```
import pytest
import psutil
from clusterops.cpu_exec_decorator import run_on_cpu


# Mock function to test the decorator
@run_on_cpu
def mock_function():
    return "Function executed"


def test_run_on_cpu_function_execution():
    result = mock_function()
    assert result == "Function executed"


def test_cpu_affinity_set():
    process = psutil.Process()
    process.cpu_affinity()

    # Call the decorated function
    mock_function()

    # Check if the CPU affinity was set to all available CPUs
    all_cpus = list(range(psutil.cpu_count()))
    assert process.cpu_affinity() == all_cpus
```

```
def test_cpu_affinity_not_supported(mocker):
    # Mock the cpu_affinity attribute to simulate unsupported platform
    mocker.patch.object(
        psutil.Process, "cpu_affinity", side_effect=AttributeError
    )

    # Call the decorated function and ensure it executes without error
    result = mock_function()
    assert result == "Function executed"


def test_memory_pre_allocation(mocker):
    # Mock the memory allocation to simulate a MemoryError
    mocker.patch(
        "psutil.virtual_memory",
        return_value=psutil._pslinux.virtual_memory(available=0),
    )

    with pytest.raises(
        RuntimeError, match="Failed to pre-allocate memory"
    ):
        mock_function()
```