

Zadatak C – Rešenje.

Listing 1 – Paralelna implementacija *JobServiceProvider* klase.

```
public int DoCalculus(int to)
{
    Trace.WriteLine(String.Format("DoCalculus method called - interval [1,{0}]",
to.ToString()), "Information");

    // all internal endpoints of all worker role processes not including this
worker role process
    List<EndpointAddress> internalEndpoints =
RoleEnvironment.Roles[RoleEnvironment.CurrentRoleInstance.Role.Name].Instances.Where(inst
ance => instance.Id != RoleEnvironment.CurrentRoleInstance.Id).Select(process => new
EndpointAddress(String.Format("net.tcp://{0}/{1}",
process.InstanceEndpoints[internalEndpointName].IPEndpoint.ToString(),
internalEndpointName))).ToList();

    int brotherInstances = internalEndpoints.Count;
    int totalSum = 0;

    Task<int>[] tasks = new Task<int>[brotherInstances];

    for (int i = 0; i < brotherInstances; i++)
    {
        Trace.WriteLine(String.Format("Calling node at: {0}",
internalEndpoints[i].ToString()), "Information");

        int index = i;

        Task<int> calculatePartialSum = new Task<int>(() =>
        {
            IPartialJob proxy = new ChannelFactory<IPartialJob>(binding,
internalEndpoints[index]).CreateChannel();
            return proxy.DoSum(0, to);
        });
        calculatePartialSum.Start();

        tasks[index] = calculatePartialSum;
    }

    Task.WaitAll(tasks);

    foreach (Task<int> task in tasks)
    {
        totalSum += task.Result;
    }
    return totalSum;
}
```