

```

//Create Action
complex<double> get_cplx()
{
return complex<double>(13.3,-23.8);
}

HPX_PLAIN_ACTION(get_cplx, actn);
HPX_ACTION_USES_MESSAGE_COALESCING(actn); ❶

//Create instance of the actions
actn act;

vector<hpx::future<complex<double>>> vec;
vec.reserve(numparcels);

//Find the other locality
auto localities=hpx::find_remote_localities();
auto other=localities[0];

int num_repeats=4;
//Repeat num_repeats times
for (int j = 0; j < num_repeats; j++)
{
for (int i = 0; i < numparcels; ++i)
{
vec.push_back(hpx::async(act, other));
}
//Wait for all the tasks to complete
hpx::wait_all(vec);
}

```

❷