Zuse Institute Berlin

Big Data Analytics on Cray XC Series DataWarp using Hadoop, Spark and Flink

CUG2016

R. Schmidtke

Zuse Institute Berlin

Big Data Analytics on Cray XC Series DataWarp using Hadoop, Spark and Flink

CUG2016

R. Schmidtke

Update May 12, 2016: How absence of DVS client caching can mess up your results in practice.

TDS at ZIB

Test & Development System: mostly exclusive usage.

16 XC30 compute nodes, 10-core IvyBridge Xeon, 32 GiB memory.

8 DataWarp nodes, 2x1.6 TiB SSDs, very quiet, persistent & striped (8MiB) & scratch.

2 Lustre (80 OST/2.3 PiB, 48 OST/1.4 PiB), production usage, no striping.

TDS at ZIB

Test & Development System: mostly exclusive usage.

16 XC30 compute nodes, 10-core IvyBridge Xeon, 32 GiB memory.

8 DataWarp nodes, 2x1.6 TiB SSDs, very quiet, persistent & striped (8MiB) & scratch.

2 Lustre (80 OST/2.3 PiB, 48 OST/1.4 PiB), production usage, no striping.

Perfect for Big Data!

Approach

Hadoop, Spark and Flink as common data processing engines on CCM.

TeraSort, Streaming and SQL Join as well understood big data applications.

Approach

Hadoop, Spark and Flink as common data processing engines on CCM.

TeraSort, Streaming and SQL Join as well understood big data applications.

Approach

Hadoop, Spark and Flink as common data processing engines on CCM.

TeraSort, Streaming and SQL Join as well understood big data applications.

: Robust but lots of I/O because of shuffle.

:

Approach

Hadoop, Spark and Flink as common data processing engines on CCM.

TeraSort, Streaming and SQL Join as well understood big data applications.

: Robust but lots of I/O because of shuffle.

: Great scaling but many IOPS (as we've heard multiple times this week already, and will again in 10 minutes).

:

Approach

Hadoop, Spark and Flink as common data processing engines on CCM.

TeraSort, Streaming and SQL Join as well understood big data applications.

: Robust but lots of I/O because of shuffle.

: Great scaling but many IOPS (as we've heard multiple times this week already, and will again in 10 minutes).


Flink: Flink?

Approach

Hadoop, Spark and Flink as common data processing engines on CCM.

TeraSort, Streaming and SQL Join as well understood big data applications.

: Robust but lots of I/O because of shuffle.

: Great scaling but many IOPS (as we've heard multiple times this week already, and will again in 10 minutes).


Flink: Flink? Think Spark with support for true stream processing, off-heap memory and support for iterations.

Suddenly: Reality

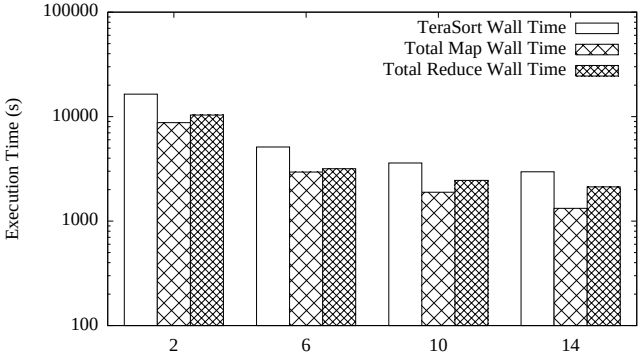
Tuning with that many parameters (TeraSort/Streaming/SQL, YARN, HDFS, Hadoop/Spark/Flink on DataWarp/Lustre) quickly becomes a life task.

We'll take you on a lightweight version of our journey top-down, let's start with TeraSort on Hadoop and DataWarp (i.e. HDFS data and Hadoop temporary directories).

Suddenly: Reality

Tuning with that many parameters (TeraSort/Streaming/SQL, YARN, HDFS, Hadoop/Spark/Flink on DataWarp/Lustre) quickly becomes a life task.

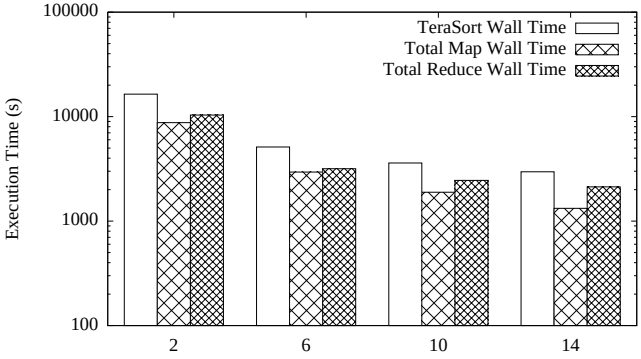
We'll take you on a lightweight version of our journey top-down, let's start with TeraSort on Hadoop and DataWarp (i.e. HDFS data and Hadoop temporary directories).



Suddenly: Reality

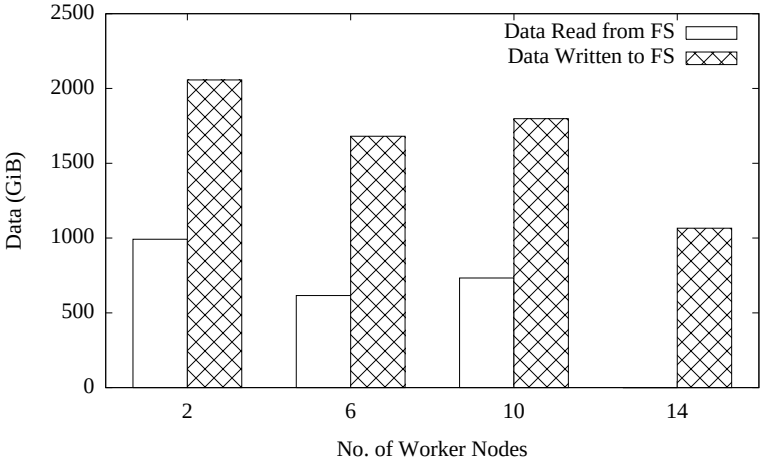
Tuning with that many parameters (TeraSort/Streaming/SQL, YARN, HDFS, Hadoop/Spark/Flink on DataWarp/Lustre) quickly becomes a life task.

We'll take you on a lightweight version of our journey top-down, let's start with TeraSort on Hadoop and DataWarp (i.e. HDFS data and Hadoop temporary directories).

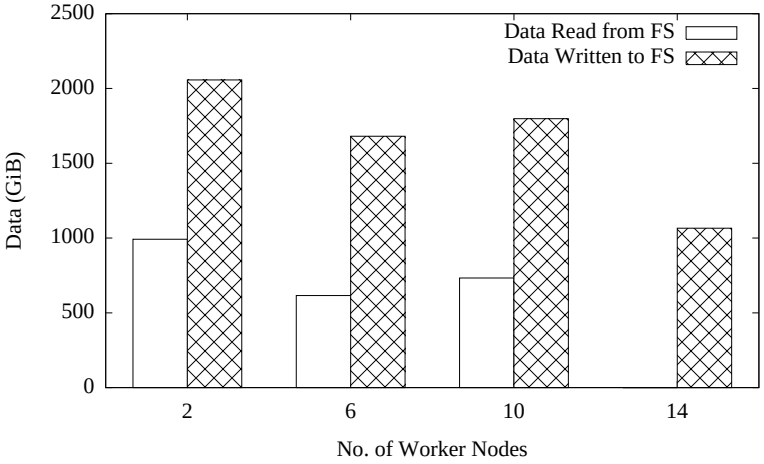


Between 4h34m to 0h49m, around 30 MiB/s per-node throughput.
(Lustre: 3h18m to 0h25m, around 50 MiB/s per-node throughput.)

Is it I/O? Hadoop FS Counters?

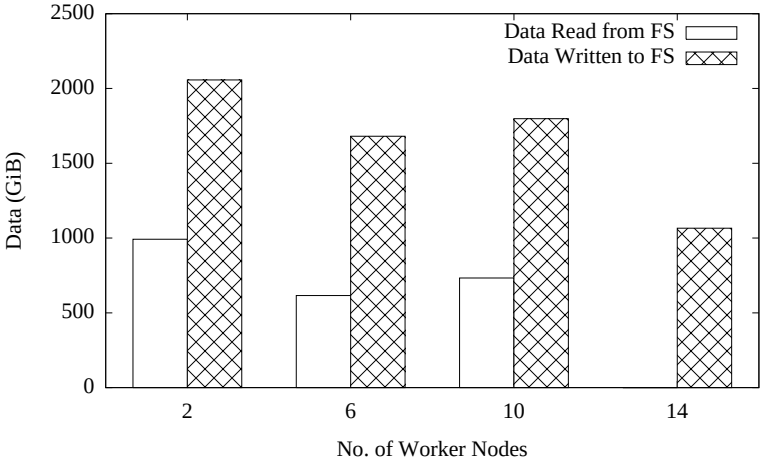


Is it I/O? Hadoop FS Counters?



Maybe? But looking at the counters ...

Is it I/O? Hadoop FS Counters?

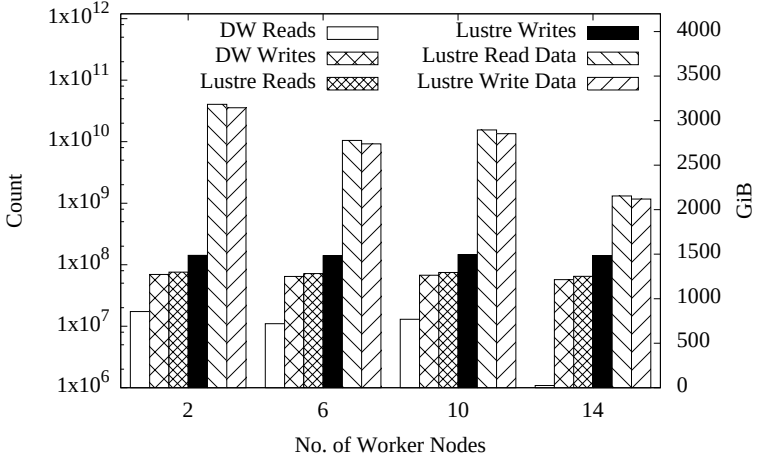


Maybe? But looking at the counters ...

We should see at least 2 TiB of read/write every run.

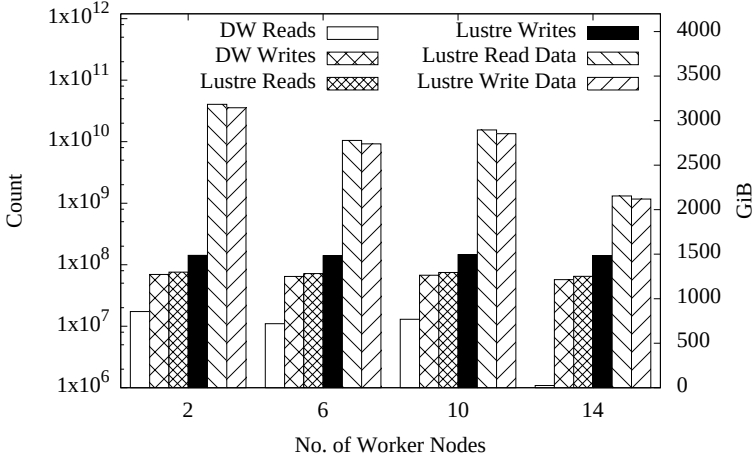
We must go deeper ...

DVS & Lustre FS counters to the rescue!



We must go deeper ...

DVS & Lustre FS counters to the rescue!

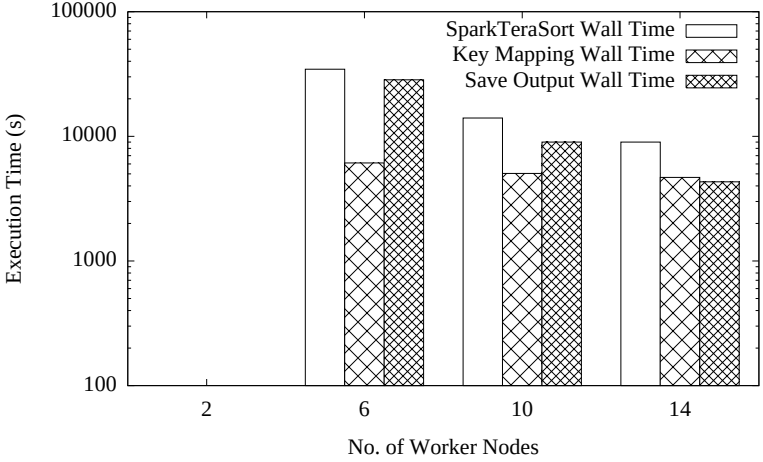


Aha! Between 2 and 3 TiB read/write, so apparently Hadoop FS counters only count shuffle and spill.
DVS counter issues:

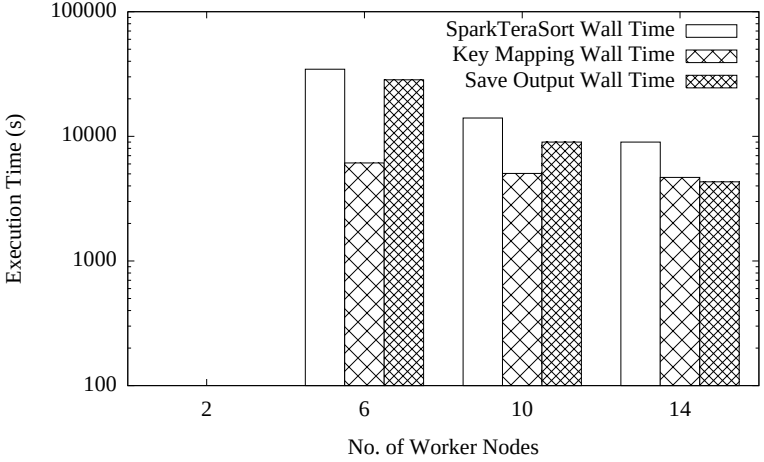
- Total no. of read/written bytes.
- Reported max. read/write sizes of 64 KiB vs. calculated avg. read/write sizes 192 KiB to 2 MiB.

No. of reads/writes DW vs. Lustre?

What about Spark?



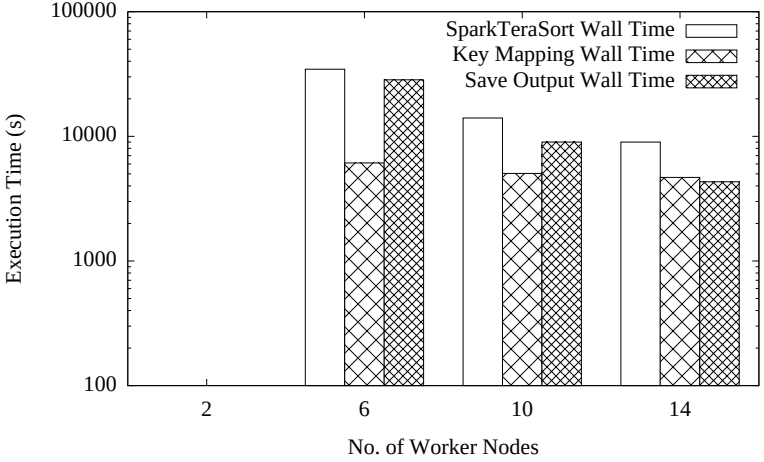
What about Spark?



Fail completely on two nodes.

Between 9h36m and 2h30m, 2x - 3x slower than Hadoop.
(Lustre: 2h18m and 0h25m, almost like Hadoop.)

What about Spark?

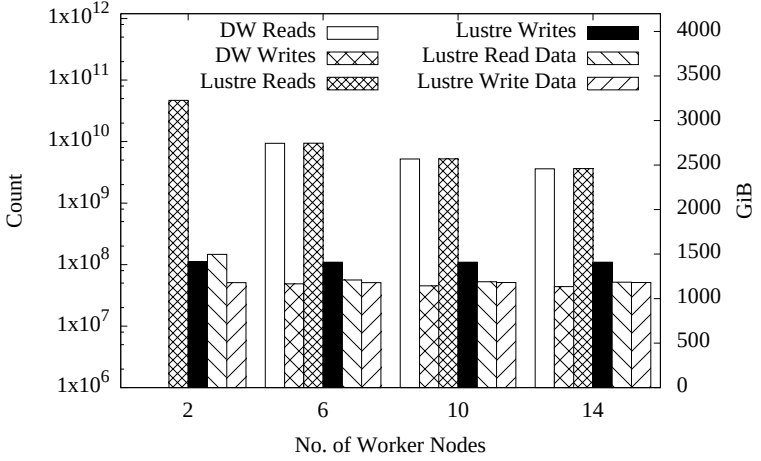


Fail completely on two nodes.

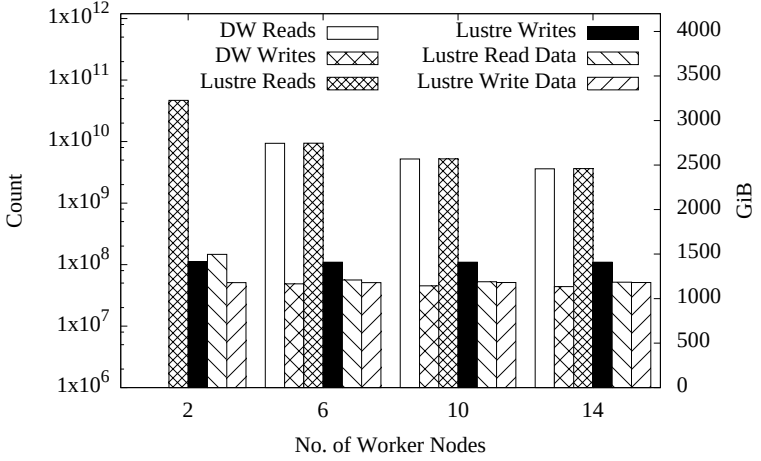
Between 9h36m and 2h30m, 2x - 3x slower than Hadoop.
(Lustre: 2h18m and 0h25m, almost like Hadoop.)

Bummer, but at least it scales better.

Count the counters



Count the counters



2x - 3x less data read/written, 1 TiB each is the minimum.

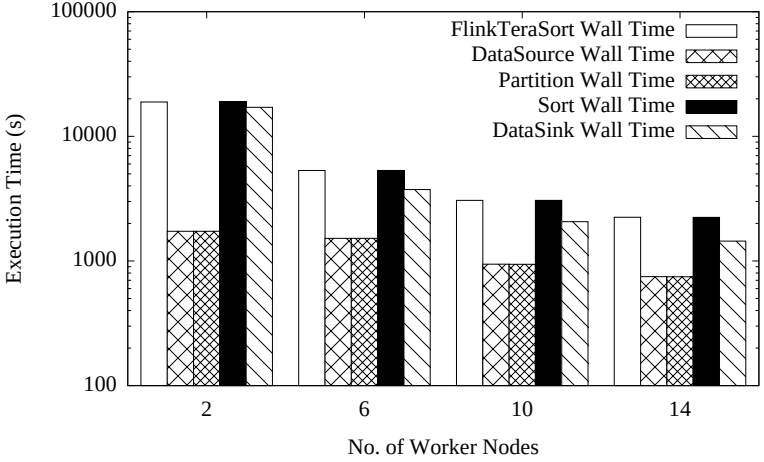
Same number of writes.

1000x the number of reads.

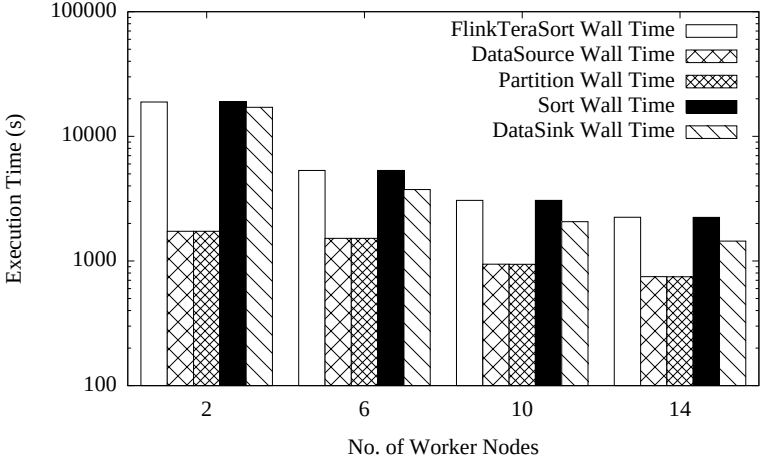
That's 100 bytes per read.

2.5x - 5x the number of opens (not shown).

Flink


100000
10000
1000
100
Execution Time (s)
2
6
10
14
No. of Worker Nodes
FlinkTeraSort Wall Time
DataSource Wall Time
Partition Wall Time
Sort Wall Time
DataSink Wall Time

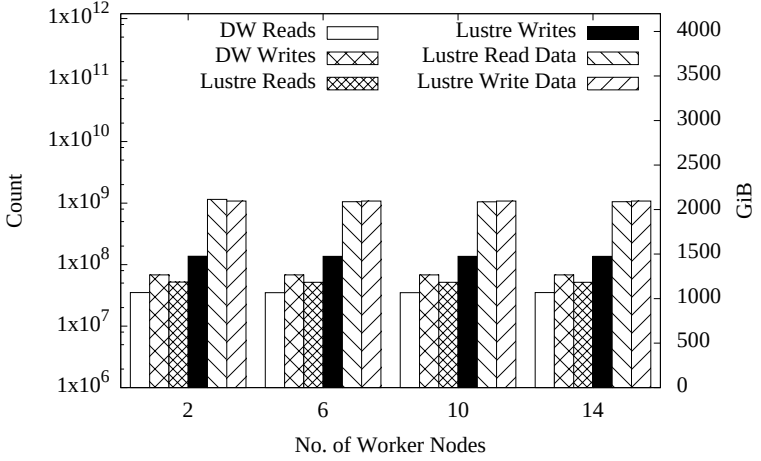
Flink



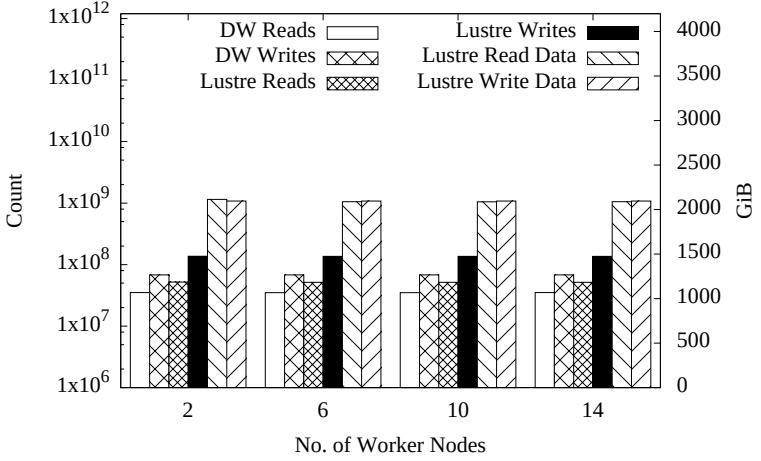
Between 5h14m and 0h37m.
(Lustre: 5h12m and 0h11m.)

At least it's a *bit* faster, half of the time.

Counting on Flink



Counting on Flink



Very constant I/O profile.

Why 2 TiB of data read/written? 1 TiB each should be enough, see Spark.

Almost exactly same I/O for 14 nodes as Hadoop, so operators must be more efficient.

Fast-forward two more benchmarks

... Flink wins throughput during TeraSort, Hadoop comes in 2nd, Spark is 3rd.[1]

... Spark wins throughput during Streaming benchmarks[1], Flink wins latency.

... Spark wins throughput during SQL[1], Flink comes in 2nd[2], Hadoop is 3rd.

... DataWarp configuration always loses to corresponding Lustre configuration, always.

[1] for the configurations it does not crash on
[2] its Table API is still beta though

Conclusions ... well, experiences.

Small site, more disk spill than necessary, however this helps our file system comparison tests.

Absolute results are bad, relation between frameworks and file systems nonetheless significant:
There are use cases for each framework, highly configuration dependent.
Don't use DataWarp without caching and small transfer sizes.

CCM can be difficult to work with.

R/W memory mapped files are not supported on DataWarp.

Spark fails to run successfully a surprising number of times.

IOR with 64 KiB reads/writes *roughly* agrees with Hadoop FS counters.

What we don't yet know

Why are there more reads/writes on Lustre than on DataWarp?

Why do the DVS counters report inconsistent values in one case?

Where does Flink's I/O come from?

How do IPC Rx/Tx bytes relate to actually read/received data?

What we don't yet know

Why are there more reads/writes on Lustre than on DataWarp?

Why do the DVS counters report inconsistent values in one case?

Where does Flink's I/O come from?

How do IPC Rx/Tx bytes relate to actually read/received data?

When do we get DataWarp Stage 2?