

## Configuration:

Following services/triggers are configured:

**parent** (Calculate SLA checked; serviceid = 1; Problem, if at least one child has a problem)

└─ **child1** (Calculate SLA; serviceid = 2; linked to **trigger1**, severity = average)

└─ **child2** (Calculate SLA; serviceid = 3; linked to **trigger2**, severity = high)

## Steps to reproduce:

1. First, move **trigger1** into problem state.

```
mysql> select * from service_alarms;
+-----+-----+-----+-----+
| servicealarmid | serviceid | clock      | value |
+-----+-----+-----+-----+
|          1 |          2 | 1584562516 | 3 |
|          2 |          1 | 1584562516 | 3 |
+-----+-----+-----+-----+
```

Having such records SLA problem time for last 7 days is calculated as:  
{CURRENT\_TIMESTAMP} - 1584562516

2. Now move **trigger2** into problem state with future time (+7200s) so it will have negative problem duration.

```
mysql> select * from service_alarms;
+-----+-----+-----+-----+
| servicealarmid | serviceid | clock      | value |
+-----+-----+-----+-----+
|          1 |          2 | 1584562516 | 3 |
|          2 |          1 | 1584562516 | 3 |
|          3 |          3 | 1584569716 | 4 |
|          4 |          1 | 1584569716 | 4 |
+-----+-----+-----+-----+
```

Having such records SLA problem time for last 7 days is calculated as follow:

- **parent** and **child1**: {CURRENT\_TIMESTAMP} - 1584562516 (future events are ignored because "last 7 days" ends now).
- **child2**: 0 seconds problem time because it has no events before {CURRENT\_TIMESTAMP} (end of last 7 days).

Pay attention that by servicealarmid = 4 **parent** service severity was increased from average to high (value column) because higher severity subservice was enforced.

3. Now move **trigger1** to OK state so **child1** stops problem time counting.

```
mysql> select * from service_alarms;
```

servicealarmid	serviceid	clock	value
1	2	1584562516	3
2	1	1584562516	3
3	3	1584569716	4
4	1	1584569716	4
5	2	1584566223	0

Now look what happened:

- **parent** service didn't changed its state because **trigger1 (child1)** has lower severity than it's current value (**correct**).
- **child2** (serviceid=3) still has 0 seconds problem time because negative current duration gaps are ignored (**correct**).
- **child1** is back in OK state with fixed problem time (1584566223 - 1584562516) (**correct**).
- **parent** service problem time continues counting because (**wrong**):
  - it was not stopped when **child1** went to OK state;
  - there is no reliable way to detect that **parent** counting was turned on by child service with negative duration.

### Proposed solution:

To fix it, at moment when last non-negative subservice (here **child1**) goes into OK state:

1. one more record in `service\_alarms` table should be entered (servicealarmid = 6):
  - a. **parent**'s serviceid;
  - b. stop time (same as for child1);
  - c. value 0.
2. parent's record entered at moment when negative problem was recorded must be moved to the end, to have greater servicealarmid (in given example servicealarmid = 4 increased to be servicealarmid = 7).

servicealarmid	serviceid	clock	value
1	2	1584562516	3
2	1	1584562516	3
3	3	1584569716	4
4	1	1584569716	4
5	2	1584566223	0
6	1	1584566223	0
7	1	1584569716	4

Parent service record in `services` table should be left as is (in problem state). IDS must be increased.