# Enduro/X Core - Feature #10

## Server's MIN/MAX using for additional spawning at high load

11/27/2015 11:24 AM - Madars

| | | | |
|---|---|---|---|
| **Status:** | New | **Start date:** | 11/27/2015 |
| **Priority:** | Normal (Code 4) | **Due date:** | |
| **Assignee:** | Madars | **% Done:** | 0% |
| **Category:** | | **Estimated time:** | 0.00 hour |
| **Target version:** | | | |

**Description**

Enduro/X server binaries automatic spawning (till max) at high low. And instance stopping still min, when there is no load

### History

#### #1 - 05/23/2018 12:12 PM - Madars

we could monitor following items:

1. Speed of the queue growing (delta items per cycle) - response speed
2. Processing speed of the service (average?)
3. Number of items in queue
4. Total number of requests processed by services.

This all infos shall be recorded for each of the service for some time period. The descision shall be made to boot some additional instance or stop it.

If average number of items in queue is 0, then instances could be shut down.

#### #2 - 05/23/2018 12:37 PM - Madars

Implement in version 5.4

#### #3 - 05/24/2018 11:40 AM - Madars

```
/* deleta messages per period in queue, to start act */
CONFIG_DELTA_Q  5

/* search for best option */
CONFIG_TIME_RECONFIG  1m

/* if system is clean */
CONFIG_TIME_DOWN  30m
```

Dynamic params:

```
/* delta messages in Q in CONFIG_TIME_RECONFIG  */
deltaq / prev_deltaq

/* delta messages processed in CONFIG_TIME_RECONFIG  */
deltaproc / prev_deltaproc

/* total number of messages in queue */
msgsq

if (deltaq > CONFIG && prev_deltaq < deltaq )
{

    if (deltaproc > prev_deltaproc)
```

```
    {
        vote + 1;
    }
    else
    {
        vote -1;
    }
}
else if (deltaq > CONFIG)
{
    vote + 1;
}
else if (msgsq == 0)
{
    vote -1;
}
```

#### #4 - 05/24/2018 11:47 AM - Madars

rules shall be written at service level.
Also accumulation shall be done from shared mem for all services and summed for each individual service.
The order of which binary will be started up is not specified by system (if multiples exes cross advertise the services).

For doing shutdown, we iterate all binaries, search for first binary which are in group of marked for shutdown, and all other services says OK for shutdown, then shutdown on that binary. And mark it as action completed.

#### #5 - 05/25/2018 08:01 AM - Madars

Other option would me measure the proportion of the queued messages and the number of service instances.

and config params would say:

CONFIG_QRATIO_UP 1.5
CONFIG_QRATIO_DOWN 1
CONFIG_QRATIO_CORRECTIONAL_DELTA 0.5 //new - ratio difference between previous action and now, e.g. have grown by 0.5
CONFIG_MSGSPROC_CORRECTIONAL_DELTA -10 //the number of messages processed before previous action and now. if it is lower than this number, then perform shutdown
CONFIG_UP_TIME 5m
CONFIG_DOWN_TIME 30m
CONFIG_DOWN_CORRECTIONAL 2m //have grown by 0.5

Say 10 messages in Q for 5 servers would give us 10/5 = 2, this would mean that we need to boot additional instance.

The timing parameters would say:
- how long to stay (with no changes) in above ratio, to boot additional instance
- how long to stay (with no changes) in bellow ratio, to stop additional instance

Needs to think when to detect that situation did worsen at the process boot up???
We could check number of processed messages for each instance. if number becomes lower and ratio keeps growing after additional instance boot, then we shall stop some instance. This could additional time setting - how long to measure correctional shutdown, for example if 2 minutes number of

messages processed by instance is lower than previous value actions value and ratio is greater than previous action's ratio, then we perform correctional shutdown of the item.

For shared mem stats min/max/last and messages OK, Failed we need to write in shared mem some two bytes of checksum, so that we know when values are good.
We do not want to lock the SHM for each statistics update due to performance reasons.