

Throughout the summer, I improved the support for singledispatch in both mypy and mypyc by:

- Adding a mypy plugin that preserves the type of the main fallback singledispatch function and checks registered functions
- Special casing singledispatch functions in mypyc by generating specialized IR for them

Mypy

I added the singledispatch plugin found [here](#) so that mypy would:

- Preserve the type of the main singledispatch function and use that type signature to type check calls to that function
- Check that the first argument of all registered implementations is a subtype of the first argument
- Check that singledispatch functions have at least one positional argument

An earlier version of this plugin tried to find the type signatures of all registered implementations and the main singledispatch function, determine which would be called at each call site using the type of the first argument passed to the function, and then type check the other arguments using that signature. This turned out to be not feasible because:

- Mypy doesn't always know what the type of the first argument is enough to figure out which implementation is used
- The results of this type checking can silently become incorrect if another registered implementation is added after mypy type checks a specific call

In order to avoid these problems, the current singledispatch support requires that the first argument of all registered implementations is a subtype of the first argument of the main singledispatch function, which allows us to type check calls to singledispatch functions without knowing about all of the registered implementations.

PRs:

- [#10627](#): Adding mypy tests for singledispatch
- [#10675](#): Fix a bug in the plugin API for method decorators
- [#10694](#): Initial support for type checking singledispatch functions
- [#10716](#), [#10727](#): Fixing bugs in mypy that the new singledispatch plugin found
- [#10717](#): A (now unnecessary) change to the plugin API for the singledispatch plugin to use
- [#10756](#): Type check registered implementations correctly when they're being called directly
- [#10831](#): Fix a crash in the singledispatch plugin

Mypyc

I special cased the singledispatch decorator so that singledispatch functions would be compiled into 2 functions:

- A main singledispatch function that is the user defined function that the singledispatch decorator is applied to
- A dispatch function that checks the type of the first argument and calls either one of the registered implementations or the main singledispatch function

PRs:

- [#10589](#): Adding tests for singledispatch
- [#10753](#): Initial support for compiling singledispatch functions
- [#10779](#): Avoid changing the AST during mypyc compilation
- [#10795](#): Add support for dispatching on ABCs, generator functions as main singledispatch functions, and decorated registered implementations
- [#10844](#): A now unnecessary change to the order that registered implementations are checked at runtime
- [#10880](#): Support registering functions in different file from main singledispatch function
- [#10888](#): Improve performance by using native function calls in dispatch functions when possible (instead of going through the python API)
- [#10930](#): Check the MRO of the first argument's type at runtime to avoid edge cases where our implementation was different from the standard library implementation
- [#10968](#): Add support for registering functions at runtime
- [#10972](#): Speed up singledispatch calls by using a cache
- [#10981](#): Use native calls to singledispatch functions to improve performance

Underscore functions

Because using functions named “_” (a single underscore) is common for registered implementations of singledispatch functions, I also improved the support for underscore functions in both mypy and mypyc.

PRs:

- [#10811](#): Allow redefining underscore functions when type checking
- [#10830](#): Allow redefining underscore functions when compiling with mypyc

Future Work

There are multiple possible ways for the features added this summer to be extended:

- Support the singledispatchmethod decorator by reusing the infrastructure added for singledispatch functions
- Add more type checking features to the singledispatch plugin (like checking if registered functions have the same first argument type as the main singledispatch function)
- Refactor the singledispatch mypyc support by moving all of the singledispatch-related code to its own file