GlideinWMS - Feature #2531

Monitoring for frontend: store the number of Job restarts

03/05/2012 03:42 PM - Parag Mhashilkar

Status:	Assigned	**Start date:**	03/05/2012
Priority:	Normal	**Due date:**	
Assignee:	Dennis Box	**% Done:**	0%
Category:	Frontend Monitoring	**Estimated time:**	24.00 hours
Target version:	v3_6_6	**Spent time:**	0.00 hour
Stakeholders:	FIFE		

Description

- Update the rrd
- But need to handle the frontend upgrades breaking
- In general how to handle upgrades when rrd' change

History

#1 - 03/05/2012 03:43 PM - Parag Mhashilkar

- Target version set to v2_5_6

#2 - 03/12/2012 01:56 PM - Parag Mhashilkar

- Target version changed from v2_5_6 to v2_5_7

#3 - 04/02/2012 03:54 PM - Burt Holzman

- Target version changed from v2_5_7 to v2_7_x

#4 - 05/07/2012 03:16 PM - Burt Holzman

- Subject changed from Monitoring for frontend store the Job restart to Monitoring for frontend: store the number of Job restarts

#5 - 08/07/2012 04:28 PM - Parag Mhashilkar

Proposal

The goal here should be to identify easily if there are problems running job(s).

Data Collection

We do not store per job information, so essentially, this boils down to either extending existing rrds or creating new ones to store information on how many jobs in the queue have been restarted and how many times.

So essentially, we store info like # jobs restarted

1) 2 < restarts <= 5
2) 5 < restarts <= 10
3) restarts > 10

Its tough to come up with good intervals that would apply to everyone, but the above seem more reasonable to me.

Monitoring

We need to either create a new page or super impose the info on top of existing plots.

Possible options:

- Average number of restarts
  If only a handful of jobs restart several times, this won't give a meaningful picture.

- Jobs in queue that restarted several times [2, 5, 10, 10+]

If the jobs are long running, this will show problem till the jobs are done. We need to show ratio of jobs that are suffering from restarts to total jobs

Anything else? Suggestions welcome.

#6 - 08/07/2012 05:05 PM - Igor Sfiligoi

A couple comments:
1)
We likely want to distinguish jobs who have never started from the ones who started at least once.

2)
The average (re)start number is likely useful, if applied only to jobs that were started at least once.
I think it gives a nice one-value to gauge the health of the system.

3)
As for the intervals, I think we need at least

- never restarted
- restarted once
- restarted twice
- restarted a few times (2<N<=X1)
- restarted a few more times (X1<N<=X2)
- restarted a lot (N>X2)

The first 3 are obvious, for the last three I think we will have to pick two arbitrary numbers.
5 and 10 as proposed above are OK with me.

And I would only apply it to the jobs in (1).

4)
I am not too worried about the Web pages...
we can chage that as many times as needed to get it right.
But the RRDs are difficult to evolve, so we should think it through before releasing it in the wild.

Thanks,
Igor

#7 - 08/08/2012 10:18 AM - Parag Mhashilkar

Just considering RRD design first.

Data Collection (RRD Intervals)

RRD Field will be populated from the NumJobStarts in classad. NumRestarts in the classad seems to apply to checkpointed jobs only and may not be useful.

- N = 0 (never started)
- N = 1 (started once)
- N = 2 (restarted once)
- N = 3 (restarted twice)
- 3 < N <= 5 (restarted few times; restarted 3 to 4 times)
- 5 < N <= 10 (restarted few more times; restarted 5 to 9 times)
- 10 < N (restarted lot of times, restarted 10+ times)

Data collected in these fields would be straight up number of jobs in the queue that have started N times. This will let us apply any rules we want for displaying the required info.

Ideally we want to get it right with first release but number of starts is relative to the user jobs. If needed we can add new fields in RRD without any upgrade issue. We can put #2667 to test :)

Am I missing anything?

#8 - 08/09/2012 12:42 AM - Igor Sfiligoi

Almost.

Your proposal does not distinguish between jobs that are currently running for the first time, and jobs that are idle, but were preempted once.

Maybe we are trying to measure the wrong thing?
Now that I think of it, the health of the system is measured in number of preemptions, not number of starts.

What do you think?

#9 - 08/09/2012 10:22 AM - Parag Mhashilkar

Igor Sfiligoi wrote:

> Almost.
>
> Your proposal does not distinguish between jobs that are currently running for the first time, and jobs that are idle, but were preempted once.

If we need to distinguish NumJobStarts for running and idle jobs, we can store the above info for both, running and idle jobs.

> Maybe we are trying to measure the wrong thing?
> Now that I think of it, the health of the system is measured in number of preemptions, not number of starts.
>
> What do you think?

But doesn't NumJobStarts directly relate to number of preemptions? NumJobStarts won't increase unless the jobs has been restarted. Now the question is how do we distinguish between jobs that have been preempted from the jobs that keep on flip-flopping between idle-running because of other errors.

#10 - 08/09/2012 10:50 AM - Parag Mhashilkar

Found following classad attribute(s). Maybe we can make use of them?

LastVacateTime: Time at which the job was last evicted from a remote workstation. Measured in the number of seconds since the epoch (00:00:00 UTC, Jan 1, 1970)

#11 - 08/09/2012 04:35 PM - Igor Sfiligoi

My comment was not that it was difficult to get the number, but you did not seem to store it.

I.e. please provide a proposal on how you want to store this information.

Thanks,
Igor

#12 - 08/10/2012 09:54 AM - Parag Mhashilkar

I did mention we can store the job numbers for both idle and running jobs in my comment. But just to spell it out, here are the fields

- Idle_JobsStart_0
- Idle_JobsStart_1
- Idle_JobsStart_2
- Idle_JobsStart_3
- Idle_JobsStart_4to5
- Idle_JobsStart_6to10
- Idle_JobsStart_10plus
- Running_JobsStart_0
- Running_JobsStart_1
- Running_JobsStart_2
- Running_JobsStart_3
- Running_JobsStart_4to5
- Running_JobsStart_6to10
- Running_JobsStart_10plus

Going back to my unanswered question. Can we use LastVacateTime to distinguish between jobs that have been preempted from the jobs that keep on flip-flopping between idle-running because of other errors?

#13 - 08/10/2012 12:51 PM - Parag Mhashilkar

Just to add Idle_JobsStart_0 is same as Running_JobsStart_0 only Idle part is needed. Running_JobsStart_0 is meaningless.

#14 - 08/13/2012 12:00 PM - Igor Sfiligoi

Yes, LastVacateTime is a good attribute to use.

As for the attribute name, while conceptually OK, they are not consistent with the rest of the attributes of this kind in the RRDs.
This would be consistent:

- ..._JobsStart_0
- ..._JobsStart_1
- ..._JobsStart_2
- ..._JobsStart_3
- ..._JobsStart_4
- ..._JobsStart_8
- ..._JobsStart_Many

Same semantics as above.

#15 - 01/09/2013 01:18 PM - Igor Sfiligoi

Was there any progress on this?

#16 - 01/10/2013 12:22 PM - Parag Mhashilkar

Sorry no progress. Other high priority stuff always get in front of the queue.

#17 - 06/24/2013 03:36 PM - Parag Mhashilkar

- Target version changed from v2_7_x to v3_x

#18 - 10/18/2016 03:30 PM - Parag Mhashilkar

- Assignee changed from Parag Mhashilkar to Dennis Box

- Target version changed from v3_x to v3_2_17

Be careful not to break anything

#19 - 12/14/2016 11:24 AM - Parag Mhashilkar

- Target version changed from v3_2_17 to v3_2_18

#20 - 02/15/2017 11:28 AM - Marco Mambelli

- Target version changed from v3_2_18 to v3_2_19

#21 - 05/24/2017 10:19 AM - Marco Mambelli

- Target version changed from v3_2_19 to v3_2_20

#22 - 07/24/2017 10:38 AM - Marco Mambelli

- Target version changed from v3_2_20 to v3_2_21

#23 - 11/15/2017 04:42 PM - Marco Mambelli

- Target version changed from v3_2_21 to v3_2_22

#24 - 03/05/2018 03:25 PM - Marco Mambelli

- Stakeholders updated

#25 - 03/07/2018 04:12 PM - Dennis Box

- Estimated time set to 24.00 h

#26 - 03/21/2018 11:45 AM - Marco Mambelli

- Target version changed from v3_2_22 to v3_2_23

#27 - 04/26/2018 04:32 PM - Marco Mambelli

- Target version changed from v3_2_23 to v3_4_0

#28 - 05/25/2018 11:54 AM - Marco Mambelli

- Target version changed from v3_4_0 to v3_4_1

#29 - 09/19/2018 10:43 AM - Dennis Box

- Target version changed from v3_4_1 to v3_5

#30 - 12/05/2018 07:36 PM - Dennis Box

- Target version changed from v3_5 to v3_5_1

#31 - 08/28/2019 10:26 AM - Marco Mambelli

- Target version changed from v3_5_1 to v3_6_1

#32 - 11/01/2019 03:47 PM - Dennis Box

- Target version changed from v3_6_1 to v3_6_2

#33 - 01/08/2020 10:23 AM - Dennis Box

- Target version changed from v3_6_2 to v3_6_3

#34 - 07/15/2020 10:19 AM - Dennis Box

- Target version changed from v3_6_3 to v3_6_4

#35 - 09/17/2020 02:14 PM - Marco Mambelli

- Target version changed from v3_6_4 to v3_6_5

#36 - 10/04/2020 01:48 AM - Marco Mambelli

- Target version changed from v3_6_5 to v3_6_6