Benchmarking ADT struct representation vs Vector struct representation

This notebook compares the representation of structures on Boogie level as:

- Vectors of the universal `$Value` type. Values for all fields are represented in boxed ($Value) representation. Selecting and updating fields amounts to vector indexing. On select/update values need to be unboxed/boxed.
- Abstract data types. Values of fields are stored in unboxed representation unless their type is generic. Equality on universal values has to be implemented by a large case distinction of the multiple ADT variants. However, equality is extensional unless a struct contains a transient field of vector type, which breaks extensionality.

Preparation

Load the prover-lab crate. This may take *long* (minutes) the first time the Jupyter server is started because it compiles a lot Rust sources.

In [2]:
```
:sccache 1
:dep prover-lab = { path = "../.." }
```

Out[2]:
```
sccache: true
```

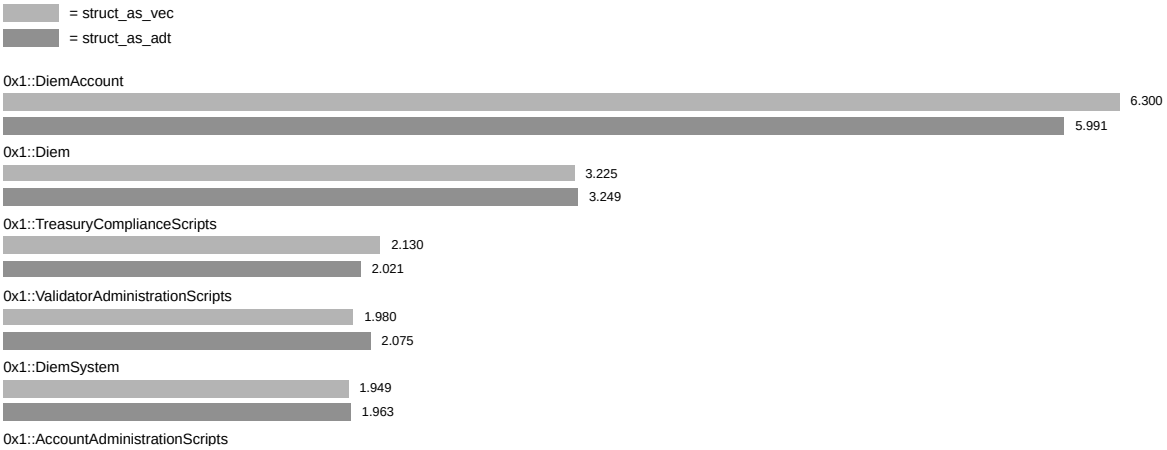
Make functions from the benchmark module available:

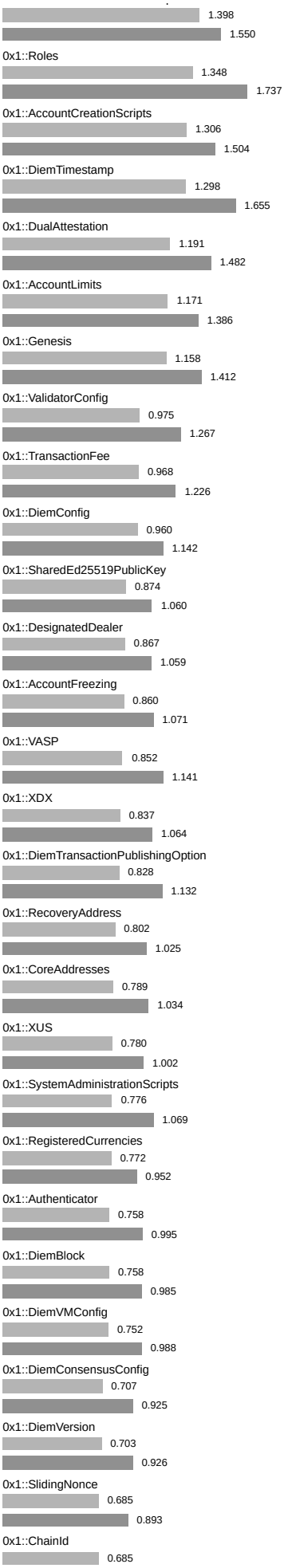
In [3]:
```
use prover_lab::benchmark::*;
```

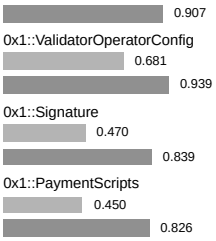
Module Verification Time

In [6]:
```
let mut struct_as_vec_mod = read_benchmark("struct_as_vec.mod_data")?;
let mut struct_as_adt_mod = read_benchmark("struct_as_adt.mod_data")?;
struct_as_vec_mod.sort(); // Will also determine order of other samples.
plot_benchmarks(&[&struct_as_vec_mod, &struct_as_adt_mod])
```

Out[6]:




1.398
1.550
0x1::Roles
1.348
1.737
0x1::AccountCreationScripts
1.306
1.504
0x1::DiemTimestamp
1.298
1.655
0x1::DualAttestation
1.191
1.482
0x1::AccountLimits
1.171
1.386
0x1::Genesis
1.158
1.412
0x1::ValidatorConfig
0.975
1.267
0x1::TransactionFee
0.968
1.226
0x1::DiemConfig
0.960
1.142
0x1::SharedEd25519PublicKey
0.874
1.060
0x1::DesignatedDealer
0.867
1.059
0x1::AccountFreezing
0.860
1.071
0x1::VASP
0.852
1.141
0x1::XDX
0.837
1.064
0x1::DiemTransactionPublishingOption
0.828
1.132
0x1::RecoveryAddress
0.802
1.025
0x1::CoreAddresses
0.789
1.034
0x1::XUS
0.780
1.002
0x1::SystemAdministrationScripts
0.776
1.069
0x1::RegisteredCurrencies
0.772
0.952
0x1::Authenticator
0.758
0.995
0x1::DiemBlock
0.758
0.985
0x1::DiemVMConfig
0.752
0.988
0x1::DiemConsensusConfig
0.707
0.925
0x1::DiemVersion
0.703
0.926
0x1::SlidingNonce
0.685
0.893
0x1::ChainId
0.685



Top 20 by Function

In [7]:

```
let mut struct_as_vec_fun = read_benchmark("struct_as_vec.fun_data")?;
let mut struct_as_adt_fun = read_benchmark("struct_as_adt.fun_data")?;
struct_as_vec_fun.sort(); // Will also determine order of other samples.
struct_as_vec_fun.take(20);
plot_benchmarks(&[&struct_as_vec_fun, &struct_as_adt_fun])
```

Out[7]:


= struct_as_vec
= struct_as_adt
DiemSystem::update_config_and_reconfigure
timeout
timeout
PaymentScripts::peer_to_peer_with_metadata
timeout
1.439
AccountCreationScripts::create_child_vasp_account
timeout
3.617
TreasuryComplianceScripts::preburn
timeout
1.192
DiemAccount::pay_from
1.672
1.501
DiemAccount::preburn
1.254
1.277
ValidatorAdministrationScripts::remove_validator_and_reconfigure
1.215
1.165
ValidatorAdministrationScripts::set_validator_config_and_reconfigure
1.209
1.050
DiemAccount::cancel_burn
1.178
1.212
DiemTimestamp::set_time_has_started
1.163
1.416
Genesis::initialize
1.162
1.412
ValidatorAdministrationScripts::add_validator_and_reconfigure
1.153
1.181
DiemSystem::remove_validator
1.127
1.214
TreasuryComplianceScripts::cancel_burn_with_amount
1.104
1.204
DiemAccount::deposit
1.029
1.158
DiemAccount::withdraw_from
0.987
1.079
DiemSystem::add_validator
0.983
1.051
DiemAccount::tiered_mint
0.970
1.102
DiemAccount::create_designated_dealer
0.928
1.110
Diem::burn_now
0.926
1.115

In [ ]: