```
import pytest

from unittest.mock import patch

from clusterops.execute_callables_parallel import (

    execute_parallel_optimized,

)




# Sample functions to be used in tests

def add(a, b):

    return a + b




def multiply(a, b):

    return a * b




def raise_exception(a, b):

    raise ValueError("Intentional error")




def test_execute_parallel_optimized_success():

    callables_with_args = [

        (add, (2, 3)),

        (multiply, (5, 4)),

    ]

    results = execute_parallel_optimized(callables_with_args)
    assert results == [5, 20]


def test_execute_parallel_optimized_with_retries():
    callables_with_args = [
        (raise_exception, (1, 2)),
        (add, (2, 3)),
    ]

    with patch(
        "time.sleep", return_value=None
    ):  # Mock sleep to speed up tests
        with pytest.raises(Exception):
            execute_parallel_optimized(callables_with_args, retries=2)


def test_execute_parallel_optimized_empty_input():
    results = execute_parallel_optimized([])
    assert results == []


def test_execute_parallel_optimized_with_custom_workers():
    callables_with_args = [
        (add, (1, 1)),
        (multiply, (2, 2)),
    ]
    results = execute_parallel_optimized(
        callables_with_args, max_workers=1
    )
    assert results == [2, 4]



def test_execute_parallel_optimized_chunk_size():
    callables_with_args = [
        (add, (1, 1)),
        (add, (2, 2)),
        (add, (3, 3)),
    ]
    results = execute_parallel_optimized(
        callables_with_args, chunk_size=1
    )
    assert results == [2, 4, 6]
```