Making CPython 3.11 Fast

Inside Python's new specializing, adaptive interpreter.

Brandt Bucher (April 21, 2022)

Making CPython 3.11 Especially Fast

Inside Python's new specializing, adaptive interpreter.

Brandt Bucher (April 21, 2022)

Brandt Bucher

Brandt Bucher

- 2017: Started using Python.
- 2018: Contributed code to CPython.
- 2019: Joined Python's Triage Team.
- 2020: Joined Python's Core Development Team.
- 2021: Joined Microsoft's CPython Performance Engineering Team.
- 2022: Made CPython faster!

Bytecode

Bytecode

```
class Point:
    def __init__(self, x: float, y: float) -> None:
        self.x = x
        self.y = y

    def shifted(self, dx: float, dy: float) -> typing.Self:
        x = dx + self.x
        y = dy + self.y
        cls = type(self)
        return cls(x, y)
```

Bytecode

```
y = dy + self.y
cls = type(self)
```

Bytecode

```
y = dy + self.y
cls = type(self)
```

```
import dis
dis.dis(Point.shifted)
```

Bytecode

`dis.dis(Point.shifted)`

```
y = dy + self.y
cls = type(self)
```

Bytecode

`dis.dis(Point.shifted)`

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST    (dy)
LOAD_FAST    (self)
LOAD_ATTR    (y)
BINARY_OP    (+)
STORE_FAST   (y)
LOAD_GLOBAL  (type)
LOAD_FAST    (self)
PRECALL      (1)
CALL         (1)
STORE_FAST   (cls)
```

Bytecode

dis.dis(Point.shifted)

```
y = dy + self.y      LOAD_FAST    (dy)
cls = type(self)     LOAD_FAST    (self)
                     LOAD_ATTR    (y)
                     BINARY_OP    (+)
                     STORE_FAST   (y)
                     LOAD_GLOBAL  (type)
                     LOAD_FAST    (self)          stack
                     PRECALL      (1)
                     CALL         (1)
                     STORE_FAST   (cls)
```

Bytecode

dis.dis(Point.shifted)

```
y = dy + self.y          LOAD_FAST    (dy)
cls = type(self)         LOAD_FAST    (self)
                         LOAD_ATTR    (y)
                         BINARY_OP    (+)
                         STORE_FAST   (y)
                         LOAD_GLOBAL  (type)
                         LOAD_FAST    (self)
                         PRECALL      (1)
                         CALL         (1)
                         STORE_FAST   (cls)
```

stack

dy

Bytecode

dis.dis(Point.shifted)

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST    (dy)
LOAD_FAST    (self)
LOAD_ATTR    (y)
BINARY_OP    (+)
STORE_FAST   (y)
LOAD_GLOBAL  (type)
LOAD_FAST    (self)
PRECALL      (1)
CALL         (1)
STORE_FAST   (cls)
```

stack
self
dy

Bytecode

dis.dis(Point.shifted)

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST    (dy)
LOAD_FAST    (self)
LOAD_ATTR    (y)
BINARY_OP    (+)
STORE_FAST   (y)
LOAD_GLOBAL  (type)
LOAD_FAST    (self)
PRECALL      (1)
CALL         (1)
STORE_FAST   (cls)
```

stack

self.y

dy

Bytecode

dis.dis(Point.shifted)

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST    (dy)
LOAD_FAST    (self)
LOAD_ATTR    (y)
BINARY_OP    (+)
STORE_FAST   (y)
LOAD_GLOBAL  (type)
LOAD_FAST    (self)
PRECALL      (1)
CALL         (1)
STORE_FAST   (cls)
```

stack

dy + self.y

Bytecode

dis.dis(Point.shifted)

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST    (dy)
LOAD_FAST    (self)
LOAD_ATTR    (y)
BINARY_OP    (+)
STORE_FAST   (y)
LOAD_GLOBAL  (type)
LOAD_FAST    (self)
PRECALL      (1)
CALL         (1)
STORE_FAST   (cls)
```

```
y = dy + self.y
```

stack

Bytecode

dis.dis(Point.shifted)

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST    (dy)
LOAD_FAST    (self)
LOAD_ATTR    (y)
BINARY_OP    (+)
STORE_FAST   (y)
LOAD_GLOBAL  (type)
LOAD_FAST    (self)
PRECALL      (1)
CALL         (1)
STORE_FAST   (cls)
```

```
y = dy + self.y
```

stack

type

Bytecode

dis.dis(Point.shifted)

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST    (dy)
LOAD_FAST    (self)
LOAD_ATTR    (y)
BINARY_OP    (+)
STORE_FAST   (y)
LOAD_GLOBAL  (type)
LOAD_FAST    (self)
PRECALL      (1)
CALL         (1)
STORE_FAST   (cls)
```

```
y = dy + self.y
```

stack

self

type

Bytecode

dis.dis(Point.shifted)

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST    (dy)
LOAD_FAST    (self)
LOAD_ATTR    (y)
BINARY_OP    (+)
STORE_FAST   (y)
LOAD_GLOBAL  (type)
LOAD_FAST    (self)
PRECALL      (1)
CALL         (1)
STORE_FAST   (cls)
```

```
y = dy + self.y
```

stack

self

type

Bytecode

dis.dis(Point.shifted)

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST    (dy)
LOAD_FAST    (self)
LOAD_ATTR    (y)
BINARY_OP    (+)
STORE_FAST   (y)
LOAD_GLOBAL  (type)
LOAD_FAST    (self)
PRECALL      (1)
CALL         (1)
STORE_FAST   (cls)
```

```
y = dy + self.y
```

stack

type(self)

Bytecode

dis.dis(Point.shifted)

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST    (dy)
LOAD_FAST    (self)
LOAD_ATTR    (y)
BINARY_OP    (+)
STORE_FAST   (y)
LOAD_GLOBAL  (type)
LOAD_FAST    (self)
PRECALL      (1)
CALL         (1)
STORE_FAST   (cls)
```

```
 y = dy + self.y
cls = type(self)
```

stack

Bytecode

dis.dis(Point.shifted)

```
y = dy + self.y      LOAD_FAST    (dy)
cls = type(self)     LOAD_FAST    (self)
                     LOAD_ATTR    (y)
                     BINARY_OP    (+)
                     STORE_FAST   (y)
                     LOAD_GLOBAL  (type)
                     LOAD_FAST    (self)          stack
                     PRECALL      (1)
                     CALL         (1)
                     STORE_FAST   (cls)
```

Bytecode

dis.dis(Point.shifted)

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST                (dy)
LOAD_FAST                (self)
LOAD_ATTR                (y)
BINARY_OP                (+)
STORE_FAST               (y)
LOAD_GLOBAL              (type)
LOAD_FAST                (self)
PRECALL                  (1)
CALL                     (1)
STORE_FAST               (cls)
```

Quickening

dis.dis(Point.shifted)

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST                (dy)
LOAD_FAST                (self)
LOAD_ATTR                (y)
BINARY_OP                (+)
STORE_FAST               (y)
LOAD_GLOBAL              (type)
LOAD_FAST                (self)
PRECALL                  (1)
CALL                     (1)
STORE_FAST               (cls)
```

Quickening

dis.dis(Point.shifted, adaptive=True)

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST                (dy)
LOAD_FAST                (self)
LOAD_ATTR                (y)
BINARY_OP                (+)
STORE_FAST               (y)
LOAD_GLOBAL              (type)
LOAD_FAST                (self)
PRECALL                  (1)
CALL                     (1)
STORE_FAST               (cls)
```

Superinstructions

`dis.dis(Point.shifted, adaptive=True)`

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST                (dy)
LOAD_FAST                (self)
LOAD_ATTR                (y)
BINARY_OP                (+)
STORE_FAST               (y)
LOAD_GLOBAL              (type)
LOAD_FAST                (self)
PRECALL                  (1)
CALL                     (1)
STORE_FAST               (cls)
```

Superinstructions

`dis.dis(Point.shifted, adaptive=True)`

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST                (dy)
LOAD_FAST                (self)
LOAD_ATTR                (y)
BINARY_OP                (+)
STORE_FAST               (y)
LOAD_GLOBAL              (type)
LOAD_FAST                (self)
PRECALL                  (1)
CALL                     (1)
STORE_FAST               (cls)
```

Superinstructions

`dis.dis(Point.shifted, adaptive=True)`

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST__LOAD_FAST      (dy, self)
LOAD_ATTR                 (y)
BINARY_OP                 (+)
STORE_FAST                (y)
LOAD_GLOBAL               (type)
LOAD_FAST                 (self)
PRECALL                   (1)
CALL                      (1)
STORE_FAST                (cls)
```

Adaptive Instructions

`dis.dis(Point.shifted, adaptive=True)`

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST__LOAD_FAST      (dy, self)
LOAD_ATTR                 (y)
BINARY_OP                 (+)
STORE_FAST                (y)
LOAD_GLOBAL               (type)
LOAD_FAST                 (self)
PRECALL                   (1)
CALL                      (1)
STORE_FAST                (cls)
```

Adaptive Instructions

`dis.dis(Point.shifted, adaptive=True)`

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST__LOAD_FAST      (dy, self)
LOAD_ATTR_ADAPTIVE        (y)
BINARY_OP_ADAPTIVE        (+)
STORE_FAST                (y)
LOAD_GLOBAL_ADAPTIVE      (type)
LOAD_FAST                 (self)
PRECALL_ADAPTIVE          (1)
CALL_ADAPTIVE             (1)
STORE_FAST                (cls)
```

Specialized Instructions

`dis.dis(Point.shifted, adaptive=True)`

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST__LOAD_FAST      (dy, self)
LOAD_ATTR_ADAPTIVE        (y)
BINARY_OP_ADAPTIVE        (+)
STORE_FAST                (y)
LOAD_GLOBAL_ADAPTIVE      (type)
LOAD_FAST                 (self)
PRECALL_ADAPTIVE          (1)
CALL_ADAPTIVE             (1)
STORE_FAST                (cls)
```

Specialized Instructions

`dis.dis(Point.shifted, adaptive=True)`

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST__LOAD_FAST      (dy, self)
LOAD_ATTR_ADAPTIVE        (y)
BINARY_OP_ADAPTIVE        (+)
STORE_FAST                (y)
LOAD_GLOBAL_ADAPTIVE      (type)
LOAD_FAST                 (self)
PRECALL_ADAPTIVE          (1)
CALL_ADAPTIVE             (1)
STORE_FAST                (cls)
```

Specialized Instructions

`dis.dis(Point.shifted, adaptive=True)`

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST__LOAD_FAST      (dy, self)
LOAD_ATTR_INSTANCE_VALUE  (y)
BINARY_OP_ADAPTIVE        (+)
STORE_FAST                (y)
LOAD_GLOBAL_ADAPTIVE      (type)
LOAD_FAST                 (self)
PRECALL_ADAPTIVE          (1)
CALL_ADAPTIVE             (1)
STORE_FAST                (cls)
```

LOAD_ATTR_INSTANCE_VALUE

- Check if the class is the same as last time.
- Check if the object's `__dict__` keys are the same as last time.
- Reach directly into the `__dict__` values at the same offset as last time.
- Return the result.

LOAD_ATTR_INSTANCE_VALUE

- Check if the class is the same as last time.
- Check if the object's `__dict__` keys are the same as last time.
- Reach directly into the `__dict__` values at the same offset as last time.
- Return the result.

Specialized Instructions

`dis.dis(Point.shifted, adaptive=True)`

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST__LOAD_FAST      (dy, self)
LOAD_ATTR_INSTANCE_VALUE  (y)
BINARY_OP_ADAPTIVE        (+)
STORE_FAST                (y)
LOAD_GLOBAL_ADAPTIVE      (type)
LOAD_FAST                 (self)
PRECALL_ADAPTIVE          (1)
CALL_ADAPTIVE             (1)
STORE_FAST                (cls)
```

Specialized Instructions

`dis.dis(Point.shifted, adaptive=True)`

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST__LOAD_FAST      (dy, self)
LOAD_ATTR_INSTANCE_VALUE  (y)
BINARY_OP_ADAPTIVE        (+)
STORE_FAST                (y)
LOAD_GLOBAL_ADAPTIVE      (type)
LOAD_FAST                 (self)
PRECALL_ADAPTIVE          (1)
CALL_ADAPTIVE             (1)
STORE_FAST                (cls)
```

Specialized Instructions

`dis.dis(Point.shifted, adaptive=True)`

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST__LOAD_FAST      (dy, self)
LOAD_ATTR_INSTANCE_VALUE  (y)
BINARY_OP_ADD_FLOAT       (+)
STORE_FAST                (y)
LOAD_GLOBAL_ADAPTIVE      (type)
LOAD_FAST                 (self)
PRECALL_ADAPTIVE          (1)
CALL_ADAPTIVE             (1)
STORE_FAST                (cls)
```

`BINARY_OP_ADD_FLOAT`

- Check if the left object's class is `float`.
- Check if the right object's class is `float`.
- Add the values together.
- Return the result.

`BINARY_OP_ADD_FLOAT`

- Check if the left object's class is `float`.
- Check if the right object's class is `float`.
- Add the values together.
- Return the result.

Specialized Instructions

`dis.dis(Point.shifted, adaptive=True)`

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST__LOAD_FAST      (dy, self)
LOAD_ATTR_INSTANCE_VALUE  (y)
BINARY_OP_ADD_FLOAT       (+)
STORE_FAST                (y)
LOAD_GLOBAL_ADAPTIVE      (type)
LOAD_FAST                 (self)
PRECALL_ADAPTIVE          (1)
CALL_ADAPTIVE             (1)
STORE_FAST                (cls)
```

Specialized Instructions

`dis.dis(Point.shifted, adaptive=True)`

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST__LOAD_FAST      (dy, self)
LOAD_ATTR_INSTANCE_VALUE  (y)
BINARY_OP_ADD_FLOAT       (+)
STORE_FAST                (y)
LOAD_GLOBAL_ADAPTIVE      (type)
LOAD_FAST                 (self)
PRECALL_ADAPTIVE          (1)
CALL_ADAPTIVE             (1)
STORE_FAST                (cls)
```

Specialized Instructions

`dis.dis(Point.shifted, adaptive=True)`

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST__LOAD_FAST      (dy, self)
LOAD_ATTR_INSTANCE_VALUE  (y)
BINARY_OP_ADD_FLOAT       (+)
STORE_FAST                (y)
LOAD_GLOBAL_BUILTIN       (type)
LOAD_FAST                 (self)
PRECALL_ADAPTIVE          (1)
CALL_ADAPTIVE             (1)
STORE_FAST                (cls)
```

LOAD_GLOBAL_BUILTIN

- Check if the globals dictionary keys are the same as last time.
- Check if the built-in dictionary keys are the same as last time.
- Reach directly into the built-in values at the same offset as last time.
- Return the result.

LOAD_GLOBAL_BUILTIN

- Check if the globals dictionary keys are the same as last time.
- Check if the built-in dictionary keys are the same as last time.
- Reach directly into the built-in values at the same offset as last time.
- Return the result.

Specialized Instructions

`dis.dis(Point.shifted, adaptive=True)`

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST__LOAD_FAST      (dy, self)
LOAD_ATTR_INSTANCE_VALUE  (y)
BINARY_OP_ADD_FLOAT       (+)
STORE_FAST                (y)
LOAD_GLOBAL_BUILTIN       (type)
LOAD_FAST                 (self)
PRECALL_ADAPTIVE          (1)
CALL_ADAPTIVE             (1)
STORE_FAST                (cls)
```

Specialized Instructions

`dis.dis(Point.shifted, adaptive=True)`

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST__LOAD_FAST      (dy, self)
LOAD_ATTR_INSTANCE_VALUE  (y)
BINARY_OP_ADD_FLOAT       (+)
STORE_FAST                (y)
LOAD_GLOBAL_BUILTIN       (type)
LOAD_FAST                 (self)
PRECALL_ADAPTIVE          (1)
CALL_ADAPTIVE             (1)
STORE_FAST                (cls)
```

Specialized Instructions

`dis.dis(Point.shifted, adaptive=True)`

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST__LOAD_FAST      (dy, self)
LOAD_ATTR_INSTANCE_VALUE  (y)
BINARY_OP_ADD_FLOAT       (+)
STORE_FAST                (y)
LOAD_GLOBAL_BUILTIN       (type)
LOAD_FAST                 (self)
PRECALL_NO_KW_TYPE_1      (1)
STORE_FAST                (cls)
```

`PRECALL_NO_KW_TYPE_1`

- Check if the object we're calling is still `type`.
- Get the argument's class.
- Return the result.

`PRECALL_NO_KW_TYPE_1`

- Check if the object we're calling is still `type`.
- Get the argument's class.
- Return the result.

PRECALL_NO_KW_TYPE_1

Specialized Instructions

`dis.dis(Point.shifted, adaptive=True)`

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST__LOAD_FAST      (dy, self)
LOAD_ATTR_INSTANCE_VALUE  (y)
BINARY_OP_ADD_FLOAT       (+)
STORE_FAST                (y)
LOAD_GLOBAL_BUILTIN       (type)
LOAD_FAST                 (self)
PRECALL_NO_KW_TYPE_1      (1)
STORE_FAST                (cls)
```

Specialized Instructions

`dis.dis(Point.shifted, adaptive=True)`

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST__LOAD_FAST      (dy, self)
LOAD_ATTR_INSTANCE_VALUE  (y)
BINARY_OP_ADD_FLOAT       (+)
STORE_FAST                (y)
LOAD_GLOBAL_BUILTIN       (type)
LOAD_FAST                 (self)
PRECALL_NO_KW_TYPE_1      (1)
STORE_FAST                (cls)
```

Specialized Instructions

`dis.dis(Point.shifted, adaptive=True)`

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST__LOAD_FAST      (dy, self)
LOAD_ATTR_INSTANCE_VALUE  (y)
BINARY_OP_ADD_FLOAT       (+)
STORE_FAST                (y)
LOAD_GLOBAL_BUILTIN       (type)
LOAD_FAST                 (self)
PRECALL_NO_KW_TYPE_1      (1)
STORE_FAST                (cls)
```

Specialized Instructions

`dis.dis(Point.shifted, adaptive=True)`

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST__LOAD_FAST      (dy, self)
LOAD_ATTR_INSTANCE_VALUE  (y)
BINARY_OP_ADAPTIVE        (+)
STORE_FAST                (y)
LOAD_GLOBAL_BUILTIN       (type)
LOAD_FAST                 (self)
PRECALL_NO_KW_TYPE_1      (1)
STORE_FAST                (cls)
```

Specialized Instructions

`dis.dis(Point.shifted, adaptive=True)`

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST__LOAD_FAST      (dy, self)
LOAD_ATTR_INSTANCE_VALUE  (y)
BINARY_OP_ADD_INT         (+)
STORE_FAST                (y)
LOAD_GLOBAL_BUILTIN       (type)
LOAD_FAST                 (self)
PRECALL_NO_KW_TYPE_1      (1)
STORE_FAST                (cls)
```

Specialized Instructions

`dis.dis(Point.shifted, adaptive=True)`

```
y = dy + self.y
cls = type(self)
```

```
LOAD_FAST__LOAD_FAST      (dy, self)
LOAD_ATTR_INSTANCE_VALUE  (y)
BINARY_OP_ADD_INT         (+)
STORE_FAST                (y)
LOAD_GLOBAL_BUILTIN       (type)
LOAD_FAST                 (self)
PRECALL_NO_KW_TYPE_1      (1)
STORE_FAST                (cls)
```

Specialized Instructions

`dis.dis(Point.shifted, adaptive=True)`

```
LOAD_FAST__LOAD_FAST      (dy, self)
LOAD_ATTR_INSTANCE_VALUE  (y)
BINARY_OP_ADD_INT         (+)
STORE_FAST                (y)
LOAD_GLOBAL_BUILTIN       (type)
LOAD_FAST                 (self)
PRECALL_NO_KW_TYPE_1      (1)
STORE_FAST                (cls)
```

Inline Caches

`dis.dis(Point.shifted, adaptive=True)`

```
LOAD_FAST__LOAD_FAST      (dy, self)
LOAD_ATTR_INSTANCE_VALUE  (y)
BINARY_OP_ADD_INT         (+)
STORE_FAST                (y)
LOAD_GLOBAL_BUILTIN       (type)
LOAD_FAST                 (self)
PRECALL_NO_KW_TYPE_1      (1)
STORE_FAST                (cls)
```

Inline Caches

`dis.dis(Point.shifted, adaptive=True, show_caches=True)`

```
LOAD_FAST__LOAD_FAST      (dy, self)
LOAD_ATTR_INSTANCE_VALUE  (y)
BINARY_OP_ADD_INT         (+)
STORE_FAST                (y)
LOAD_GLOBAL_BUILTIN       (type)
LOAD_FAST                 (self)
PRECALL_NO_KW_TYPE_1      (1)
STORE_FAST                (cls)
```

Inline Caches

`dis.dis(Point.shifted, adaptive=True, show_caches=True)`

```
LOAD_FAST__LOAD_FAST     (dy, self)
LOAD_ATTR_INSTANCE_VALUE (y)
CACHE                    (counter)
CACHE                    (version)
CACHE                    (version)
CACHE                    (index)
BINARY_OP_ADD_INT        (+)
CACHE                    (counter)
STORE_FAST               (y)
LOAD_GLOBAL_BUILTIN      (type)
CACHE                    (counter)
CACHE                    (index)
CACHE                    (version)
CACHE                    (version)
CACHE                    (version)
LOAD_FAST                (self)
PRECALL_NO_KW_TYPE_1     (1)
CACHE                    (counter)
STORE_FAST               (cls)
```

Inline Caches

`dis.dis(Point.shifted, adaptive=True)`

```
LOAD_FAST__LOAD_FAST      (dy, self)
LOAD_ATTR_INSTANCE_VALUE  (y)
BINARY_OP_ADD_INT         (+)
STORE_FAST                (y)
LOAD_GLOBAL_BUILTIN       (type)
LOAD_FAST                 (self)
PRECALL_NO_KW_TYPE_1      (1)
STORE_FAST                (cls)
```

Inline Caches

`dis.dis(Point.shifted)`

```
LOAD_FAST    (dy)
LOAD_FAST    (self)
LOAD_ATTR    (y)
BINARY_OP    (+)
STORE_FAST   (y)
LOAD_GLOBAL  (type)
LOAD_FAST    (self)
PRECALL      (1)
CALL         (1)
STORE_FAST   (cls)
```

Bytecode

dis.dis(Point.shifted, adaptive=True)

```
RESUME_QUICK
LOAD_FAST__LOAD_FAST          (dx)
LOAD_FAST                     (self)
LOAD_ATTR_INSTANCE_VALUE      (x)
BINARY_OP_ADD_FLOAT           (+)
STORE_FAST__LOAD_FAST         (x)
LOAD_FAST__LOAD_FAST          (dy)
LOAD_FAST                     (self)
LOAD_ATTR_INSTANCE_VALUE      (y)
BINARY_OP_ADD_FLOAT           (+)
STORE_FAST                    (y)
```

```
LOAD_GLOBAL_BUILTIN   (type)
LOAD_FAST             (self)
PRECALL_NO_KW_TYPE_1  (1)
CALL_ADAPTIVE         (1)
STORE_FAST            (cls)
PUSH_NULL
LOAD_FAST__LOAD_FAST  (cls)
LOAD_FAST__LOAD_FAST  (x)
LOAD_FAST             (y)
PRECALL_ADAPTIVE      (2)
CALL_ADAPTIVE         (2)
RETURN_VALUE
```

Bytecode

dis.dis(Point.shifted, adaptive=True)

```
RESUME_QUICK
LOAD_FAST__LOAD_FAST                (dx)
LOAD_FAST                           (self)
LOAD_ATTR_INSTANCE_VALUE            (x)
BINARY_OP_ADD_FLOAT                 (+)
STORE_FAST__LOAD_FAST               (x)
LOAD_FAST__LOAD_FAST                (dy)
LOAD_FAST                           (self)
LOAD_ATTR_INSTANCE_VALUE            (y)
BINARY_OP_ADD_FLOAT                 (+)
STORE_FAST                          (y)
```

```
LOAD_GLOBAL_BUILTIN                 (type)
LOAD_FAST                           (self)
PRECALL_NO_KW_TYPE_1                (1)
CALL_ADAPTIVE                       (1)
STORE_FAST                          (cls)
PUSH_NULL
LOAD_FAST__LOAD_FAST                (cls)
LOAD_FAST__LOAD_FAST                (x)
LOAD_FAST                           (y)
PRECALL_ADAPTIVE                    (2)
CALL_ADAPTIVE                       (2)
RETURN_VALUE
```

Bytecode

`dis.dis(Point.shifted, adaptive=True)`

```
RESUME_QUICK
LOAD_FAST__LOAD_FAST                (dx)
LOAD_FAST                           (self)
LOAD_ATTR_INSTANCE_VALUE            (x)
BINARY_OP_ADD_FLOAT                 (+)
STORE_FAST__LOAD_FAST               (x)
LOAD_FAST__LOAD_FAST                (dy)
LOAD_FAST                           (self)
LOAD_ATTR_INSTANCE_VALUE            (y)
BINARY_OP_ADD_FLOAT                 (+)
STORE_FAST                          (y)
LOAD_GLOBAL_BUILTIN                 (type)
LOAD_FAST                           (self)
PRECALL_NO_KW_TYPE_1                (1)
CALL_ADAPTIVE                       (1)
STORE_FAST                          (cls)
PUSH_NULL
LOAD_FAST__LOAD_FAST                (cls)
LOAD_FAST__LOAD_FAST                (x)
LOAD_FAST                           (y)
PRECALL_ADAPTIVE                    (2)
CALL_ADAPTIVE                       (2)
RETURN_VALUE
```

Source Code

```
RESUME_QUICK
LOAD_FAST__LOAD_FAST                (dx)
LOAD_FAST                           (self)
LOAD_ATTR_INSTANCE_VALUE            (x)
BINARY_OP_ADD_FLOAT                 (+)
STORE_FAST__LOAD_FAST               (x)
LOAD_FAST__LOAD_FAST                (dy)
LOAD_FAST                           (self)
LOAD_ATTR_INSTANCE_VALUE            (y)
BINARY_OP_ADD_FLOAT                 (+)
STORE_FAST                          (y)
```

```
LOAD_GLOBAL_BUILTIN                 (type)
LOAD_FAST                           (self)
PRECALL_NO_KW_TYPE_1                (1)
CALL_ADAPTIVE                       (1)
STORE_FAST                          (cls)
PUSH_NULL
LOAD_FAST__LOAD_FAST                (cls)
LOAD_FAST__LOAD_FAST                (x)
LOAD_FAST                           (y)
PRECALL_ADAPTIVE                    (2)
CALL_ADAPTIVE                       (2)
RETURN_VALUE
```

Source Code

```
x = dx + self.x
```

```
LOAD_FAST__LOAD_FAST      (dy)
LOAD_FAST                 (self)
LOAD_ATTR_INSTANCE_VALUE  (y)
BINARY_OP_ADD_FLOAT       (+)
STORE_FAST                (y)
```

```
LOAD_GLOBAL_BUILTIN   (type)
LOAD_FAST             (self)
PRECALL_NO_KW_TYPE_1  (1)
CALL_ADAPTIVE         (1)
STORE_FAST            (cls)
PUSH_NULL
LOAD_FAST__LOAD_FAST  (cls)
LOAD_FAST__LOAD_FAST  (x)
LOAD_FAST             (y)
PRECALL_ADAPTIVE      (2)
CALL_ADAPTIVE         (2)
RETURN_VALUE
```

Source Code

```
x = dx + self.x

y = dy + self.y
```

```
LOAD_GLOBAL_BUILTIN   (type)
LOAD_FAST             (self)
PRECALL_NO_KW_TYPE_1  (1)
CALL_ADAPTIVE         (1)
STORE_FAST            (cls)
PUSH_NULL
LOAD_FAST__LOAD_FAST  (cls)
LOAD_FAST__LOAD_FAST  (x)
LOAD_FAST             (y)
PRECALL_ADAPTIVE      (2)
CALL_ADAPTIVE         (2)
RETURN_VALUE
```

Source Code

```
x = dx + self.x
```

```
y = dy + self.y
```

```
cls = type(self)
```

```
LOAD_FAST__LOAD_FAST (cls)
LOAD_FAST__LOAD_FAST (x)
LOAD_FAST            (y)
PRECALL_ADAPTIVE     (2)
CALL_ADAPTIVE        (2)
RETURN_VALUE
```

Source Code

```
x = dx + self.x
```

```
y = dy + self.y
```

```
cls = type(self)
```

```
return cls(x, y)
```

Source Code

```
class Point:
    def __init__(self, x: float, y: float) -> None:
        self.x = x
        self.y = y

    def shifted(self, dx: float, dy: float) -> typing.Self:
        x = dx + self.x
        y = dy + self.y
        cls = type(self)
        return cls(x, y)
```

Source Code

```
class Point:
    def __init__(self, x: float, y: float) -> None:
        self.x = x
        self.y = y

    def shifted(self, dx: float, dy: float) -> typing.Self:
        x = dx + self.x
        y = dy + self.y
        cls = type(self)
        return cls(x, y)
```

Specialist

```
class Point:
    def __init__(self, x: float, y: float) -> None:
        self.x = x
        self.y = y

    def shifted(self, dx: float, dy: float) -> typing.Self:
        x = dx + self.x
        y = dy + self.y
        cls = type(self)
        return cls(x, y)
```

Specialist

$ specialist --blue point.py

```
class Point:
    def __init__(self, x: float, y: float) -> None:
        self.x = x
        self.y = y

    def shifted(self, dx: float, dy: float) -> typing.Self:
        x = dx + self.x
        y = dy + self.y
        cls = type(self)
        return cls(x, y)
```

Specialist

$ specialist --blue point.py

```
import typing


class Point:
    def __init__(self, x: float, y: float) -> None:
        self.x = x
        self.y = y

    def shifted(self, dx: float, dy: float) -> typing.Self:
        x = dx + self.x
        y = dy + self.y
        cls = type(self)
        return cls(x, y)


def actually_run_something() -> None:
    p = Point(3.14, 2.72)
    for i in range(100):
        p = p.shifted(19.95, 12.06)


if __name__ == "__main__":
    actually_run_something()
```

Specialist

$ specialist --blue point.py

```
def actually_run_something() -> None:
    p = Point(3.14, 2.72)
    for i in range(100):
        p = p.shifted(19.95, 12.06)


if __name__ == "__main__":
    actually_run_something()
```

Specialist

brandtbucher/specialist

Specialist

Specialist uses fine-grained location information to create visual representations of exactly *where* and *how* CPython 3.11's new specializing, adaptive interpreter optimizes your code.

```
def encode_decode(key: str, text: str) -> str:
    out = []
    for i, t in enumerate(text):
        k = key[i % len(key)]
        out.append(chr(ord(t) ^ ord(k)))
    return "".join(out)
```

Installation

Specialist supports CPython 3.11+ on all platforms.

To install, just run:

```
$ pip install specialist
```

CPython 3.11

CPython 3.12

CPython 3.12

CPython 3.12

Faster Quickening

CPython 3.12

Faster Quickening

```
LOAD_FAST
LOAD_FAST
LOAD_ATTR
BINARY_OP
STORE_FAST
LOAD_GLOBAL
LOAD_FAST
PRECALL
CALL
STORE_FAST
```

CPython 3.12

Faster Quickening

```
# Python 3.11

LOAD_FAST
LOAD_FAST
LOAD_ATTR
BINARY_OP
STORE_FAST
LOAD_GLOBAL
LOAD_FAST
PRECALL
CALL
STORE_FAST
```

CPython 3.12

Faster Quickening

```
# Python 3.11

LOAD_FAST
LOAD_FAST
LOAD_ATTR
BINARY_OP
STORE_FAST
LOAD_GLOBAL
LOAD_FAST
PRECALL
CALL
STORE_FAST
```

```
# Python 3.12

LOAD_FAST__LOAD_FAST
LOAD_ATTR
BINARY_OP
STORE_FAST
LOAD_GLOBAL
LOAD_FAST
CALL
STORE_FAST
```

CPython 3.12

Faster Quickening

```
# Python 3.11

LOAD_FAST
LOAD_FAST
LOAD_ATTR
BINARY_OP
STORE_FAST
LOAD_GLOBAL
LOAD_FAST
PRECALL
CALL
STORE_FAST
```

```
# Python 3.12

LOAD_FAST__LOAD_FAST
LOAD_ATTR
BINARY_OP
STORE_FAST
LOAD_GLOBAL
LOAD_FAST
CALL
STORE_FAST
```

CPython 3.12

Faster Quickening

```
# Python 3.11

LOAD_FAST
LOAD_FAST
LOAD_ATTR
BINARY_OP
STORE_FAST
LOAD_GLOBAL
LOAD_FAST
PRECALL
CALL
STORE_FAST
```

```
# Python 3.12

LOAD_FAST__LOAD_FAST
LOAD_ATTR
BINARY_OP
STORE_FAST
LOAD_GLOBAL
LOAD_FAST
CALL
STORE_FAST
```

CPython 3.12

Faster Quickening

```
point.shifted(...)
```

```
# Python 3.11

LOAD_FAST
LOAD_FAST
LOAD_ATTR
BINARY_OP
STORE_FAST
LOAD_GLOBAL
LOAD_FAST
PRECALL
CALL
STORE_FAST
```

```
# Python 3.12

LOAD_FAST__LOAD_FAST
LOAD_ATTR
BINARY_OP
STORE_FAST
LOAD_GLOBAL
LOAD_FAST
CALL
STORE_FAST
```

CPython 3.12

Faster Quickening

```
point.shifted(...)
point.shifted(...)
```

```
# Python 3.11

LOAD_FAST
LOAD_FAST
LOAD_ATTR
BINARY_OP
STORE_FAST
LOAD_GLOBAL
LOAD_FAST
PRECALL
CALL
STORE_FAST
```

```
# Python 3.12

LOAD_FAST__LOAD_FAST
LOAD_ATTR
BINARY_OP
STORE_FAST
LOAD_GLOBAL
LOAD_FAST
CALL
STORE_FAST
```

CPython 3.12

Faster Quickening

```
point.shifted(...)
point.shifted(...)
point.shifted(...)
```

```
# Python 3.11

LOAD_FAST
LOAD_FAST
LOAD_ATTR
BINARY_OP
STORE_FAST
LOAD_GLOBAL
LOAD_FAST
PRECALL
CALL
STORE_FAST
```

```
# Python 3.12

LOAD_FAST__LOAD_FAST
LOAD_ATTR_INSTANCE_VALUE
BINARY_OP_ADD_FLOAT
STORE_FAST
LOAD_GLOBAL_BUILTIN
LOAD_FAST
CALL_NO_KW_TYPE_1
STORE_FAST
```

CPython 3.12

Faster Quickening

```
point.shifted(...)
point.shifted(...)
point.shifted(...)
point.shifted(...)
```

```
# Python 3.11

LOAD_FAST
LOAD_FAST
LOAD_ATTR
BINARY_OP
STORE_FAST
LOAD_GLOBAL
LOAD_FAST
PRECALL
CALL
STORE_FAST
```

```
# Python 3.12

LOAD_FAST__LOAD_FAST
LOAD_ATTR_INSTANCE_VALUE
BINARY_OP_ADD_FLOAT
STORE_FAST
LOAD_GLOBAL_BUILTIN
LOAD_FAST
CALL_NO_KW_TYPE_1
STORE_FAST
```

CPython 3.12

Faster Quickening

```
point.shifted(...)
point.shifted(...)
point.shifted(...)
point.shifted(...)
point.shifted(...)
```

```
# Python 3.11

LOAD_FAST
LOAD_FAST
LOAD_ATTR
BINARY_OP
STORE_FAST
LOAD_GLOBAL
LOAD_FAST
PRECALL
CALL
STORE_FAST
```

```
# Python 3.12

LOAD_FAST__LOAD_FAST
LOAD_ATTR_INSTANCE_VALUE
BINARY_OP_ADD_FLOAT
STORE_FAST
LOAD_GLOBAL_BUILTIN
LOAD_FAST
CALL_NO_KW_TYPE_1
STORE_FAST
```

CPython 3.12

Faster Quickening

```
point.shifted(...)
point.shifted(...)
point.shifted(...)
point.shifted(...)
point.shifted(...)
point.shifted(...)
```

```
# Python 3.11

LOAD_FAST
LOAD_FAST
LOAD_ATTR
BINARY_OP
STORE_FAST
LOAD_GLOBAL
LOAD_FAST
PRECALL
CALL
STORE_FAST
```

```
# Python 3.12

LOAD_FAST__LOAD_FAST
LOAD_ATTR_INSTANCE_VALUE
BINARY_OP_ADD_FLOAT
STORE_FAST
LOAD_GLOBAL_BUILTIN
LOAD_FAST
CALL_NO_KW_TYPE_1
STORE_FAST
```

CPython 3.12

Faster Quickening

```
point.shifted(...)
point.shifted(...)
point.shifted(...)
point.shifted(...)
point.shifted(...)
point.shifted(...)
```

```
# Python 3.11

LOAD_FAST
LOAD_FAST
LOAD_ATTR
BINARY_OP
STORE_FAST
LOAD_GLOBAL
LOAD_FAST
PRECALL
CALL
STORE_FAST
```

```
# Python 3.12

LOAD_FAST__LOAD_FAST
LOAD_ATTR_INSTANCE_VALUE
BINARY_OP_ADD_FLOAT
STORE_FAST
LOAD_GLOBAL_BUILTIN
LOAD_FAST
CALL_NO_KW_TYPE_1
STORE_FAST
```

CPython 3.12

Faster Quickening

```
point.shifted(...)
point.shifted(...)
point.shifted(...)
point.shifted(...)
point.shifted(...)
point.shifted(...)
point.shifted(...)
point.shifted(...)
```

```
# Python 3.11

LOAD_FAST__LOAD_FAST
LOAD_ATTR
BINARY_OP_ADAPTIVE
STORE_FAST
LOAD_GLOBAL_ADAPTIVE
LOAD_FAST
PRECALL_ADAPTIVE
CALL_ADAPTIVE
STORE_FAST
```

```
# Python 3.12

LOAD_FAST__LOAD_FAST
LOAD_ATTR_INSTANCE_VALUE
BINARY_OP_ADD_FLOAT
STORE_FAST
LOAD_GLOBAL_BUILTIN
LOAD_FAST
CALL_NO_KW_TYPE_1
STORE_FAST
```

CPython 3.12

Faster Quickening

```
point.shifted(...)          # Python 3.11                 # Python 3.12
point.shifted(...)
point.shifted(...)          LOAD_FAST__LOAD_FAST        LOAD_FAST__LOAD_FAST
point.shifted(...)          LOAD_ATTR_INSTANCE_VALUE    LOAD_ATTR_INSTANCE_VALUE
point.shifted(...)          BINARY_OP_ADD_FLOAT         BINARY_OP_ADD_FLOAT
point.shifted(...)          STORE_FAST                  STORE_FAST
point.shifted(...)          LOAD_GLOBAL_BUILTIN         LOAD_GLOBAL_BUILTIN
point.shifted(...)          LOAD_FAST                   LOAD_FAST
point.shifted(...)          PRECALL_NO_KW_TYPE_1        CALL_NO_KW_TYPE_1
                            STORE_FAST                  STORE_FAST
```

CPython 3.12

More Specializations

CPython 3.12

More Specializations

```
BINARY_OP_ADD_FLOAT
BINARY_OP_ADD_INT
BINARY_OP_ADD_UNICODE
BINARY_OP_INPLACE_ADD_UNICODE
BINARY_OP_MULTIPLY_FLOAT
BINARY_OP_MULTIPLY_INT
BINARY_OP_SUBTRACT_FLOAT
BINARY_OP_SUBTRACT_INT

BINARY_SUBSCR_DICT
BINARY_SUBSCR_GETITEM
BINARY_SUBSCR_LIST_INT
BINARY_SUBSCR_TUPLE_INT

LOAD_GLOBAL_BUILTIN
LOAD_GLOBAL_MODULE

STORE_SUBSCR_DICT
STORE_SUBSCR_LIST_INT

UNPACK_SEQUENCE_LIST
UNPACK_SEQUENCE_TUPLE
UNPACK_SEQUENCE_TWO_TUPLE
```

```
PRECALL_BOUND_METHOD
PRECALL_BUILTIN_CLASS
PRECALL_BUILTIN_FAST_WITH_KEYWORDS
PRECALL_METHOD_DESCRIPTOR_FAST_WITH_KEYWORDS
PRECALL_NO_KW_BUILTIN_FAST
PRECALL_NO_KW_BUILTIN_O
PRECALL_NO_KW_ISINSTANCE
PRECALL_NO_KW_LEN
PRECALL_NO_KW_LIST_APPEND
PRECALL_NO_KW_METHOD_DESCRIPTOR_FAST
PRECALL_NO_KW_METHOD_DESCRIPTOR_NOARGS
PRECALL_NO_KW_METHOD_DESCRIPTOR_O
PRECALL_NO_KW_STR_1
PRECALL_NO_KW_TUPLE_1
PRECALL_NO_KW_TYPE_1
PRECALL_PYFUNC

CALL_PY_EXACT_ARGS
CALL_PY_WITH_DEFAULTS

COMPARE_OP_FLOAT_JUMP
COMPARE_OP_INT_JUMP
COMPARE_OP_STR_JUMP
```

```
LOAD_ATTR_INSTANCE_VALUE
LOAD_ATTR_MODULE
LOAD_ATTR_SLOT
LOAD_ATTR_WITH_HINT
LOAD_GLOBAL_BUILTIN
LOAD_GLOBAL_MODULE

LOAD_METHOD_CLASS
LOAD_METHOD_MODULE
LOAD_METHOD_NO_DICT
LOAD_METHOD_WITH_DICT
LOAD_METHOD_WITH_VALUES

STORE_ATTR_INSTANCE_VALUE
STORE_ATTR_SLOT
STORE_ATTR_WITH_HINT
```

CPython 3.12

More Specializations

```
BINARY_OP_ADD_FLOAT
BINARY_OP_ADD_INT
BINARY_OP_ADD_UNICODE
BINARY_OP_INPLACE_ADD_UNICODE
BINARY_OP_MULTIPLY_FLOAT
BINARY_OP_MULTIPLY_INT
BINARY_OP_SUBTRACT_FLOAT
BINARY_OP_SUBTRACT_INT

BINARY_SUBSCR_DICT
BINARY_SUBSCR_GETITEM
BINARY_SUBSCR_LIST_INT
BINARY_SUBSCR_TUPLE_INT

LOAD_GLOBAL_BUILTIN
LOAD_GLOBAL_MODULE

STORE_SUBSCR_DICT
STORE_SUBSCR_LIST_INT

UNPACK_SEQUENCE_LIST
UNPACK_SEQUENCE_TUPLE
UNPACK_SEQUENCE_TWO_TUPLE
```

```
CALL_BOUND_METHOD_EXACT_ARGS
CALL_BUILTIN_CLASS
CALL_BUILTIN_FAST_WITH_KEYWORDS
CALL_METHOD_DESCRIPTOR_FAST_WITH_KEYWORDS
CALL_NO_KW_BUILTIN_FAST
CALL_NO_KW_BUILTIN_O
CALL_NO_KW_ISINSTANCE
CALL_NO_KW_LEN
CALL_NO_KW_LIST_APPEND
CALL_NO_KW_METHOD_DESCRIPTOR_FAST
CALL_NO_KW_METHOD_DESCRIPTOR_NOARGS
CALL_NO_KW_METHOD_DESCRIPTOR_O
CALL_NO_KW_STR_1
CALL_NO_KW_TUPLE_1
CALL_NO_KW_TYPE_1
CALL_PYFUNC
CALL_PY_EXACT_ARGS
CALL_PY_WITH_DEFAULTS

COMPARE_OP_FLOAT_JUMP
COMPARE_OP_INT_JUMP
COMPARE_OP_STR_JUMP
```

```
LOAD_ATTR_INSTANCE_VALUE
LOAD_ATTR_MODULE
LOAD_ATTR_SLOT
LOAD_ATTR_WITH_HINT
LOAD_GLOBAL_BUILTIN
LOAD_GLOBAL_MODULE

LOAD_METHOD_CLASS
LOAD_METHOD_MODULE
LOAD_METHOD_NO_DICT
LOAD_METHOD_WITH_DICT
LOAD_METHOD_WITH_VALUES

STORE_ATTR_INSTANCE_VALUE
STORE_ATTR_SLOT
STORE_ATTR_WITH_HINT
```

CPython 3.12

More Specializations

```
BINARY_OP_ADD_FLOAT
BINARY_OP_ADD_INT
BINARY_OP_ADD_UNICODE
BINARY_OP_INPLACE_ADD_UNICODE
BINARY_OP_MULTIPLY_FLOAT
BINARY_OP_MULTIPLY_INT
BINARY_OP_SUBTRACT_FLOAT
BINARY_OP_SUBTRACT_INT

BINARY_SUBSCR_DICT
BINARY_SUBSCR_GETITEM
BINARY_SUBSCR_LIST_INT
BINARY_SUBSCR_TUPLE_INT

LOAD_GLOBAL_BUILTIN
LOAD_GLOBAL_MODULE

STORE_SUBSCR_DICT
STORE_SUBSCR_LIST_INT

UNPACK_SEQUENCE_LIST
UNPACK_SEQUENCE_TUPLE
UNPACK_SEQUENCE_TWO_TUPLE
```

```
CALL_BOUND_METHOD_EXACT_ARGS
CALL_BUILTIN_CLASS
CALL_BUILTIN_FAST_WITH_KEYWORDS
CALL_METHOD_DESCRIPTOR_FAST_WITH_KEYWORDS
CALL_NO_KW_BUILTIN_FAST
CALL_NO_KW_BUILTIN_O
CALL_NO_KW_ISINSTANCE
CALL_NO_KW_LEN
CALL_NO_KW_LIST_APPEND
CALL_NO_KW_METHOD_DESCRIPTOR_FAST
CALL_NO_KW_METHOD_DESCRIPTOR_NOARGS
CALL_NO_KW_METHOD_DESCRIPTOR_O
CALL_NO_KW_STR_1
CALL_NO_KW_TUPLE_1
CALL_NO_KW_TYPE_1
CALL_PYFUNC
CALL_PY_EXACT_ARGS
CALL_PY_WITH_DEFAULTS

COMPARE_OP_FLOAT
COMPARE_OP_INT
COMPARE_OP_STR
```

```
LOAD_ATTR_INSTANCE_VALUE
LOAD_ATTR_MODULE
LOAD_ATTR_SLOT
LOAD_ATTR_WITH_HINT
LOAD_GLOBAL_BUILTIN
LOAD_GLOBAL_MODULE

LOAD_METHOD_CLASS
LOAD_METHOD_MODULE
LOAD_METHOD_NO_DICT
LOAD_METHOD_WITH_DICT
LOAD_METHOD_WITH_VALUES

STORE_ATTR_INSTANCE_VALUE
STORE_ATTR_SLOT
STORE_ATTR_WITH_HINT
```

CPython 3.12

More Specializations

```
BINARY_OP_ADD_FLOAT
BINARY_OP_ADD_INT
BINARY_OP_ADD_UNICODE
BINARY_OP_INPLACE_ADD_UNICODE
BINARY_OP_MULTIPLY_FLOAT
BINARY_OP_MULTIPLY_INT
BINARY_OP_SUBTRACT_FLOAT
BINARY_OP_SUBTRACT_INT

BINARY_SUBSCR_DICT
BINARY_SUBSCR_GETITEM
BINARY_SUBSCR_LIST_INT
BINARY_SUBSCR_TUPLE_INT

LOAD_GLOBAL_BUILTIN
LOAD_GLOBAL_MODULE

STORE_SUBSCR_DICT
STORE_SUBSCR_LIST_INT

UNPACK_SEQUENCE_LIST
UNPACK_SEQUENCE_TUPLE
UNPACK_SEQUENCE_TWO_TUPLE
```

```
CALL_BOUND_METHOD_EXACT_ARGS
CALL_BUILTIN_CLASS
CALL_BUILTIN_FAST_WITH_KEYWORDS
CALL_METHOD_DESCRIPTOR_FAST_WITH_KEYWORDS
CALL_NO_KW_BUILTIN_FAST
CALL_NO_KW_BUILTIN_O
CALL_NO_KW_ISINSTANCE
CALL_NO_KW_LEN
CALL_NO_KW_LIST_APPEND
CALL_NO_KW_METHOD_DESCRIPTOR_FAST
CALL_NO_KW_METHOD_DESCRIPTOR_NOARGS
CALL_NO_KW_METHOD_DESCRIPTOR_O
CALL_NO_KW_STR_1
CALL_NO_KW_TUPLE_1
CALL_NO_KW_TYPE_1
CALL_PYFUNC
CALL_PY_EXACT_ARGS
CALL_PY_WITH_DEFAULTS

COMPARE_OP_FLOAT
COMPARE_OP_INT
COMPARE_OP_STR
```

```
LOAD_ATTR_CLASS
LOAD_ATTR_INSTANCE_VALUE
LOAD_ATTR_METHOD_LAZY_DICT
LOAD_ATTR_METHOD_NO_DICT
LOAD_ATTR_METHOD_WITH_VALUES
LOAD_ATTR_MODULE
LOAD_ATTR_SLOT
LOAD_ATTR_WITH_HINT

STORE_ATTR_INSTANCE_VALUE
STORE_ATTR_SLOT
STORE_ATTR_WITH_HINT
```

CPython 3.12

More Specializations

```
BINARY_OP_ADD_FLOAT
BINARY_OP_ADD_INT
BINARY_OP_ADD_UNICODE
BINARY_OP_INPLACE_ADD_UNICODE
BINARY_OP_MULTIPLY_FLOAT
BINARY_OP_MULTIPLY_INT
BINARY_OP_SUBTRACT_FLOAT
BINARY_OP_SUBTRACT_INT

BINARY_SUBSCR_DICT
BINARY_SUBSCR_GETITEM
BINARY_SUBSCR_LIST_INT
BINARY_SUBSCR_TUPLE_INT

LOAD_GLOBAL_BUILTIN
LOAD_GLOBAL_MODULE

STORE_SUBSCR_DICT
STORE_SUBSCR_LIST_INT

UNPACK_SEQUENCE_LIST
UNPACK_SEQUENCE_TUPLE
UNPACK_SEQUENCE_TWO_TUPLE
```

```
CALL_BOUND_METHOD_EXACT_ARGS
CALL_BUILTIN_CLASS
CALL_BUILTIN_FAST_WITH_KEYWORDS
CALL_METHOD_DESCRIPTOR_FAST_WITH_KEYWORDS
CALL_NO_KW_BUILTIN_FAST
CALL_NO_KW_BUILTIN_O
CALL_NO_KW_ISINSTANCE
CALL_NO_KW_LEN
CALL_NO_KW_LIST_APPEND
CALL_NO_KW_METHOD_DESCRIPTOR_FAST
CALL_NO_KW_METHOD_DESCRIPTOR_NOARGS
CALL_NO_KW_METHOD_DESCRIPTOR_O
CALL_NO_KW_STR_1
CALL_NO_KW_TUPLE_1
CALL_NO_KW_TYPE_1
CALL_PYFUNC
CALL_PY_EXACT_ARGS
CALL_PY_WITH_DEFAULTS

COMPARE_OP_FLOAT
COMPARE_OP_INT
COMPARE_OP_STR
```

```
LOAD_ATTR_CLASS
LOAD_ATTR_GETATTRIBUTE_OVERRIDDEN
LOAD_ATTR_INSTANCE_VALUE
LOAD_ATTR_METHOD_LAZY_DICT
LOAD_ATTR_METHOD_NO_DICT
LOAD_ATTR_METHOD_WITH_VALUES
LOAD_ATTR_MODULE
LOAD_ATTR_PROPERTY
LOAD_ATTR_SLOT
LOAD_ATTR_WITH_HINT

STORE_ATTR_INSTANCE_VALUE
STORE_ATTR_SLOT
STORE_ATTR_WITH_HINT

FOR_ITER_LIST
FOR_ITER_TUPLE
FOR_ITER_RANGE
FOR_ITER_GEN

SEND_GEN
```

CPython 3.12

More Specializations

```
LOAD_ATTR_GETATTRIBUTE_OVERRIDDEN
LOAD_ATTR_PROPERTY

FOR_ITER_LIST
FOR_ITER_TUPLE
FOR_ITER_RANGE
FOR_ITER_GEN

SEND_GEN
```

CPython 3.12

More Specializations

```
LOAD_ATTR_GETATTRIBUTE_OVERRIDDEN
LOAD_ATTR_PROPERTY

FOR_ITER_LIST
FOR_ITER_TUPLE
FOR_ITER_RANGE
FOR_ITER_GEN

SEND_GEN
```

CPython 3.12

More Specializations

```
LOAD_ATTR_GETATTRIBUTE_OVERRIDDEN
LOAD_ATTR_PROPERTY

FOR_ITER_LIST
FOR_ITER_TUPLE
FOR_ITER_RANGE
FOR_ITER_GEN

SEND_GEN
```

CPython 3.12

More Specializations

```
LOAD_ATTR_GETATTRIBUTE_OVERRIDDEN
LOAD_ATTR_PROPERTY

FOR_ITER_LIST
FOR_ITER_TUPLE
FOR_ITER_RANGE
FOR_ITER_GEN

SEND_GEN
```

CPython 3.12

More Specializations

```
LOAD_ATTR_GETATTRIBUTE_OVERRIDDEN
LOAD_ATTR_PROPERTY

FOR_ITER_LIST
FOR_ITER_TUPLE
FOR_ITER_RANGE
FOR_ITER_GEN

SEND_GEN
```

CPython 3.12

More Specializations

```
BINARY_OP_ADD_FLOAT
BINARY_OP_ADD_INT
BINARY_OP_ADD_UNICODE
BINARY_OP_INPLACE_ADD_UNICODE
BINARY_OP_MULTIPLY_FLOAT
BINARY_OP_MULTIPLY_INT
BINARY_OP_SUBTRACT_FLOAT
BINARY_OP_SUBTRACT_INT

BINARY_SUBSCR_DICT
BINARY_SUBSCR_GETITEM
BINARY_SUBSCR_LIST_INT
BINARY_SUBSCR_TUPLE_INT

LOAD_GLOBAL_BUILTIN
LOAD_GLOBAL_MODULE

STORE_SUBSCR_DICT
STORE_SUBSCR_LIST_INT

UNPACK_SEQUENCE_LIST
UNPACK_SEQUENCE_TUPLE
UNPACK_SEQUENCE_TWO_TUPLE
```

```
CALL_BOUND_METHOD_EXACT_ARGS
CALL_BUILTIN_CLASS
CALL_BUILTIN_FAST_WITH_KEYWORDS
CALL_METHOD_DESCRIPTOR_FAST_WITH_KEYWORDS
CALL_NO_KW_BUILTIN_FAST
CALL_NO_KW_BUILTIN_O
CALL_NO_KW_ISINSTANCE
CALL_NO_KW_LEN
CALL_NO_KW_LIST_APPEND
CALL_NO_KW_METHOD_DESCRIPTOR_FAST
CALL_NO_KW_METHOD_DESCRIPTOR_NOARGS
CALL_NO_KW_METHOD_DESCRIPTOR_O
CALL_NO_KW_STR_1
CALL_NO_KW_TUPLE_1
CALL_NO_KW_TYPE_1
CALL_PYFUNC
CALL_PY_EXACT_ARGS
CALL_PY_WITH_DEFAULTS

COMPARE_OP_FLOAT
COMPARE_OP_INT
COMPARE_OP_STR
```

```
LOAD_ATTR_CLASS
LOAD_ATTR_GETATTRIBUTE_OVERRIDDEN
LOAD_ATTR_INSTANCE_VALUE
LOAD_ATTR_METHOD_LAZY_DICT
LOAD_ATTR_METHOD_NO_DICT
LOAD_ATTR_METHOD_WITH_VALUES
LOAD_ATTR_MODULE
LOAD_ATTR_PROPERTY
LOAD_ATTR_SLOT
LOAD_ATTR_WITH_HINT

STORE_ATTR_INSTANCE_VALUE
STORE_ATTR_SLOT
STORE_ATTR_WITH_HINT

FOR_ITER_LIST
FOR_ITER_TUPLE
FOR_ITER_RANGE
FOR_ITER_GEN

SEND_GEN
```

CPython 3.12

More Specializations

CPython 3.12

Smaller Caches

CPython 3.12

Smaller Caches

```
BINARY_SUBSCR   CALL                COMPARE_OP      LOAD_GLOBAL
```

CPython 3.12

Smaller Caches

```
                  PRECALL
                  CACHE
BINARY_SUBSCR     CALL          COMPARE_OP    LOAD_GLOBAL
CACHE             CACHE         CACHE         CACHE
CACHE             CACHE         CACHE         CACHE
CACHE             CACHE                       CACHE
CACHE             CACHE                       CACHE
                                              CACHE
```

CPython 3.12

Smaller Caches

```
                  PRECALL
                  CACHE
BINARY_SUBSCR     CALL            COMPARE_OP      LOAD_GLOBAL
CACHE             CACHE           CACHE           CACHE
CACHE             CACHE           CACHE           CACHE
CACHE             CACHE                           CACHE
CACHE             CACHE                           CACHE
                                                  CACHE
```

CPython 3.12

Smaller Caches

```
BINARY_SUBSCR   CALL            COMPARE_OP      LOAD_GLOBAL
CACHE           CACHE           CACHE           CACHE
CACHE           CACHE           CACHE           CACHE
CACHE           CACHE                           CACHE
CACHE           CACHE                           CACHE
                                                CACHE
                                                CACHE
```

CPython 3.12

Smaller Caches

```
BINARY_SUBSCR
CACHE
CACHE
CACHE
CACHE
```

```
CALL
CACHE
CACHE
CACHE
CACHE
```

```
COMPARE_OP
CACHE
CACHE
```

```
LOAD_GLOBAL
CACHE
CACHE
CACHE
CACHE
CACHE
```

CPython 3.12

Smaller Caches

```
BINARY_SUBSCR
CACHE

CALL
CACHE
CACHE
CACHE

COMPARE_OP
CACHE

LOAD_GLOBAL
CACHE
CACHE
CACHE
CACHE
```

CPython 3.12

Smaller Caches

```
BINARY_SUBSCR
CACHE
```

```
CALL
CACHE
CACHE
CACHE
```

```
COMPARE_OP
CACHE
```

```
LOAD_GLOBAL
CACHE
CACHE
CACHE
CACHE
```

PEP 659:
Specializing Adaptive Interpreter

Thank you!

@brandtbucher

Thank you!

@brandtbucher | brandt@python.org