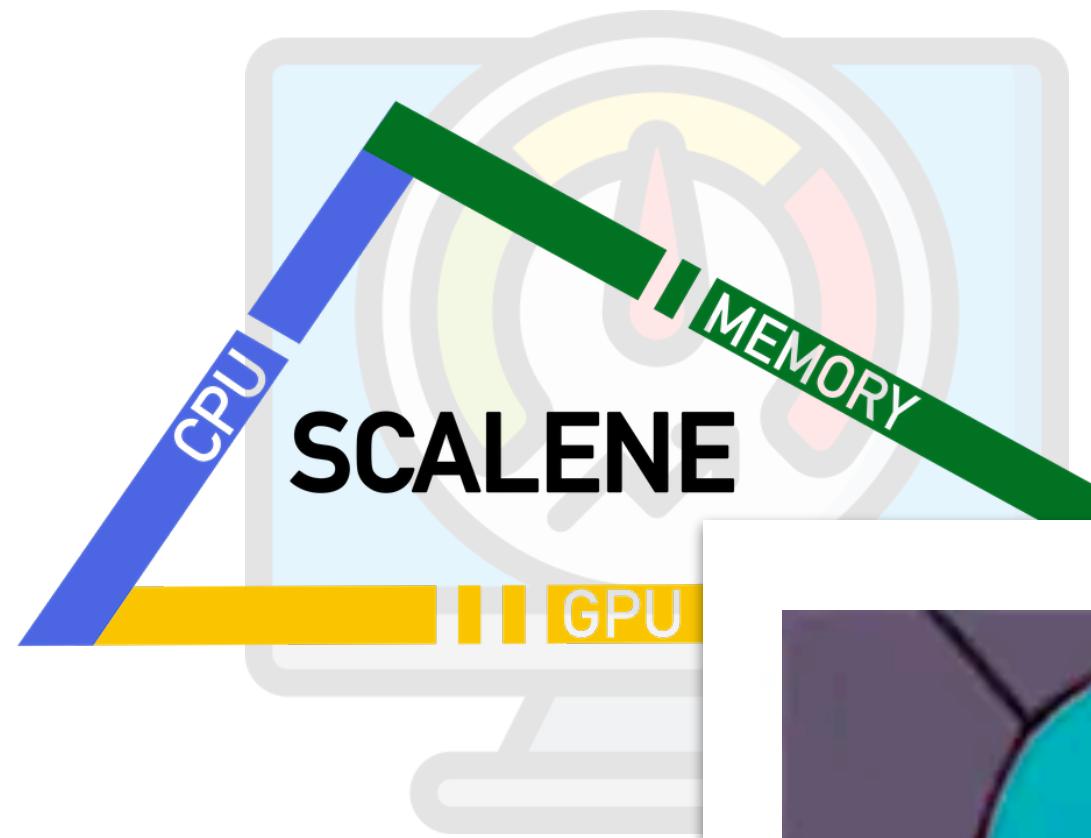




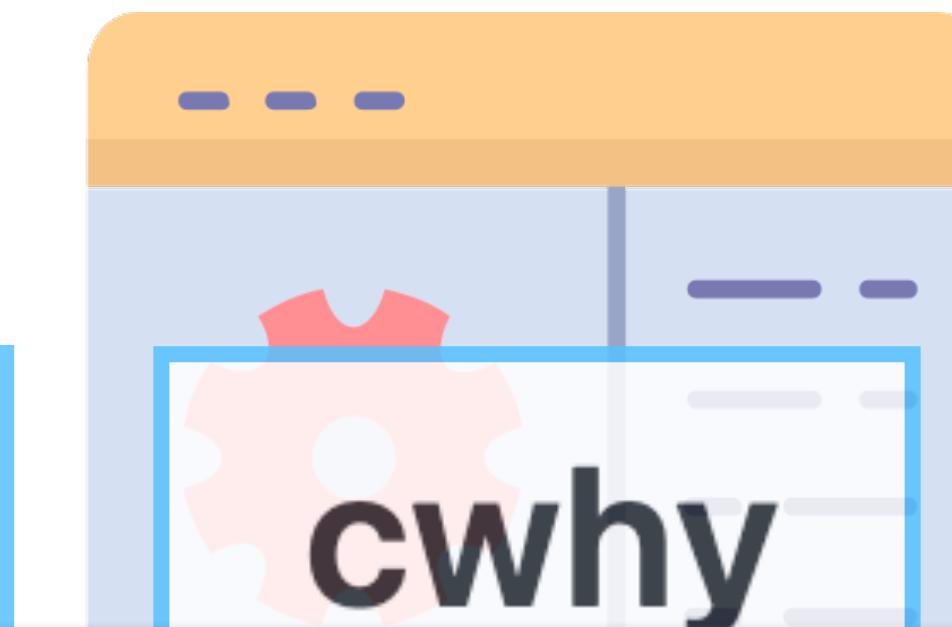
prof

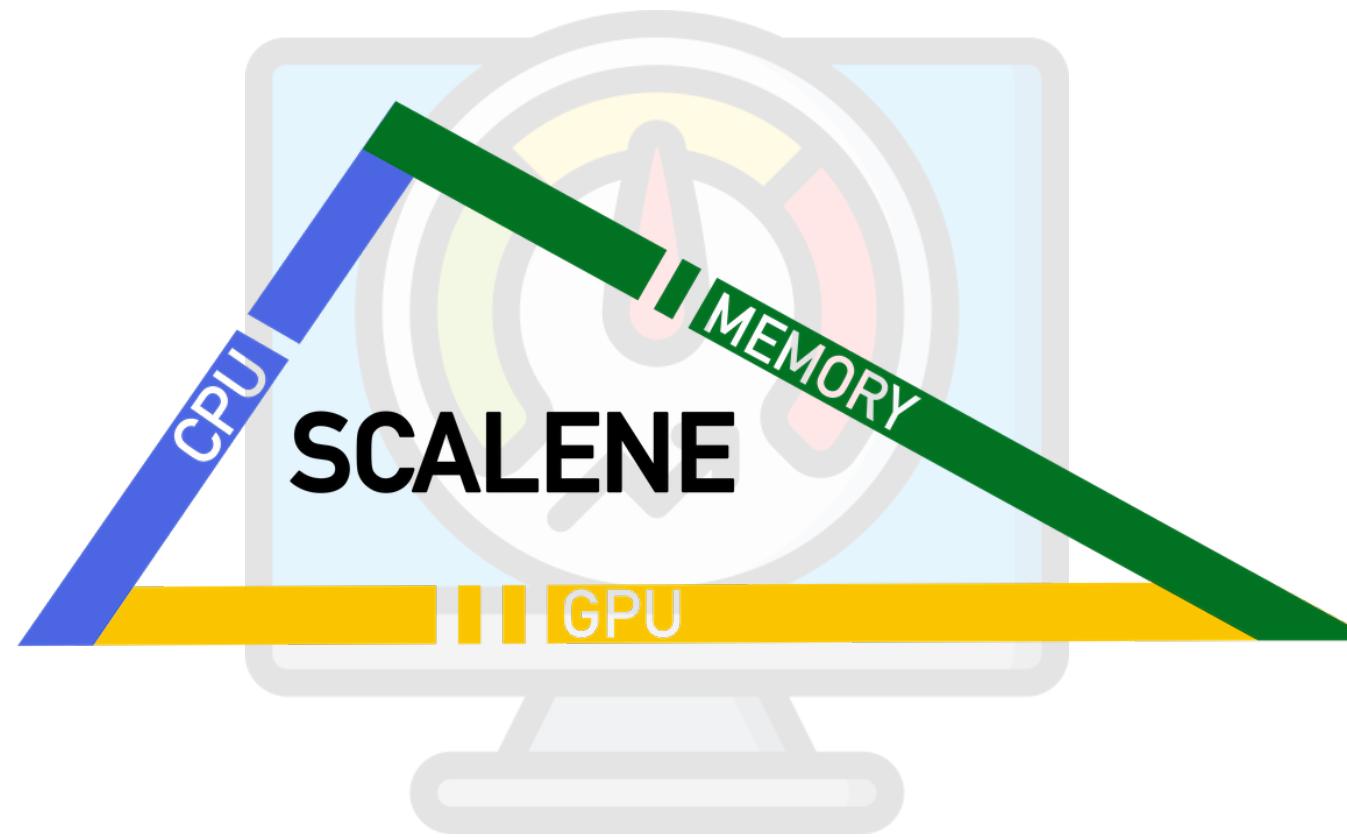


"I WORRY WHAT THE NEXT MASS EXTINCTION
MIGHT BE..."



prof





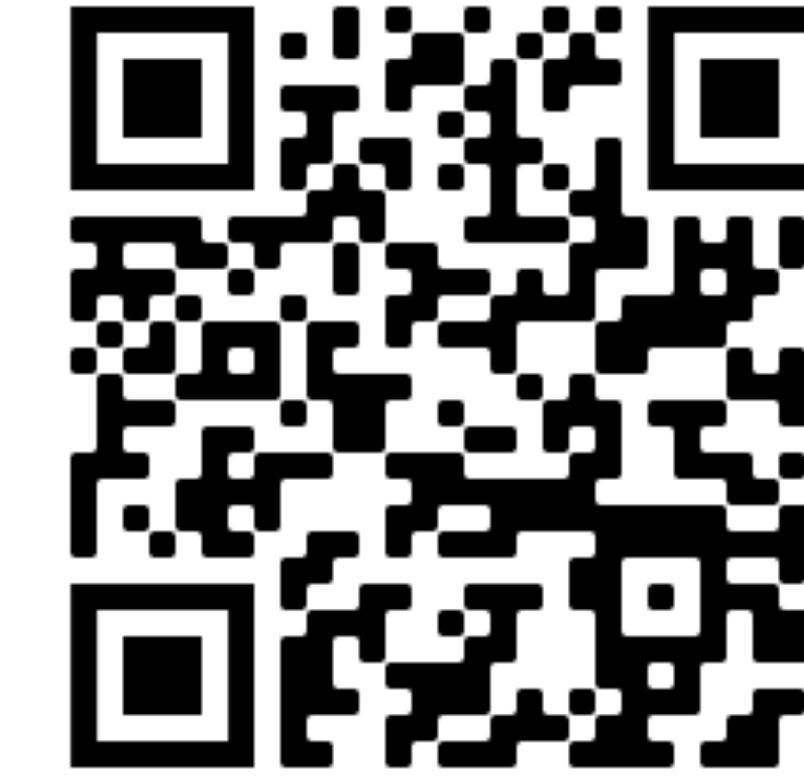
profiler

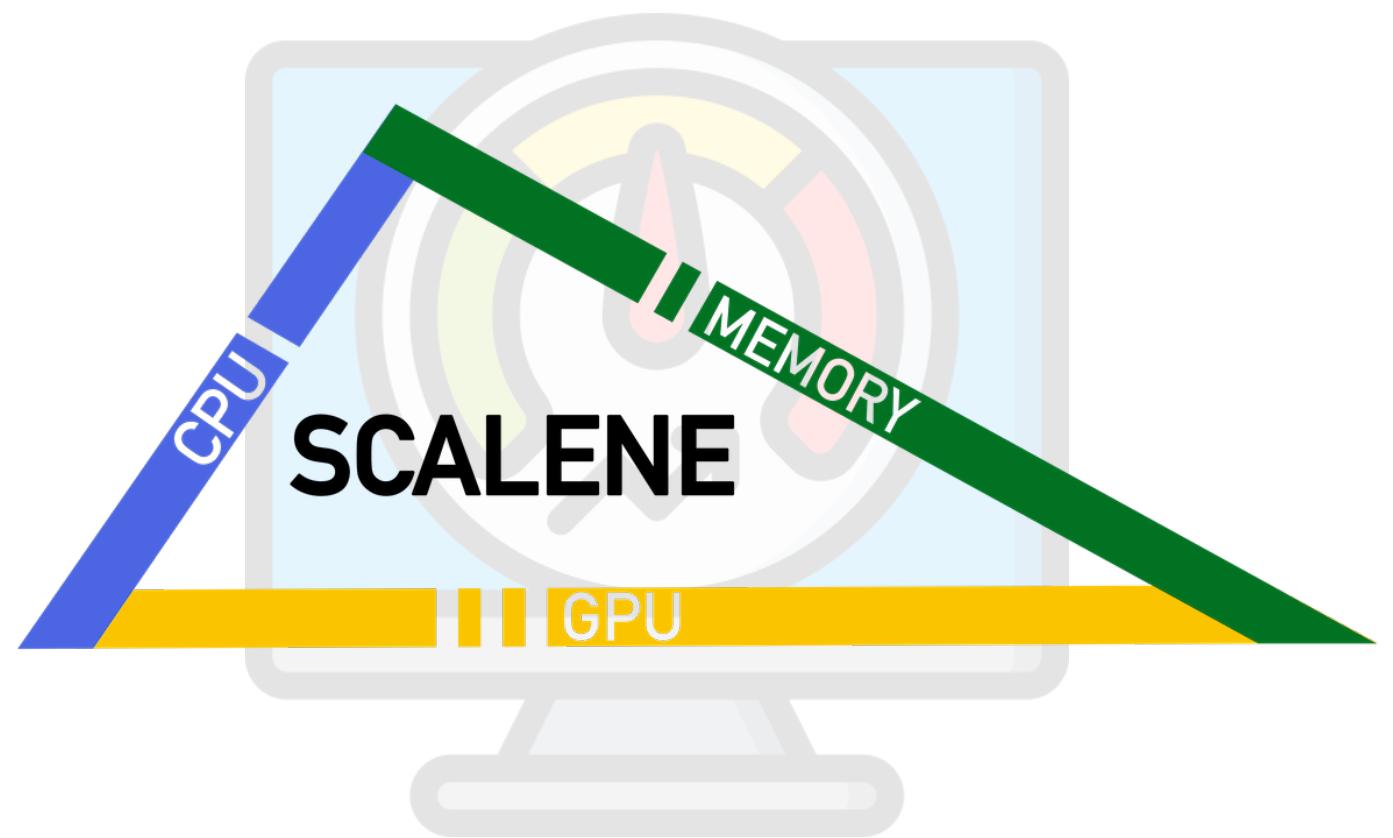


debugger

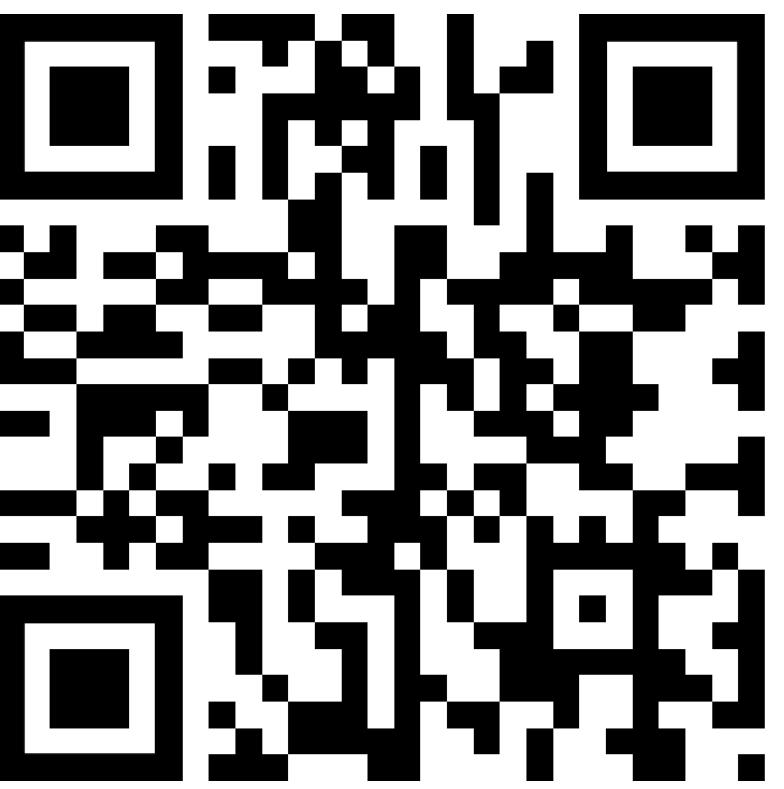


compiler





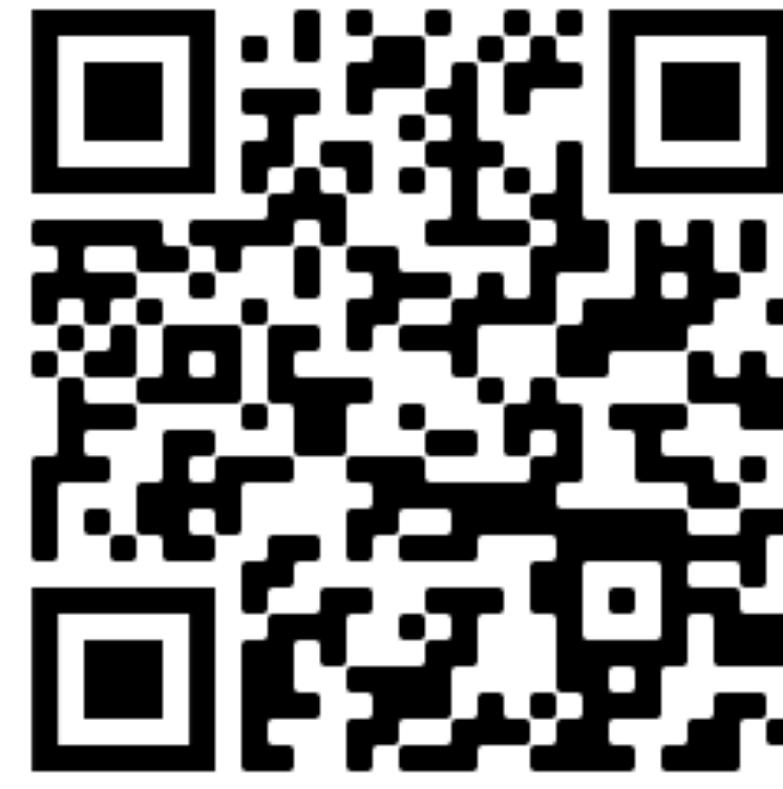
profiler

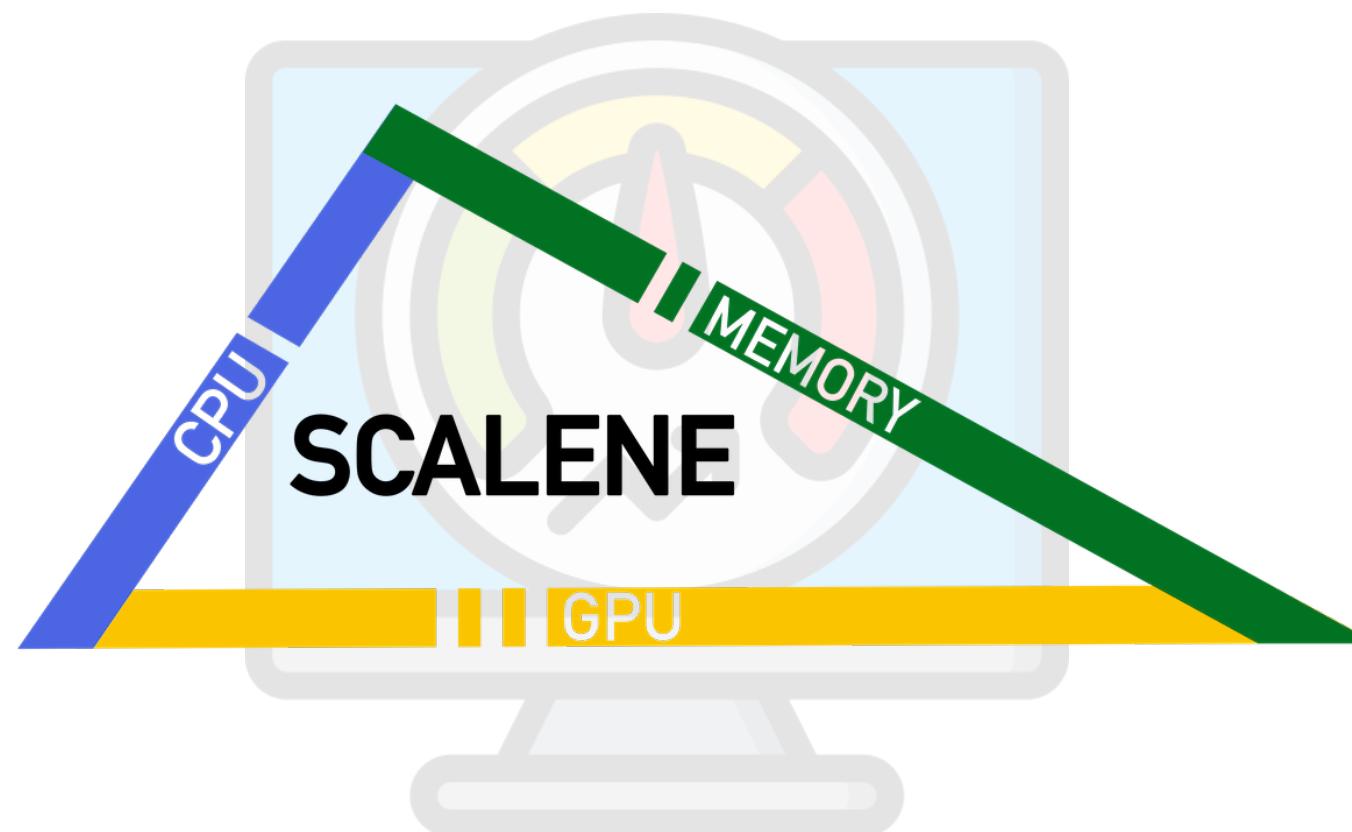


debugger



compiler





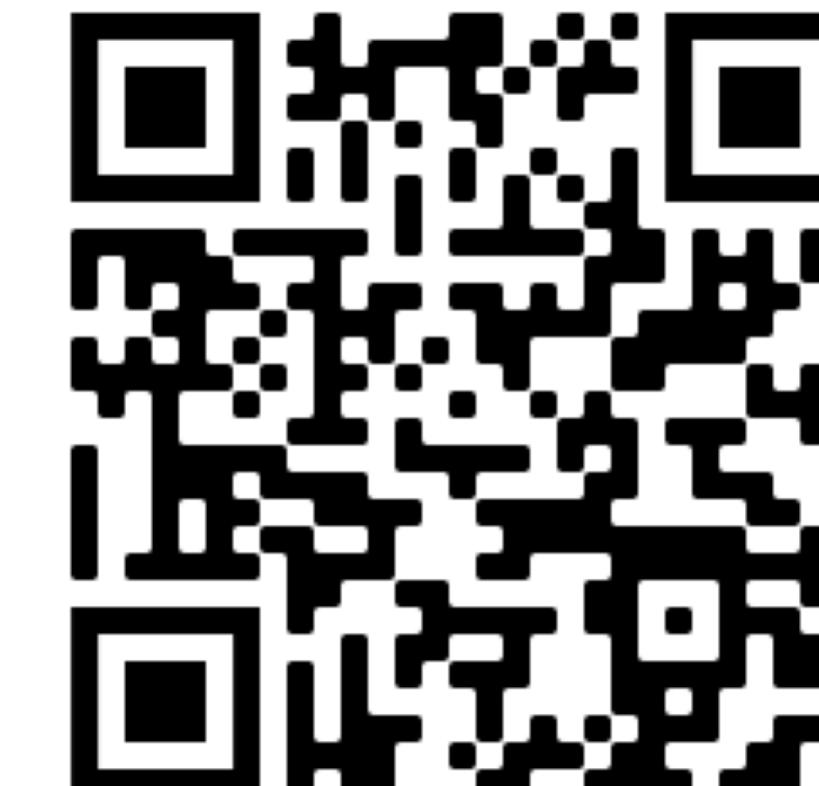
profiler

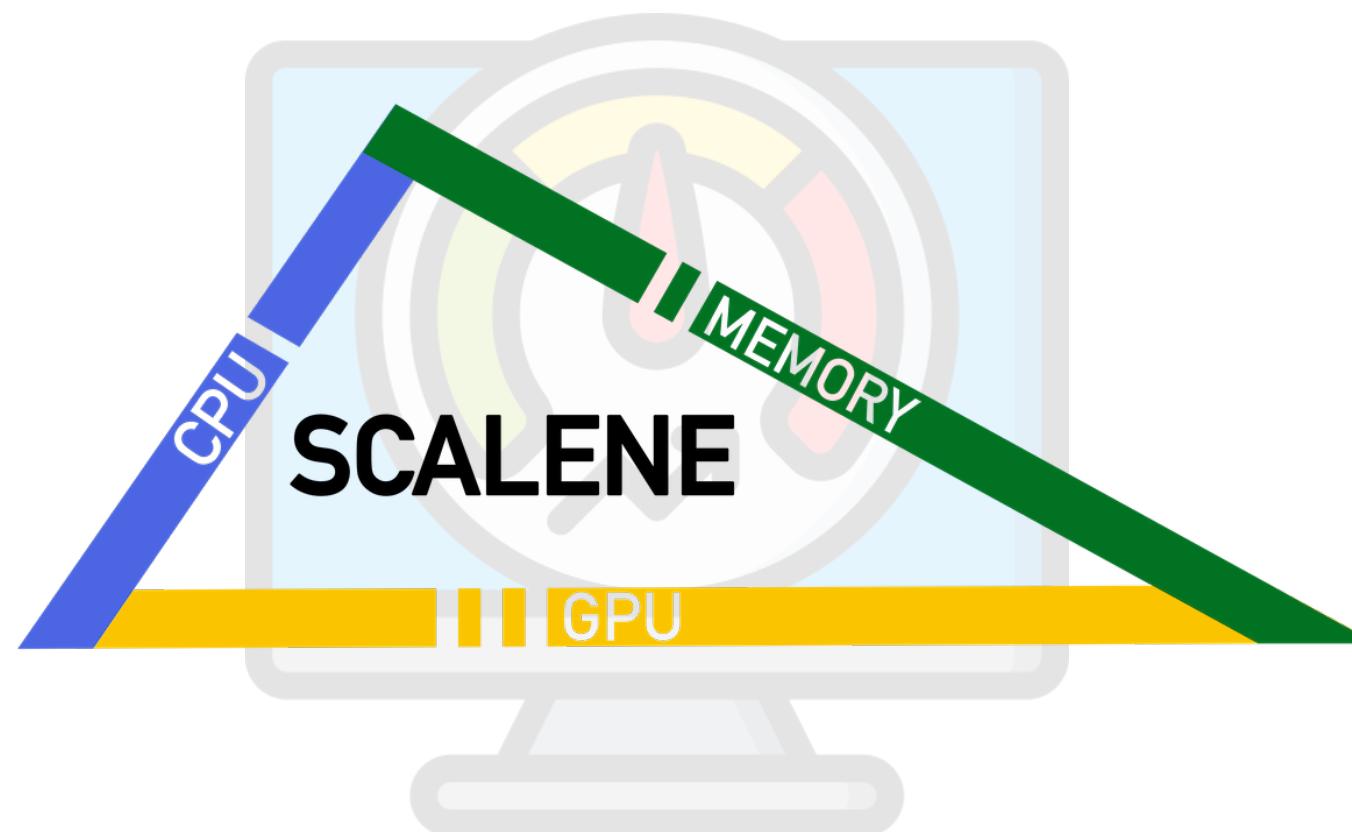


debugger



compiler





profiler



debugger



compiler

