ORP-2.3.3 Ballooning issue

We have a case where the paired-end transcriptome is both hangs and also balloons space on the HPC to where my partition ends up being completely filled up.

- In the instance here it does not fail the run. Of note it does sometimes fail the run.

Here are the SRA files used for this assembly -
SRR11659659_suffixes_1.fastq.gz - 2.8 GB
SRR11659659_suffixes_2.fastq.gz- 2.9 GB

Doing a du -lh on of the fasta outputs in a CBins is high for a few hung up runs listed below.

```
1.9M
./read_partitions/Fb_0/CBin_15/c1594.trinity.reads.fa.out/chrysalis/Component_bins/C
bin0
1.9M
./read_partitions/Fb_0/CBin_15/c1594.trinity.reads.fa.out/chrysalis/Component_bins
14M     ./read_partitions/Fb_0/CBin_15/c1594.trinity.reads.fa.out/chrysalis
0       ./read_partitions/Fb_0/CBin_15/c1594.trinity.reads.fa.out/__polish.chkpts
100K
./read_partitions/Fb_0/CBin_15/c1594.trinity.reads.fa.out/Trinity.tmp.fasta.salmon.i
dx
2.3T    ./read_partitions/Fb_0/CBin_15/c1594.trinity.reads.fa.out
```

```
1.2M
./read_partitions/Fb_0/CBin_71/c7126.trinity.reads.fa.out/chrysalis/Component_bins/C
bin0
1.2M
./read_partitions/Fb_0/CBin_71/c7126.trinity.reads.fa.out/chrysalis/Component_bins
11M     ./read_partitions/Fb_0/CBin_71/c7126.trinity.reads.fa.out/chrysalis
0       ./read_partitions/Fb_0/CBin_71/c7126.trinity.reads.fa.out/__polish.chkpts
112K
./read_partitions/Fb_0/CBin_71/c7126.trinity.reads.fa.out/Trinity.tmp.fasta.salmon.i
dx
1.1T    ./read_partitions/Fb_0/CBin_71/c7126.trinity.reads.fa.out
```

```
440K
./read_partitions/Fb_0/CBin_107/c10786.trinity.reads.fa.out/chrysalis/Component_bins
/Cbin0
440K
./read_partitions/Fb_0/CBin_107/c10786.trinity.reads.fa.out/chrysalis/Component_bins
9.2M    ./read_partitions/Fb_0/CBin_107/c10786.trinity.reads.fa.out/chrysalis
0       ./read_partitions/Fb_0/CBin_107/c10786.trinity.reads.fa.out/__polish.chkpts
52K
./read_partitions/Fb_0/CBin_107/c10786.trinity.reads.fa.out/Trinity.tmp.fasta.salmon
.idx
1.3T    ./read_partitions/Fb_0/CBin_107/c10786.trinity.reads.fa.out
```

This filled up the directory. It is still “running” ie, did not end the run. Fb_0 is 4.6 TB while Fb_1 is only 1.3 GB.

- For SRR11659659 we have commands hung up here

```
wc -l
/projects/adornburg/venom_gland_project/venom_gland_ORP2.3.3_assemblies/SRR11659659_
ORP_2.3.3_assembly/assemblies/SRR11659659_ORP_2.3.3_output.trinity/recursive_trinity
.cmds*

   165598
/projects/adornburg/venom_gland_project/venom_gland_ORP2.3.3_assemblies/SRR11659659_
ORP_2.3.3_assembly/assemblies/SRR11659659_ORP_2.3.3_output.trinity/recursive_trinity
.cmds
   165593
/projects/adornburg/venom_gland_project/venom_gland_ORP2.3.3_assemblies/SRR11659659_
ORP_2.3.3_assembly/assemblies/SRR11659659_ORP_2.3.3_output.trinity/recursive_trinity
.cmds.completed
        0
/projects/adornburg/venom_gland_project/venom_gland_ORP2.3.3_assemblies/SRR11659659_
ORP_2.3.3_assembly/assemblies/SRR11659659_ORP_2.3.3_output.trinity/recursive_trinity
.cmds.ok
   331191 total
```

- Let us find the commands that are stuck

```
sort recursive_trinity.cmds >foo1
sort recursive_trinity.cmds.completed >foo2
diff foo1 foo2
```

Here we already have found the runs that are off with the disk usage command -

```
1.9M
./read_partitions/Fb_0/CBin_15/c1594.trinity.reads.fa.out/chrysalis/Component_bins/C
bin0
1.9M
./read_partitions/Fb_0/CBin_15/c1594.trinity.reads.fa.out/chrysalis/Component_bins
14M     ./read_partitions/Fb_0/CBin_15/c1594.trinity.reads.fa.out/chrysalis
0       ./read_partitions/Fb_0/CBin_15/c1594.trinity.reads.fa.out/__polish.chkpts
100K
./read_partitions/Fb_0/CBin_15/c1594.trinity.reads.fa.out/Trinity.tmp.fasta.salmon.i
dx
2.3T    ./read_partitions/Fb_0/CBin_15/c1594.trinity.reads.fa.out
.....
1.2M
./read_partitions/Fb_0/CBin_71/c7126.trinity.reads.fa.out/chrysalis/Component_bins/C
bin0
1.2M
./read_partitions/Fb_0/CBin_71/c7126.trinity.reads.fa.out/chrysalis/Component_bins
11M     ./read_partitions/Fb_0/CBin_71/c7126.trinity.reads.fa.out/chrysalis
0       ./read_partitions/Fb_0/CBin_71/c7126.trinity.reads.fa.out/__polish.chkpts
112K
./read_partitions/Fb_0/CBin_71/c7126.trinity.reads.fa.out/Trinity.tmp.fasta.salmon.i
dx
1.1T    ./read_partitions/Fb_0/CBin_71/c7126.trinity.reads.fa.out
.....
440K
./read_partitions/Fb_0/CBin_107/c10786.trinity.reads.fa.out/chrysalis/Component_bins
/Cbin0
440K
./read_partitions/Fb_0/CBin_107/c10786.trinity.reads.fa.out/chrysalis/Component_bins
9.2M    ./read_partitions/Fb_0/CBin_107/c10786.trinity.reads.fa.out/chrysalis
0       ./read_partitions/Fb_0/CBin_107/c10786.trinity.reads.fa.out/__polish.chkpts
52K
./read_partitions/Fb_0/CBin_107/c10786.trinity.reads.fa.out/Trinity.tmp.fasta.salmon
.idx
1.3T    ./read_partitions/Fb_0/CBin_107/c10786.trinity.reads.fa.out
.....
28K
./read_partitions/Fb_0/CBin_200/c20056.trinity.reads.fa.out/chrysalis/Component_bins
/Cbin0
28K
./read_partitions/Fb_0/CBin_200/c20056.trinity.reads.fa.out/chrysalis/Component_bins
8.2M    ./read_partitions/Fb_0/CBin_200/c20056.trinity.reads.fa.out/chrysalis
0       ./read_partitions/Fb_0/CBin_200/c20056.trinity.reads.fa.out/__polish.chkpts
40K
./read_partitions/Fb_0/CBin_200/c20056.trinity.reads.fa.out/Trinity.tmp.fasta.salmon
.idx
8.3M    ./read_partitions/Fb_0/CBin_200/c20056.trinity.reads.fa.out
```

```
.....
24K
./read_partitions/Fb_1/CBin_1248/c124850.trinity.reads.fa.out/chrysalis/Component_bi
ns/Cbin0
24K
./read_partitions/Fb_1/CBin_1248/c124850.trinity.reads.fa.out/chrysalis/Component_bi
ns
8.2M    ./read_partitions/Fb_1/CBin_1248/c124850.trinity.reads.fa.out/chrysalis
0
./read_partitions/Fb_1/CBin_1248/c124850.trinity.reads.fa.out/__polish.chkpts
40K
./read_partitions/Fb_1/CBin_1248/c124850.trinity.reads.fa.out/Trinity.tmp.fasta.salm
on.idx
8.3M    ./read_partitions/Fb_1/CBin_1248/c124850.trinity.reads.fa.out
```

- These may have become hung up from disk usage, but likely also became stuck.

This issue for Terabytes being used for read partitions that should be a few megabytes is that the Building BooPHF is caught in a seemingly endless loop constantly posting the elapse time end estimated finishing time, being printed to the **_salmon.406001.stderr file.**

```
cat  ./read_partitions/Fb_0/CBin_15/c1594.trinity.reads.fa.out/_salmon.406001.stderr
```

Output

```
index
["/projects/adornburg/venom_gland_project/venom_gland_ORP2.3.3_assemblies/SRR1165965
9_ORP_2.3.3_assembly/assemblies/SRR11659659_ORP_2.3.3_output.trinity/read_partitions
/Fb_0/CBin_15/c1594.trinity.reads.fa.out/Trinity.tmp.fasta.salmon.idx"] did not
previously exist  . . . creating it
[2023-03-27 16:47:44.497] [jLog] [warning] The salmon index is being built without
any decoy sequences.  It is recommended that decoy sequence (either computed
auxiliary decoy sequence or the genome of the organism) be provided during
indexing. Further details can be found at
https://salmon.readthedocs.io/en/latest/salmon.html#preparing-transcriptome-indices-
mapping-based-mode.
[2023-03-27 16:47:44.497] [jLog] [info] building index
out :
/projects/adornburg/venom_gland_project/venom_gland_ORP2.3.3_assemblies/SRR11659659_
ORP_2.3.3_assembly/assemblies/SRR11659659_ORP_2.3.3_output.trinity/read_partitions/F
b_0/CBin_15/c1594.trinity.reads.fa.out/Trinity.tmp.fasta.salmon.idx
[2023-03-27 16:47:44.498] [puff::index::jointLog] [info] Running fixFasta

[Step 1 of 4] : counting k-mers
```

```
[2023-03-27 16:47:44.506] [puff::index::jointLog] [info] Replaced 0 non-ATCG
nucleotides
[2023-03-27 16:47:44.506] [puff::index::jointLog] [info] Clipped poly-A tails from
0 transcripts
wrote 47 cleaned references
[2023-03-27 16:47:44.516] [puff::index::jointLog] [info] Filter size not provided;
estimating from number of distinct k-mers
[2023-03-27 16:47:44.517] [puff::index::jointLog] [info] ntHll estimated 52727
distinct k-mers, setting filter size to 2^20
Threads = 1
Vertex length = 25
Hash functions = 5
Filter size = 1048576
Capacity = 1
Files:
/projects/adornburg/venom_gland_project/venom_gland_ORP2.3.3_assemblies/SRR11659659_
ORP_2.3.3_assembly/assemblies/SRR11659659_ORP_2.3.3_output.trinity/read_partitions/F
b_0/CBin_15/c1594.trinity.reads.fa.out/Trinity.tmp.fasta.salmon.idx/ref_k25_fixed.fa
--------------------------------------------------------------------------------
Round 0, 0:1048576
Pass    Filling Filtering
1       0       0
2       0       0
True junctions count = 130
False junctions count = 89
Hash table size = 219
Candidate marks count = 859
--------------------------------------------------------------------------------
Reallocating bifurcations time: 0
True marks count: 643
Edges construction time: 0
--------------------------------------------------------------------------------
Distinct junctions = 130

allowedIn: 9
Max Junction ID: 192
seen.size():1545 kmerInfo.size():193
approximateContigTotalLength: 11982
counters for complex kmers:
(prec>1 & succ>1)=4 | (succ>1 & isStart)=0 | (prec>1 & isEnd)=0 | (isStart &
isEnd)=0
contig count: 187 element count: 18392 complex nodes: 4
# of ones in rank vector: 186
[2023-03-27 16:47:44.572] [puff::index::jointLog] [info] Starting the Pufferfish
indexing by reading the GFA binary file.
[2023-03-27 16:47:44.572] [puff::index::jointLog] [info] Setting the
index/BinaryGfa directory
```

```
/projects/adornburg/venom_gland_project/venom_gland_ORP2.3.3_assemblies/SRR11659659_
ORP_2.3.3_assembly/assemblies/SRR11659659_ORP_2.3.3_output.trinity/read_partitions/F
b_0/CBin_15/c1594.trinity.reads.fa.out/Trinity.tmp.fasta.salmon.idx
size = 18392
----------------------------------------
| Loading contigs | Time = 989.55 us
----------------------------------------
size = 18392
----------------------------------------
| Loading contig boundaries | Time = 1.9911 ms
----------------------------------------
Number of ones: 186
Number of ones per inventory item: 512
Inventory entries filled: 1
186
[2023-03-27 16:47:44.575] [puff::index::jointLog] [info] Done wrapping the rank
vector with a rank9sel structure.
[2023-03-27 16:47:44.575] [puff::index::jointLog] [info] contig count for
validation: 186
[2023-03-27 16:47:44.579] [puff::index::jointLog] [info] Total # of Contigs : 186
[2023-03-27 16:47:44.579] [puff::index::jointLog] [info] Total # of numerical
Contigs : 186
[2023-03-27 16:47:44.583] [puff::index::jointLog] [info] Total # of contig vec
entries: 613
[2023-03-27 16:47:44.583] [puff::index::jointLog] [info] bits per offset entry 10
[2023-03-27 16:47:44.583] [puff::index::jointLog] [info] Done constructing the
contig vector. 187
[2023-03-27 16:47:44.600] [puff::index::jointLog] [info] # segments = 186
[2023-03-27 16:47:44.600] [puff::index::jointLog] [info] total length = 18,392
[2023-03-27 16:47:44.600] [puff::index::jointLog] [info] Reading the reference
files ...
[2023-03-27 16:47:44.610] [puff::index::jointLog] [info] positional integer width =
15
[2023-03-27 16:47:44.610] [puff::index::jointLog] [info] seqSize = 18,392
[2023-03-27 16:47:44.610] [puff::index::jointLog] [info] rankSize = 18,392
[2023-03-27 16:47:44.610] [puff::index::jointLog] [info] edgeVecSize = 0
[2023-03-27 16:47:44.610] [puff::index::jointLog] [info] num keys = 13,928
[Building BooPHF]  6.31 %   elapsed:   0 min 2  sec   remaining:   0 min 36 s^Cec
```

- ^^^ The cat command "freaks out" populated the last line <[Building BooPHF]> elapse and remaining timing.
- This is a salmon dependency BBHash using this code <BooPHF.h>- https://github.com/rizkg/BBHash/blob/master/BooPHF.h

The fix!

The fix is easy enough. Force the **recursive_trinity.cmds** and the **recursive_trinity.cmds.completed** to be the same.

- We move the mv Trinity.tmp.fasta temp file found in the cXXX.trinity.reads.fa.out directory and rename in the final FASTA file output is following this schema on directory below. c.XXXX.trinity.reads.fa.out.Trinity.fasta
- Then remove the direcory.

```
cd
/scratch/jmcquil2/venom_gland_ORP2.3.3_assemblies/XXXX_ORP_2.3.3_assembly/assemblies
/XXXX_ORP_2.3.3_output.trinity/read_partitions/Fb_X/CBin_XX/cXXXX.trinity.reads.fa.o
ut
mv Trinity.tmp.fasta ../cXXXX.trinity.reads.fa.out.Trinity.fasta
cd ..
rm -r cXXXX.trinity.reads.fa.out
```

Summary: In the future we set the ORP runs to 12 hours, to make sure they one, are not hung up, but more importantly that they do not fill you the all space in our partition.