

Original source - Sergey Zvezdin: personal blog (<http://blog.zvezdin.com/en/37>)

## Throttling in WCF

---

Here I want to give the main information which need to be known at setting of the service working with high load.

### ConcurrencyMode

Parameter which defines a service operating mode - how many real threads will process messages and to execute actions. This parameter can have following values:

- Single – Messages are processed sequentially in a single thread (default value).
- Reentrant – Similarly Single with some singularity (see [MSDN](#)).
- Multiple – Messages are processed in parallel in several threads.

MaxConcurrentCalls parameter which limits a maximum quantity of threads of processing of competing calls is refered with this parameter. Usually ConcurrencyMode parameter is set at service definition.

```
01 [ServiceBehavior(ConcurrencyMode = ConcurrencyMode.Multiple)]
02 public class Service1 : IService1
03 {
04     public string GetData(int value)
05     {
06         return string.Format("You entered: {0}", value);
07     }
08
09     public CompositeType GetDataUsingDataContract(CompositeType composite)
10     {
11         if (composite.BoolValue)
12         {
13             composite.StringValue += "Suffix";
14         }
15         return composite;
16     }
17 }
```

### InstanceContextMode

Very important parameter which defines in what way is created object on the service side for query processing. The parameter can have following values:

- PerCall – Creates a new instance of service object on each call (default value).
- PerSession – Creates a new instance of service object on each session.
- Single – Creates the single instance of service object which will proceed all queries.

It is clear that more effectively to use Single mode. In this case the single object of service will be created and it is not necessary to care of use of memory and other resources. However, in cases when service contains a state to use mode Single it is impossible. In this case it is possible to limit quantity of simultaneously existing instances of service in [MaxConcurrentInstances](#) parameter. Also it is possible to limit quantity of simultaneously working sessions, it becomes by means of [MaxConcurrentSessions](#) parameter.

Usually InstanceContextMode parameter is set at service definition.

```
01 [ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]
02 public class Service1 : IService1
03 {
04     public string GetData(int value)
05     {
06         return string.Format("You entered: {0}", value);
07     }
08
09     public CompositeType GetDataUsingDataContract(CompositeType composite)
10     {
11         if (composite.BoolValue)
12         {
13             composite.StringValue += "Suffix";
14         }
15         return composite;
16     }
17 }
```

Besides, on properties of stability of the application on external attacks a quotas can depends various parameters. For example, sendTimeout, receiveTimeout, maxReceivedMessageSize etc. More information about it you can read at [MSDN](#).