Navigating Your Way Toward UVM version 1.2

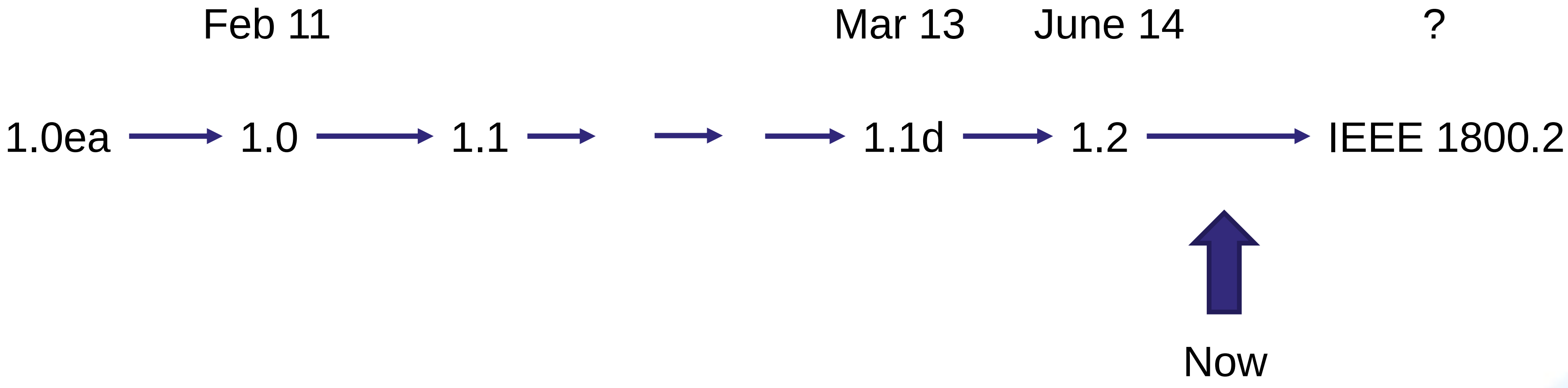
David C Black and Eileen Hickey, Doulos Inc. *Paper*

John Aynsley, Doulos Ltd *Presentation*

23-25 June 2015

SNUG Europe

Evolution of UVM in Accellera



Feb 11 | Mar 13 | June 14 | ?

1.0ea → 1.0 → 1.1 → → → 1.1d → 1.2 → IEEE 1800.2

Now

UVM-1.2

- Is the standard!
- Will be the basis for IEEE standard 1800.2
- So use it ! (?)

Agenda

Features

Backward compatibility and deprecation

Simulation speed

Conclusions

Notable New Features

- get/set_starting_phase
- set_automatic_phase_objection
- get_objection_count
- set_propagate_mode

- Factory override undo

- Sequence library now official
- Sequencer get/get_next_item etc now official

- Object-based reporting API

starting_phase

UVM-1.1d

```
seq.starting_phase = phase;
seq.start(...);
```

```
starting_phase.raise_objection(this);
```

UVM-1.2

```
seq.set_starting_phase(phase);
seq.start(...);
```

```
starting_phase = get_starting_phase();
starting_phase.raise_objection(this);
```

set_automatic_phase_objection

```
seq = my_sequence::type_id::create("seq");
seq.set_starting_phase(phase);

seq.set_automatic_phase_objection(1);

seq.start(...);
```

Equivalent to

```
task pre_start;
  phase = get_starting_phase();
  if (phase != null) phase.raise_objection(this);
```

```
task post_start;
  phase = get_starting_phase();
  if (phase != null) phase.drop_objection(this);
```

get_objection_count

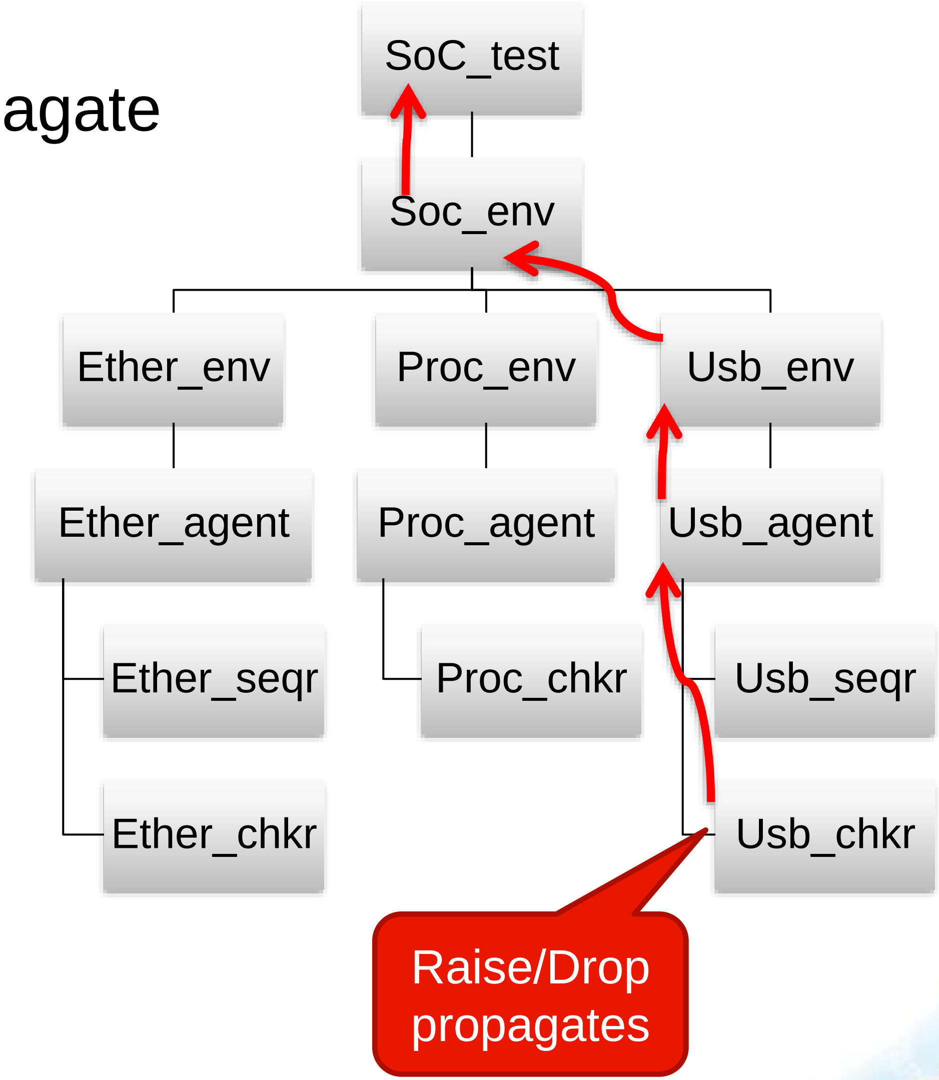
```
task run_phase(uvm_phase phase);
  ...

  phase.drop_objection(this, "",
    phase.get_objection_count(this) );
```

Propagating Objections

- UVM 1.1 - objections propagate up component hierarchy
  - Help with debug?
  - Speed overhead

- UVM 1.2 - can disable propagation
  - Applies to all phases



set_propagate_mode

```
task run_phase(uvm_phase phase);
  uvm_objection objection;
  objection = phase.get_objection();

  objection.set_propagate_mode(0);

  objection.set_drain_time(uvm_top, 100ns);
  phase.raise_objection(this);
```

Factory Override Undo

```
tx_t::type_id::set_type_override( tx_t::get_type() );
```

Same type!

Object-Based Reporting API

```
`uvm_info("", $sformatf("i = %0d", i), UVM_MEDIUM)
```

```
UVM_INFO foo.sv(381) @ 95: uvm_test_top [] i = 99
```

```
`uvm_info_begin("", "", UVM_MEDIUM)
  `uvm_message_add_int(i, UVM_DEC, "i")
`uvm_info_end
```

From uvm_printer

```
UVM_INFO foo.sv(381) @ 95: uvm_test_top []
 +-------------------------------------------------------------------
 +Name                Type                                    Size  Value
 +-------------------------------------------------------------------
 +element_container   uvm_report_message_element_container     -    @791
 +  i                 integral                                 32   'd99
 +-------------------------------------------------------------------
```

New Methods and Macros

```
class uvm_report_message extends uvm_object;
  ...
  function void add_int(...);
  function void add_string(...);
  function void add_object(...);
```

```
`uvm_info_begin(id, message, verbosity)
  `uvm_message_add_tag("Label", value)
  `uvm_message_add_int(value, UVM_DEC, "Label")
  `uvm_message_add_string(value, "Label")
  `uvm_message_add_object(value, "Label")
`uvm_info_end
```

Agenda

Features

Backward compatibility and deprecation

Simulation speed

Conclusions

UVM 1.2 Detailed Changes

- 24 semantic and API changes (clean-up)
- 76 bug fixes
- 13 changes incompatible with UVM-1.1d
  - **Perl script uvm11-to-uvm12.pl does trivial text replacement**

Deprecated Features

- starting_phase
- set_config_int, get_config_int/string/object

- global_stop_request, enable_stop_interrupt, ...
- uvm_test_done, ...
- run_hooks, report_info_hook, ...
- `uvm_sequence_utils, `uvm_sequencer_utils, ...
- uvm_sequence_base::count, max_random_count, ...
- seq_kind, pick_sequence, num_sequences, ...

UVM_NO_DEPRECATED

- By default, no warning of deprecated feature

```
+define+UVM_NO_DEPRECATED
```

Deprecated features give errors

- Or can check the version

```
`ifdef UVM_POST_VERSION_1_1
  seq.set_starting_phase(phase);
`else
  seq.starting_phase = phase;
`endif
```

uvm_object Constructors

- Every uvm_object must have a constructor in UVM-1.2

```
class my_config extends uvm_object;
  `uvm_object_utils(myconfig)

  function new(string name = "");
  endfunction

endclass
```

```
+define+UVM_OBJECT_DO_NOT_NEED_CONSTRUCTOR
```

Restores UVM-1.1d behavior

Agenda

Features

Backward compatibility and deprecation

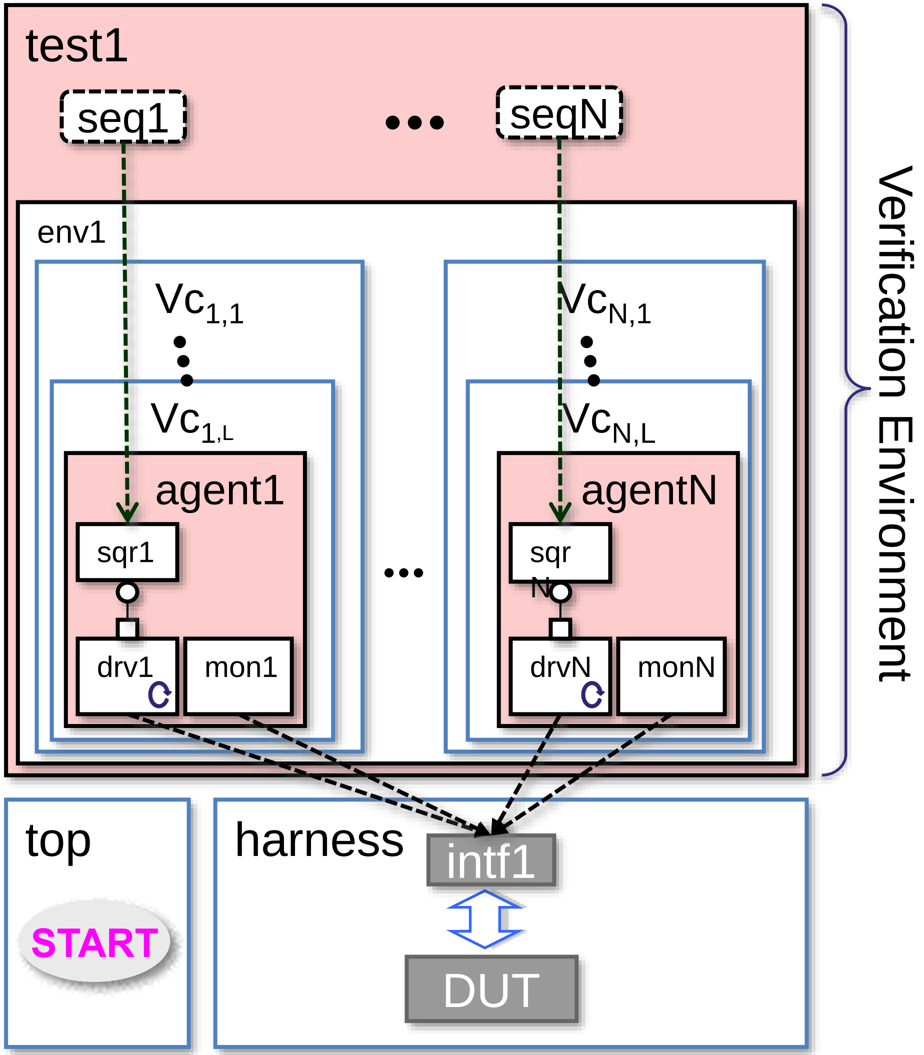
Simulation speed

Conclusions

Is UVM-1.2 Slow?

- Objection mechanism
- Re-vamped reporting mechanism?

The Test Rig



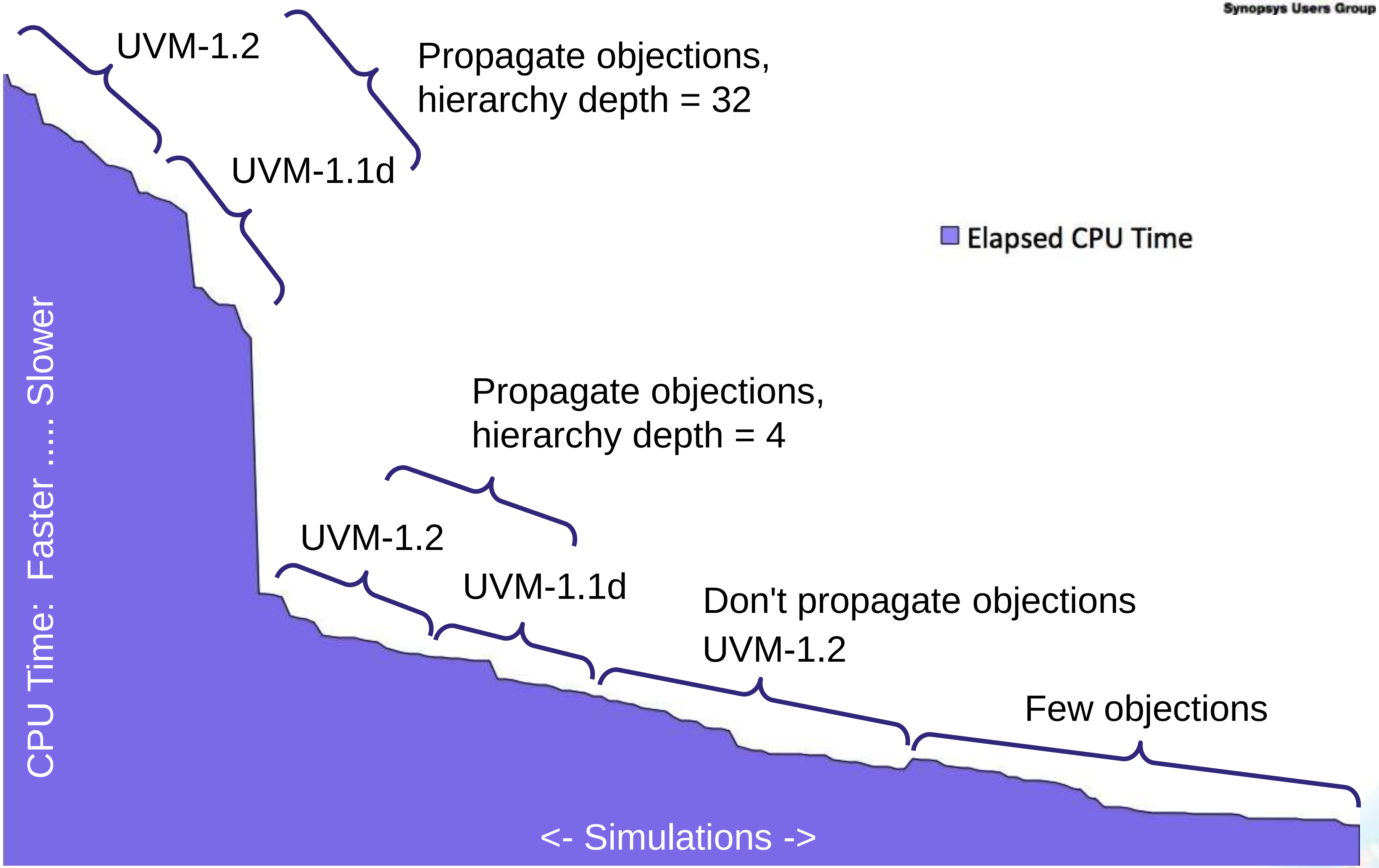
0 or 1 objection per-transaction

0 or 1 report per-transaction

4 or 32 levels of component

Trivial interface, trivial DUT

Simulation Runs Sorted by Time



UVM-1.2

Propagate objections,
hierarchy depth = 32

UVM-1.1d

Propagate objections,
hierarchy depth = 4

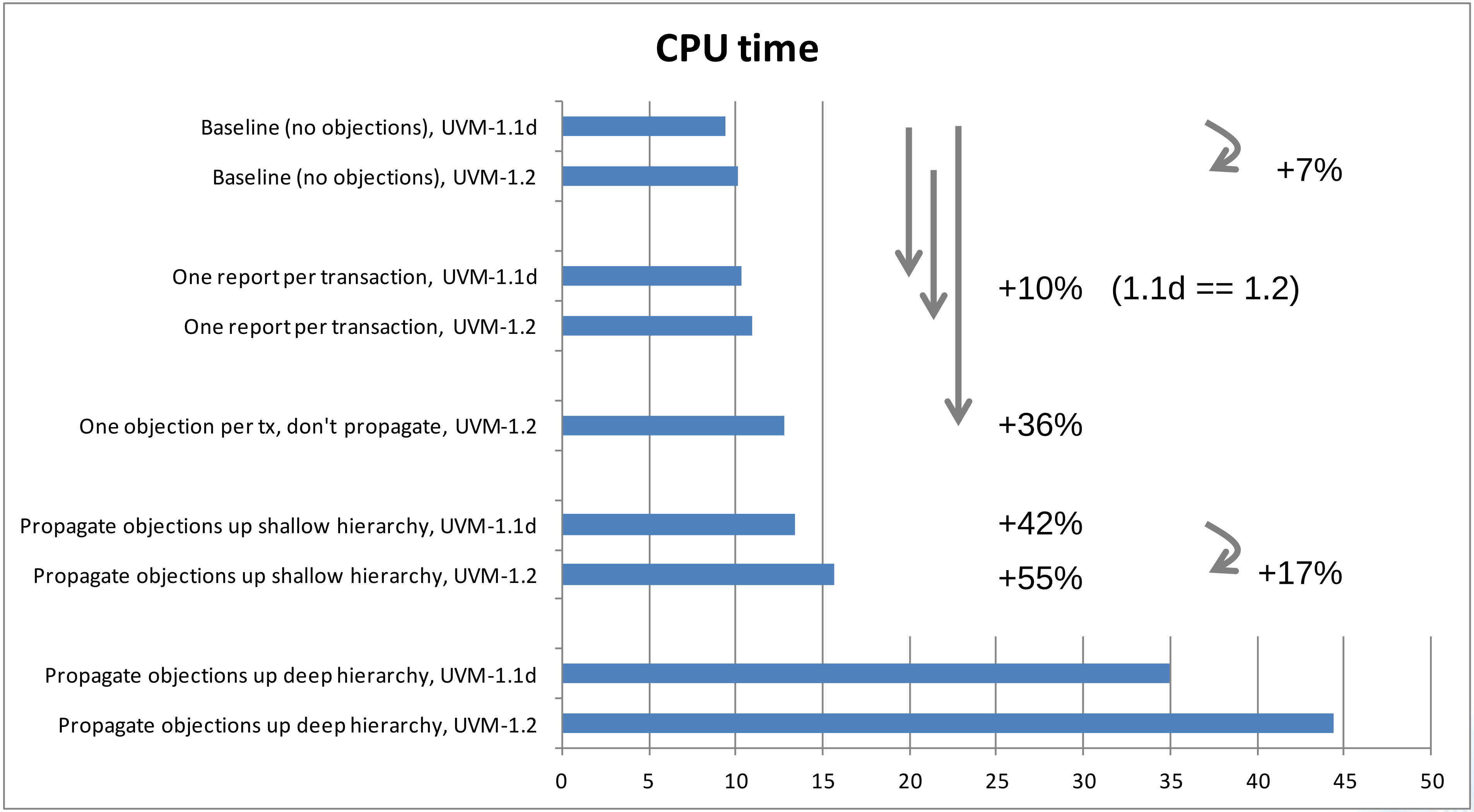
UVM-1.2

UVM-1.1d

Don't propagate objections
UVM-1.2

Few objections

Reproduced Results (VCS)



Agenda

Features

Backward compatibility and deprecation

Simulation speed

Conclusions

UVM 1.2 is Cleaner and Safer

- Locking the starting_phase data member
- Error on raising objections in non-task (function) phases
- Consistent naming of enumerations with UVM_ prefix
- Passing enumerations from command-line by name
- Deprecated features
  - get/set_config_int/string vs config_db#(int/string)::get/set
  - factory reference no longer available

Simulation Speed

- UVM-1.2 is slower than 1.1d, like-for-like
- Turning off objection propagation clearly improves speed
- No observable slow-down in reporting from 1.1d to 1.2

Navigating Your Way Toward UVM version 1.2

- Beware deprecation and incompatibility
  - Use UVM_NO_DEPRECATED and UVM_POST_VERSION_1_1
- Get educated
  - Need to properly understand the new features

DOULOS

snug 25 Synopsys Users Group

Thank You