StoRM: recent problems with ATLAS

End of Aug - Beginning of Sep: very frequent crashes of storm-frontend (ATLAS ticket). When it crashes, the FE is serving around 20000 synchronous requests per minute (plus 2000 async requests), i.e. more than 350 Hz rate, not on the same token. However, ATLAS claims their workflow is not changed, and these are transfers accumulated in the period 27.08- 01.09.

- We increased the resources of the FE (2 to 4 CPUs) and the frequency of restarts, flanking sensu with **check_daemons.py**
- We created a **grep_log.py** script to understand the requests to the FE; many of these requests were statusPtP and statusPtG (served directly by the FE and not passed to the BE).
- With the help of Lorenzo, we managed to reproduce the problem: 300 hundreds jobs, each one 1000 status request requests in parallel -> crash. Hereafter this is called "*Rinaldi crash test*".

StoRM developers stepped in

Sep 10th: a new package, with detailed monitoring for the frontend, is installed on storm-atlas. During the weekend, no problems. On Monday, Rinaldi crash test -> crash.

Sep 11th: a new package, with a configurable limit on the size of the queue of the requests accepted by the FE, is installed. *fe.threadpool.maxpending=300* Rinaldi crash test - > no crash, but the requests are not processed (even worse).

Sep 14th: new packages for both FE and BE were installed. Rinaldi crash test -> crash, but we had to wait longer for it (starting to be optimistic)

Sep 17th: FE configuration modified using a lower number of threads. Rinaldi crash tests -> NO crash!

The improved FE and BE

- A new configurable limit on the queue of requests for the FE is introduced
- The logic of XMLRPC calls between FE and BE is changed: each thread gets a dedicated XMLRPC client
- The configuration of BE threadpool is changed, *maxthreads* is no longer ignored
- On the BE, the queue for XMLRPC requests is now limited
- An endpoint to expose backend metrics via http is introduced

Working configuration for the FE (very very empiric):

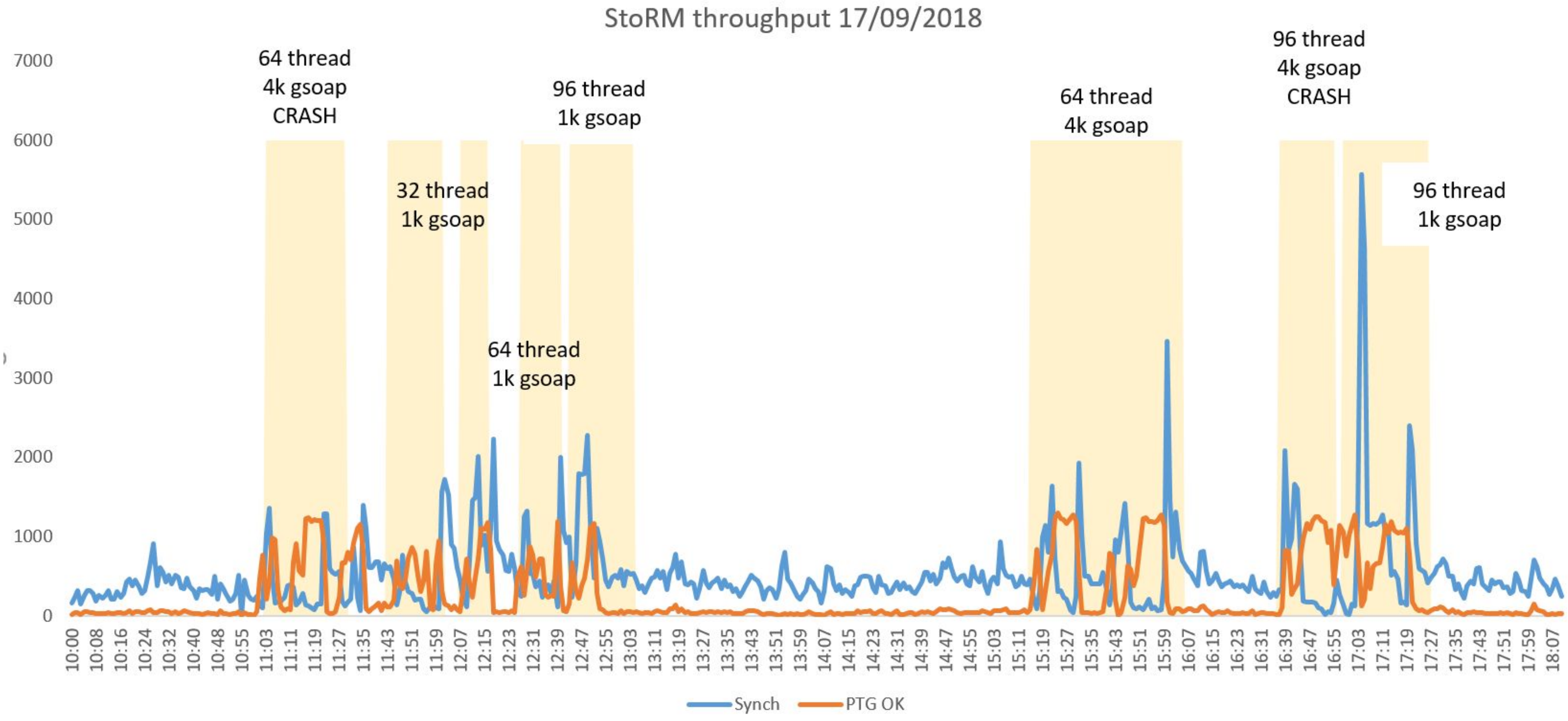
```
fe.threadpool.threads.number=64   # previously much higher to contain FE-BE communication inefficiencies
fe.gsoap.maxpending=1000          # previously much higher, still not understood why we need to keep it low
fe.threadpool.maxpending=1000     # limit on the request queue size, previously ignored
```

What happens: /var/log/storm/monitoring.log

```
09/17 17:15:47 : [#   18 lifetime=00:18:00] S [OK:25993,F:920,E:0,m:0.000,M:3.521,Avg:0.186] A [OK:11335,F:0,E:0,m:0.006,M:0.771,Avg:0.026]
Last:(S [OK:1097,F:4,E:0,m:0.000,M:1.112] A [OK:12,F:0,E:0,m:0.007,M:0.132]) Tasks(max_active:64,active:64,max_pending:1000,pending:1000)
09/17 17:16:47 : [#   19 lifetime=00:19:00] S [OK:26933,F:927,E:0,m:0.000,M:3.521,Avg:0.180] A [OK:11347,F:0,E:0,m:0.006,M:0.771,Avg:0.026]
Last:(S [OK:940,F:7,E:0,m:0.000,M:0.482] A [OK:12,F:0,E:0,m:0.007,M:0.177]) Tasks(max_active:64,active:64,max_pending:1000,pending:789)
09/17 17:17:47 : [#   20 lifetime=00:20:00] S [OK:27793,F:934,E:0,m:0.000,M:3.521,Avg:0.175] A [OK:11363,F:0,E:0,m:0.006,M:0.771,Avg:0.026]
Last:(S [OK:860,F:7,E:0,m:0.000,M:0.538] A [OK:16,F:0,E:0,m:0.007,M:0.120]) Tasks(max_active:64,active:64,max_pending:1000,pending:556)
09/17 17:18:47 : [#   21 lifetime=00:21:00] S [OK:28625,F:947,E:0,m:0.000,M:3.521,Avg:0.170] A [OK:11371,F:0,E:0,m:0.006,M:0.771,Avg:0.026]
Last:(S [OK:832,F:13,E:0,m:0.000,M:0.160] A [OK:8,F:0,E:0,m:0.007,M:0.013]) Tasks(max_active:64,active:64,max_pending:1000,pending:474)
09/17 17:19:47 : [#   22 lifetime=00:22:00] S [OK:29511,F:964,E:0,m:0.000,M:3.521,Avg:0.165] A [OK:11390,F:0,E:0,m:0.006,M:0.771,Avg:0.026]
Last:(S [OK:886,F:17,E:0,m:0.000,M:1.036] A [OK:19,F:0,E:0,m:0.007,M:0.015]) Tasks(max_active:64,active:64,max_pending:1000,pending:171)
09/17 17:20:47 : [#   23 lifetime=00:23:00] S [OK:31871,F:1139,E:0,m:0.000,M:3.521,Avg:0.157] A [OK:11651,F:0,E:0,m:0.006,M:0.771,Avg:0.025]
Last:(S [OK:2360,F:175,E:0,m:0.000,M:1.925] A [OK:261,F:0,E:0,m:0.007,M:0.317]) Tasks(max_active:64,active:14,max_pending:1000,pending:0)
```

What happens: /var/log/messages

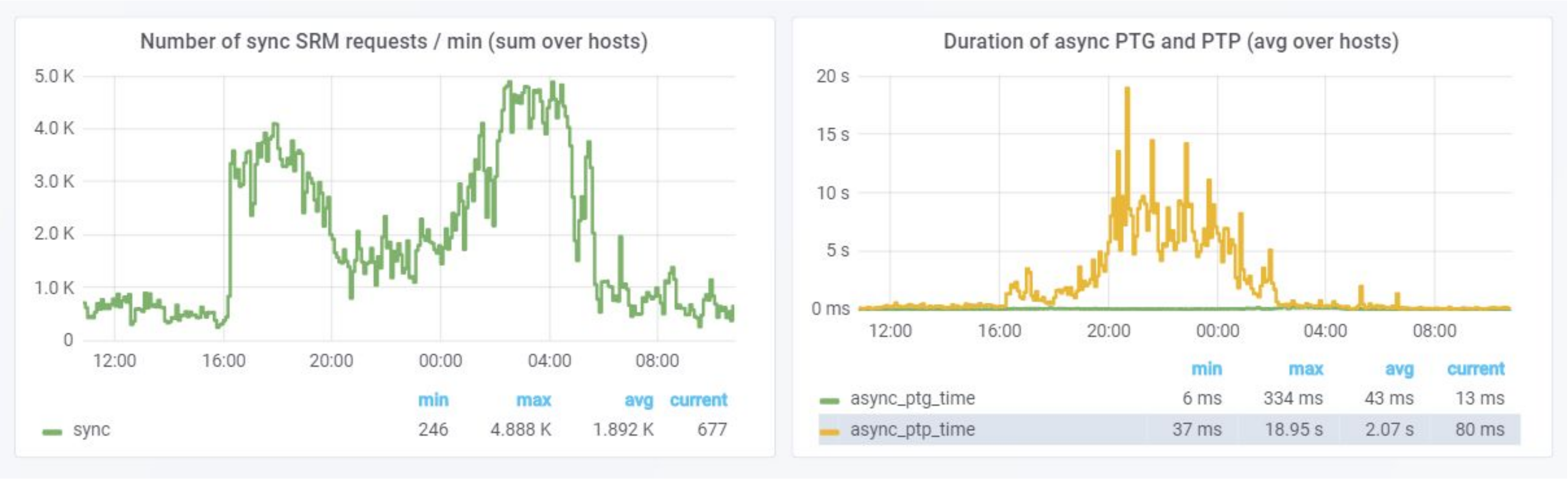
```
Sep 17 17:05:58 storm-fe-atlas-07 kernel: TCPv6: Possible SYN flooding on port 8444. Dropping request.
```



200 threads, no matter the configuration -> crash.

The effect of *fe.gsoap.maxpending* is still not clear.

Last night



	min	max	avg	current
🟩 sync	246	4.888 K	1.892 K	677

	min	max	avg	current
🟩 async_ptg_time	6 ms	334 ms	43 ms	13 ms
🟨 async_ptp_time	37 ms	18.95 s	2.07 s	80 ms

Apart from storm-atlas, StoRM 1.1.14 everywhere

- On Sep 6th we completed the upgrade to the latest version of StoRM on all our storm and gridftp endpoints, following http://italiangrid.github.io/storm/documentation/sysadmin-guide/1.11.14/#upgrading
  - storm-backend-server-1.11.14-1.
  - storm-frontend-server-1.8.11-1
  - storm-globus-gridftp-server-1.2.1-1
  - storm-native-libs-java-1.0.5-2, storm-native-libs-lcmaps-1.0.5-2, storm-native-libs-gpfs-1.0.5-2
  - storm-xmlrpc-c-client-1.39.12-1
  - yaim-storm-4.3.11-1
  - storm-webdav-1.0.5-1
- found problems with the latest jdk on storm-webdav, which are solved downgrading to 1.8.0_25
- Just one problem, with recall not working (CMS ticket), due to a missed upgrade of the GEMMS version installed on the HSM; fixed with GEMSS 1.7.4-1

Thank you

… and big thanks to Andrea and Francesco for their time and help!