Tvheadend - Feature #3642

Add more usable stats for recording quality

2016-03-18 07:52 - Dietmar Konermann

Status:	New	**Start date:**	2016-03-18
Priority:	Normal	**Due date:**	
Assignee:	Adam Sutton	**% Done:**	0%
Category:	PVR / DVR	**Estimated time:**	0.00 hour
Target version:	4.6		

Description

I find the curent absolute error and data error counters not being really usable to determine if a recording is usable. Instead a relative metric would be more usable, e.g. "% of recorded minutes with errors".

Currently the stats of a recording with some errors per minute look better than a recording with just a single black-out minute generating thousands of errors.

The metric suggested above would better match what the user experience when trying to watch. Of course it would be nice to have this metric trigger mark recordings as failed on a configurable threshold, triggering re-recordings.

Thanks in advance
Dietmar.

History

#1 - 2016-03-18 09:42 - Jaroslav Kysela

- Target version set to 4.4

#2 - 2016-03-18 19:22 - Dietmar Konermann

Hi Jaroslav,

thanks for considering this. My naive approach would be simple... during a given recording check every minute if any error counter during the last minute changed. If yes, increase the "faulty minute count" and update the % stats accordingly. Sounds simple. wink.png

Thanks again
Dietmar.

#3 - 2016-03-18 20:14 - Jaroslav Kysela

I understand the requirement. I'll think about this more, but the absolute errors makes also sense if you get occasional dropouts. I had a complete time-log in my head to see which parts of recordings are broken.

#4 - 2016-06-04 01:25 - Steve Black

Glad I searched before adding my own much-wordier request!

I would suggest that this would be highly useful for internal recording/re-recording management as well as for passing to post-processing. (That's probably obvious, but given how often I miss the obvious myself... wink.png)

#5 - 2017-11-28 17:13 - Dietmar Konermann

Again had some unusable recordings and noticed them too late. Thought it would be ok to ping this request again... although being flagged for 4.4. wink.png
I would really appreciate the simple relative metric.

Cheers
Diddle.

#6 - 2017-11-28 19:10 - Em Smith

In case it's useful for comparison, MythTV logs this at the end of a recording, and persists the xml to the DB for the client to use.

```
FinishedRecording(CHANNEL_YYYY-MM-DDTHH:MM:SSZ) damaged recq:
```

```
<RecordingQuality overall_score="0" key="<filename>" countinuity_error_count="3" packet_count="84856">
```

```
   <Gap start="YYYY-MM-DDTHH:MM:SSZ" end="YYYY-MM-DDTHH:MM:SSZ" duration="6431" />

   <Gap start="YYYY-MM-DDTHH:MM:SSZ" end="YYYY-MM-DDTHH:MM:SSZ" duration="999" />

</RecordingQuality>
```

And another example for a different recording:

```
damaged recq:
<RecordingQuality overall_score="0.9" key="<filename>" countinuity_error_count="277" packet_count="2288569" />
```

I suspect the overall_score percentage (however it's calculated) is the useful stat for differentiating long recordings vs short ones, and the Gaps for determining the time log for how bad the problems were.

Of course, xml was last decade so we'd use json.

#7 - 2017-12-01 15:03 - Em Smith

This was the MythTV (GPLv2+) commit that brought in its quality tracker (with a few bug fixes later) with the main logic in the recordingquality.cpp

[[https://github.com/MythTV/mythtv/commit/ca0419d8969b97bba80c590e5fed02f7c1d948c8]]

The "interesting" thing (according to the comments) is that the quality is weighted so if there are problems at the start/end of a programme they are given higher priority, on the assumption they contain important information such as "Who Shot J.R.?"

The quality score is degraded if the total number of continuity errors are >0.1% of the total recording, and set far lower if the errors are >1% of the total recording.

The score_gaps function seems to also modify the score based on the lengths of gaps (for example a five second gap is worse than a 0.1 second gap, but not as bad as a two minute gap). So, my #4747 where Tvheadend only recorded for one minute but had the recording marked as success, would be detected by this algorithm.

At a quality score of <0.95 the programme is automatically flagged as damaged and re-recorded.

MythTV also has a RecStatus::Failing so that it can schedule on the +1 timeshift channel if the current recording won't be good, but this seems to only be used for the initial tuner lock rather than checking quality during the recording. But I think an ongoing check is good since many programmes are an hour and with a padding it means they end after the +1 timeshift would've started (e.g., recording 9pm--10:02pm, but if it's a bad recording then it should re-record on the 10pm--11pm timeshift channel).

I don't know enough about the logic of either tvh streaming or mythtv to implement this one.

#8 - 2017-12-03 19:05 - Em Smith

- File tvh_find_corrupt_rec added

In case it's useful, I created a script to do basic checking of files to determine if they are heavily corrupted.

It does this by simply checking the length of the recording against the expected duration. If it's within a few seconds the file is considered ok; if it is within a few minutes then it's marked as failed and re-record; if it's really wrong then it will be also be deleted after a day. The program outputs a script to do the updates and will do no updates unless you execute the generated output file.

The script accesses internals it really shouldn't access, and isn't very user-friendly, but it has identified a large number of files I'd consider corrupt, such as files that were <100K, or hour-long shows that were over 20 minutes too short.

Usage:

```
./tvh_find_corrupt_rec --output-script curl.sh /storage/tvheadend/movie/file.ts
```

The will check the file.ts and output a curl script to curl.sh.

To the screen it will output a lot of debug. The interesting line is where it will tell you the duration expected and the actual duration of the file. So this is saying that the show is 20 minutes too short:

```
Corrupt file: Bad duration 614.40 (00:10:14) expect 1860.00 (00:31:00) cutoff 1841.4, total delta 1245.60 (00:
20:45) for /path/file.ts
```

The curl.sh will contain lines such as:

```
curl http://localhost:9981/api/dvr/entry/move/failed --data-urlencode 'uuid=["abcdef01234567890"]'
```

Or for heavily corrupted files it would output:

```
...'node={"removal":1,"norerecord":0,"retention":1,"noresched":0,"uuid":"abcdef01234567890"}'
```

There are a few options such as --user, --password, --host, --port.

Multiple filenames can be given on the command line. They must be the full pathnames as recorded in the dvr/log. Alternatively a pipeline can be used such as:

```
find /net/nas/tvheadend/movies | ./tvh_find_corrupt_rec --output-script curl.sh && sh -v curl.sh
```

It appears to work, but caveat emptor.

#9 - 2019-10-25 08:40 - Joe User

Another "bug" which should really be a "feature request".

#10 - 2019-10-25 11:12 - Mark Clarkstone

- Tracker changed from Bug to Feature

- Found in version deleted (4.1)

Joe User wrote:

> Another "bug" which should really be a "feature request".

I agree smile.png

#11 - 2021-02-20 22:32 - Flole Systems

- Target version changed from 4.4 to 4.6

Files

tvh_find_corrupt_rec	15 KB	2017-12-03	Em Smith