A Model of Checkpoint Behavior for Parallel Scientific Applications

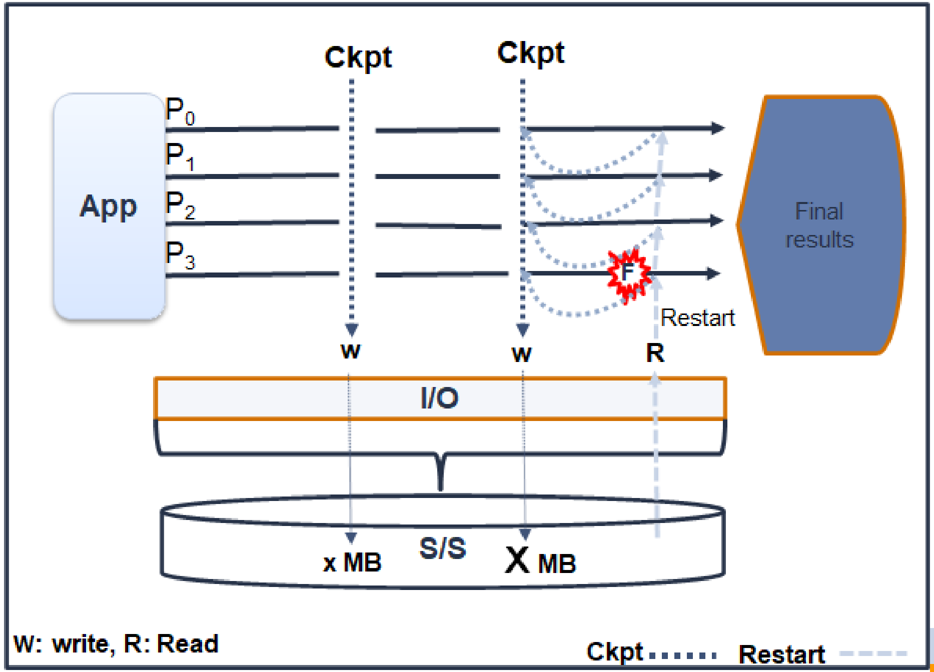
Betzabeth León, ***Sandra Méndez****, Daniel Franco, Dolores Rexachs, and Emilio Luque*

Agenda

- Introduction
- Objective
- Model of the Checkpoint Behavior
- Experimentation
- Conclusions

Context

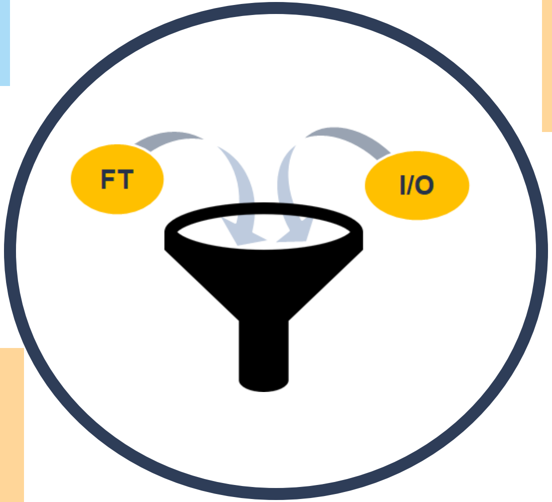
- **Checkpoints are a Fault Tolerance (FT) strategy** which require intensive large-scale access to the storage system, through I/O operations.
- It is a **defensive I/O** that is used for restarting a job in the event of application failure in order to retain the state of the computation, and hence the forward progress since the last checkpoint.
- The information of the **state of a process is stored periodically in a <u>stable</u> storage system, <u>suspending the execution</u>** of this while saving it and consuming resources of I/O and bandwidth of the network.
- There are **<u>checkpoints triggered by time (interval) and by events (when something happens)</u>**, but normally they depend on the interval, which have a low frequency.



Motivation

Applications that run on a large scale can spend more than 50% of total time storing checkpoints, restarting and redoing lost work, which is one of the main causes of overhead.

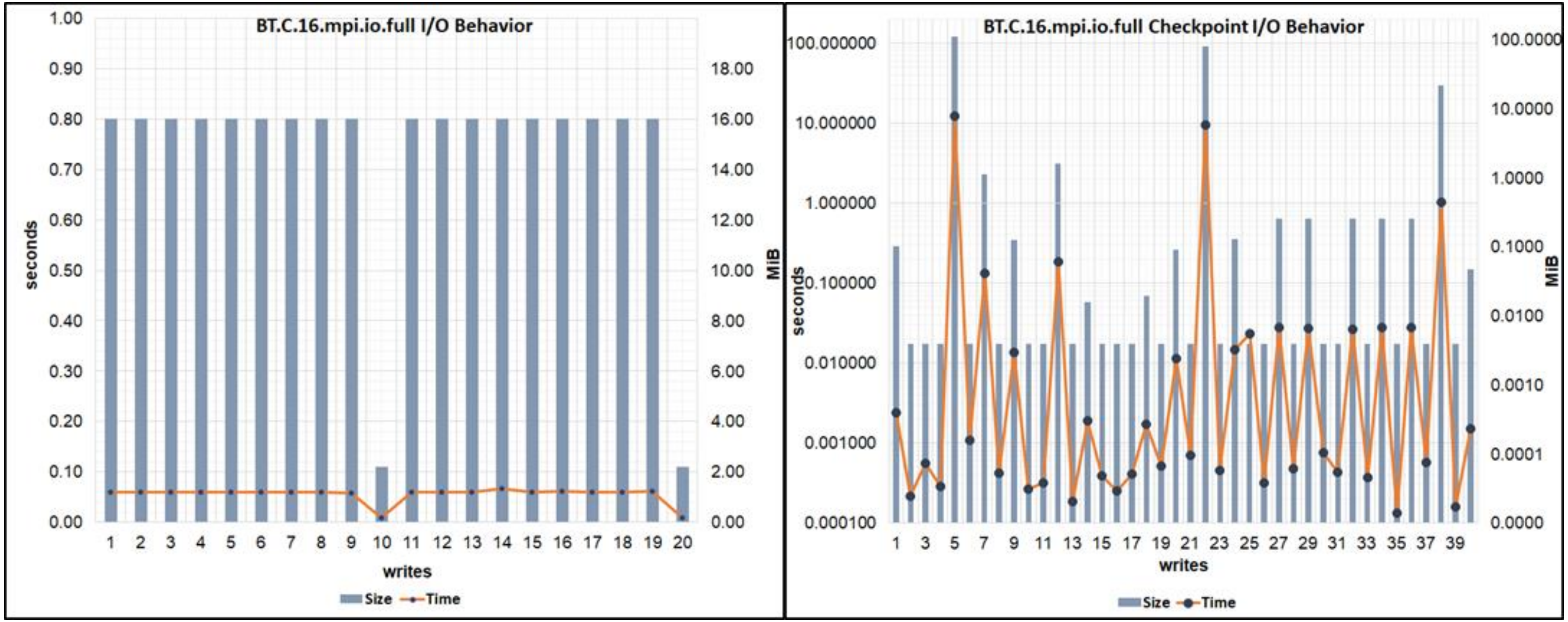
This requires continuous interaction with the storage system and consequently occupies a large amount of space in terabytes of data. Therefore, the checkpoint can easily collapse the I/O system.



The checkpointing operation is an I/O-intensive write operation, which can be executed on a large number of computing nodes, which would generate thousands of fles.

For these types of strategies, such as checkpoints, to be useful on a large scale, the normal execution of the application should be afected as little as possible.

Motivation



I/O behavior (writes size and time)

Agenda

- Introduction
- Objective
- Model of the Checkpoint Behavior
- Experimentation
- Conclusions

Objective

Model of checkpoint behavior for MPI parallel applications that performs Parallel I/O.

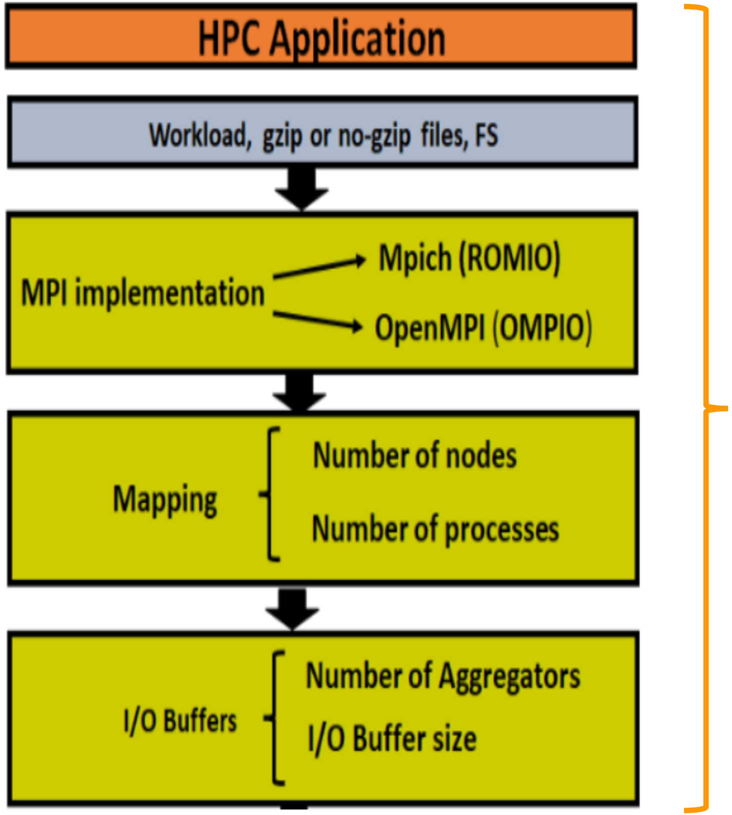
- It is focused on the checkpoint file sizes in relation to different MPI parameters.
- We analyze coordinated checkpoints carried out at user-layer level by the DMTCP library.
- We focus our study on the parallel applications that perform parallel I/O at MPI-IO level.
- Our model describes the behavior of the checkpoint size based on the number of processes, mapping, the number I/O aggregators and buffering size utilized by the two-phase I/O optimization technique.

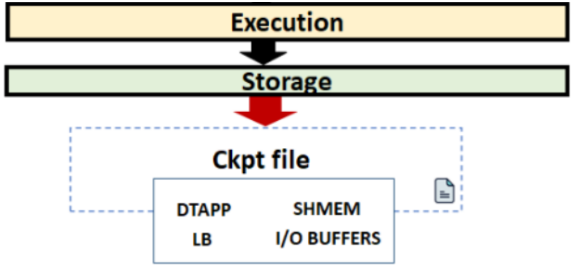
The model can be useful when selecting what type of checkpoint configuration is more appropriate according to the applications characteristics and the available resources.

Agenda

- Introduction
- Objective
- Model of the Checkpoint Behavior
- Experimentation
- Conclusions

Parameters that impact on the size and time of the Checkpoint

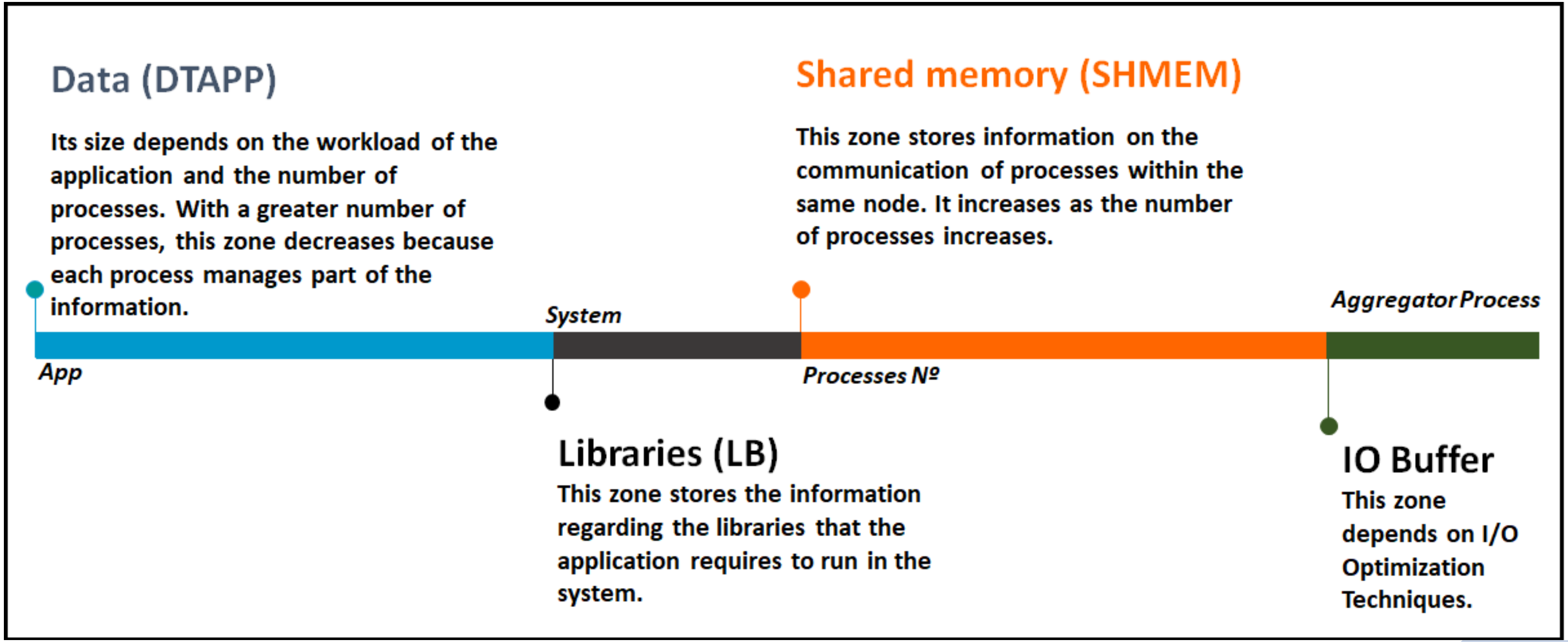




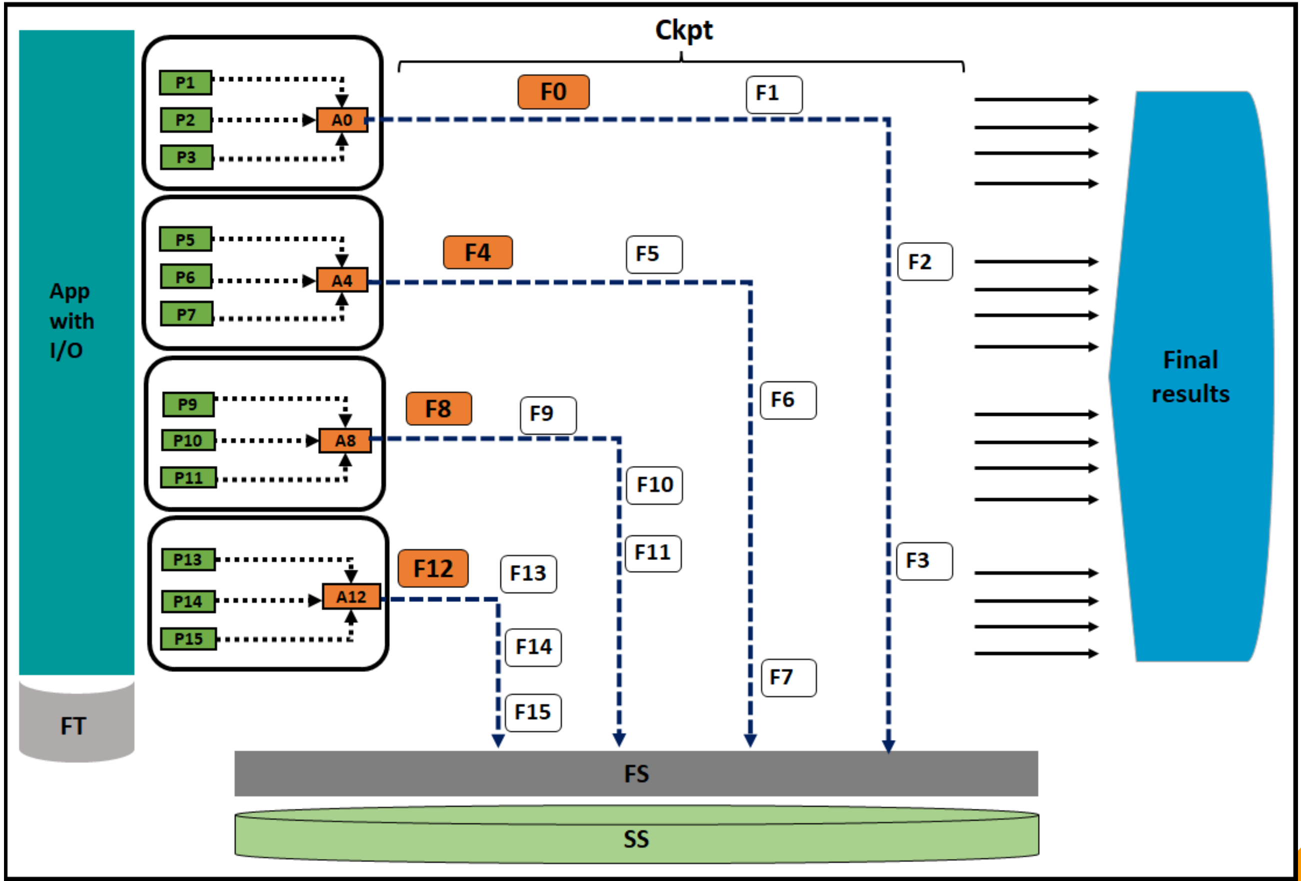
Checkpoint's size is in function of the workload (W), the number of processes(Np), the number of nodes (Nn), and the I/O aggregators (Na).

$$CkptSize(_{APP\text{-}IOfull}) = f(W, N_p , N_n, N_a)$$

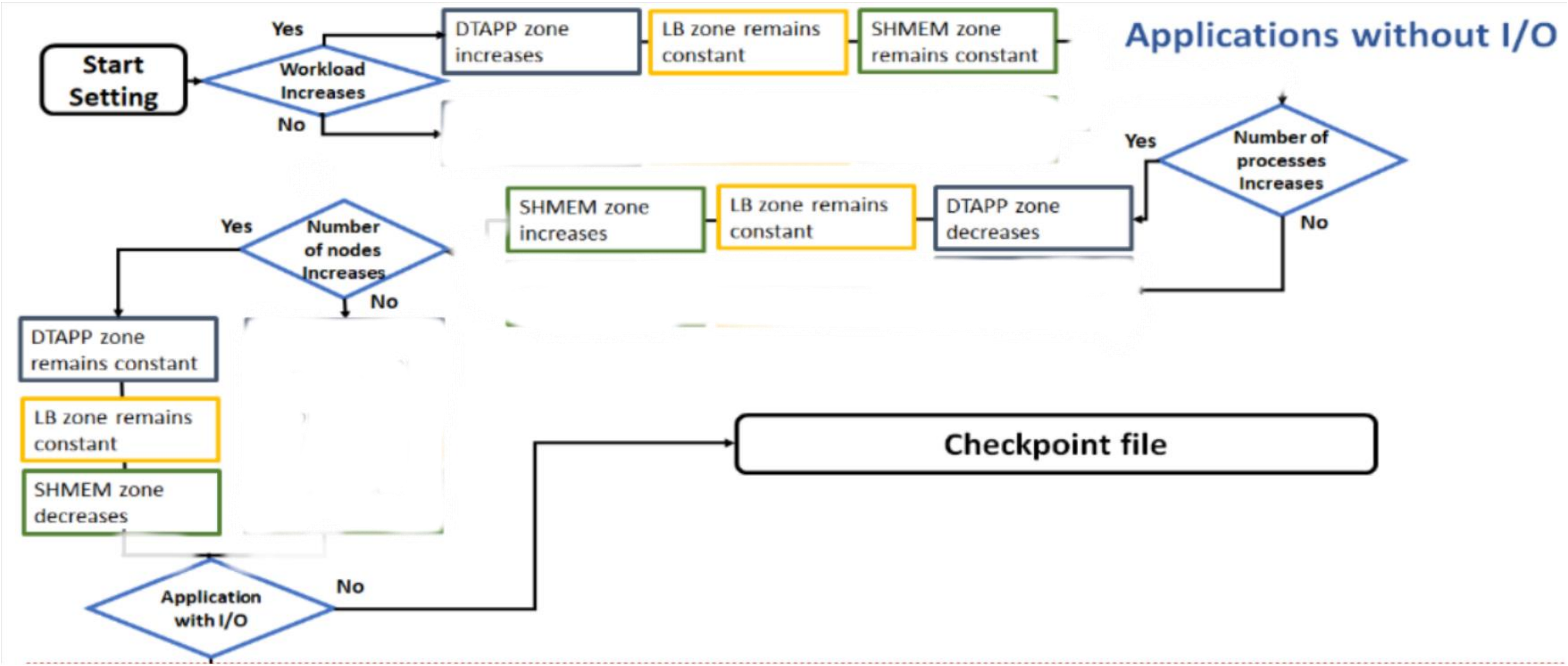
Zones that make up the checkpoints performed by applications with I/O



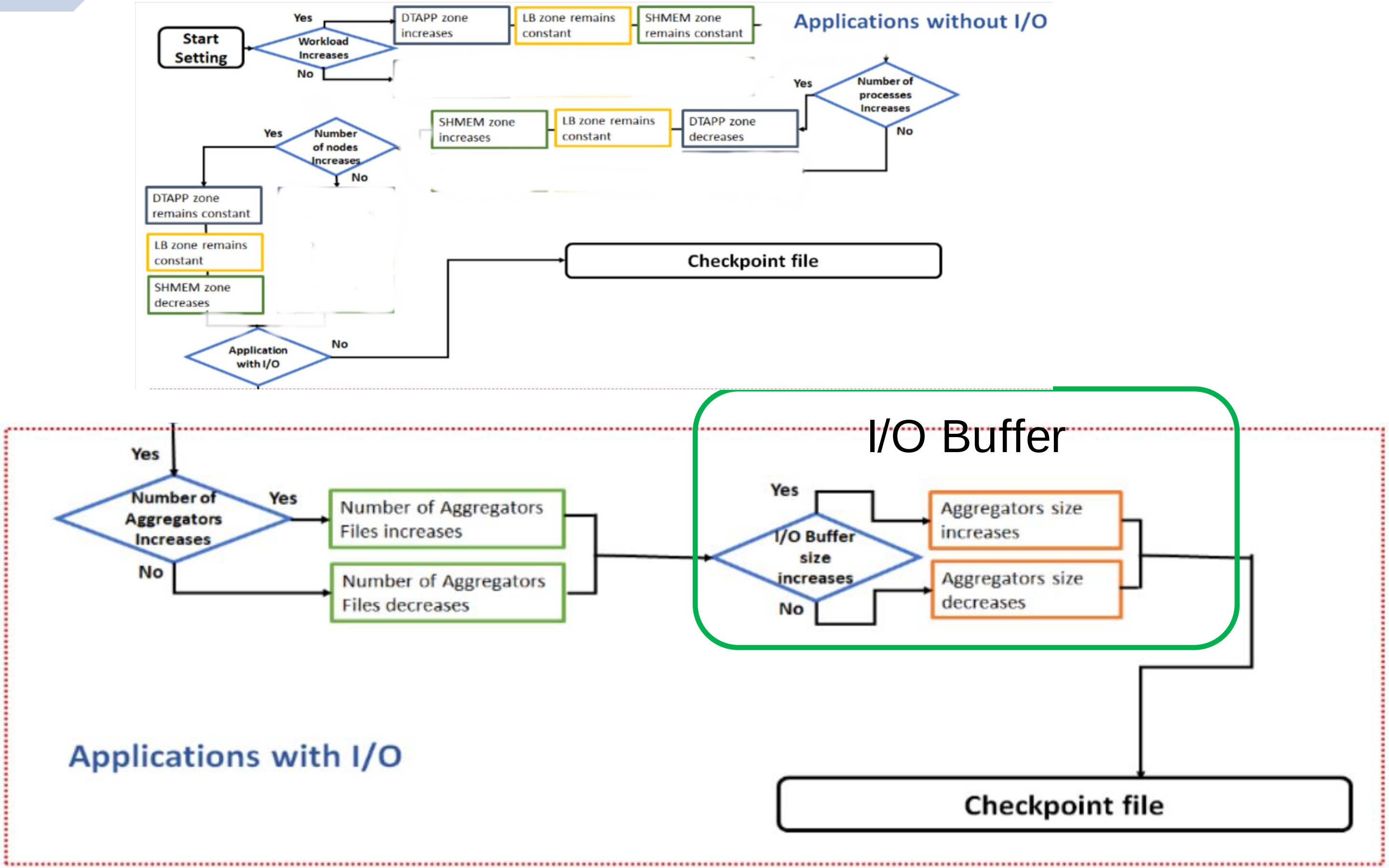
Checkpoint data layout for a parallel application that performs I/O by using collective operations run in four compute nodes.



Behavior of Checkpoint Zones



Behavior of Checkpoint Zones



Agenda

- Introduction
- Objective
- Model of the Checkpoint Behavior
- Experimentation
- Conclusions

Experimental Validation

- Compute nodes:
  - AMD Athlon(™) II X4 610e CPU 2.4GHz, processors: 1, CPU cores: 4, memory: 16 GiB.
  - AMD Opteron(™) 6200 @ CPU 1.56 GHz, processors: 4, CPU cores: 16, memory: 256 GiB.

- Software stack:
  - MPICH 3.2.1.
  - OpenMPI 4.1.1.
  - DMTCP-2.4.5.
  - NAS Parallel Benchmark BT-IO-FULL.
  - Instrumentation: readDMTCP

- I/O system:
  - NFS

- I/O Library:
  - ROMIO (MPICH 3.2.): buffer size by default for collective I/O = 16 MiB
  - OMPIO (Open MPI 4.1.1): buffer size by default for collective I/O = 32 MiB

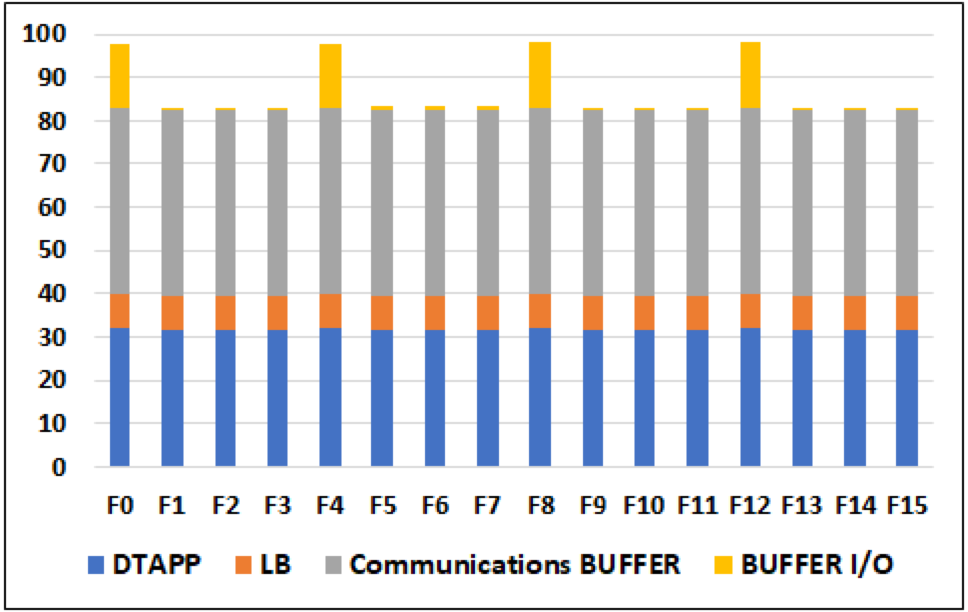
Analyzing the impact of aggregators processes number on the checkpoint file size

Bench.	BT.B.MPI.IO FULL				BT.C.MPI.IO FULL			
Np.	4	9	16	25	4	9	16	25
Map.	1n x 4p	2n x 4p 1n x 1p	4n x 4p	6n x 4p 1n x 1p	1n x 4p	3n x 3p	4n x 4p	6n x 4p 1n x 1p
F0	**182.10**	**122.11**	**98.43**	**88.24**	**482.90**	**266.10**	**184.01**	**149.95**
F1	165.60	106.26	83.74	74.34	466.18	249.05	164.99	128.72
F2	166.63	106.00	84.02	74.60	466.16	248.80	165.25	128.98
F3	166.64	106.00	83.80	74.34	466.16	248.79	165.17	128.75
F4		**122.12**	**98.48**	**85.21**		**266.00**	**183.96**	**149.84**
F5		106.22	83.74	74.48		248.79	165.00	128.71
F6		105.99	83.96	74.59		248.84	165.25	129.02
F7		106.26	83.74	74.34		249.08	165.05	128.76
F8		**108.32**	**98.35**	**86.32**		**253.23**	**184.02**	**149.85**
F9			83.70	74.21			164.99	128.46
F10			83.96	74.48			165.25	128.72
F11			83.75	74.47			165.04	128.72
F12			**98.38**	**86.64**			**184.00**	**149.78**
F13			83.71	74.48			165.03	128.72
F14			84.08	74.50			165.25	128.71
F15			83.75	74.40			165.03	128.46
F16				**86.40**				**152.08**
F17				74.36				128.70
F18				74.59				129.02
F19				74.57				128.72
F20				**86.23**				**149.91**
F21				74.46				128.70
F22				74.73				129.02
F23				74.47				128.76
F24				**73.34**				**137.12**

Checkpoint files sizes (MiB) for the BT-IO Class B, C FULL benchmark (MPICH)

- An aggregator per node has been generated (F0, F4, F8, F12, F16, F20, F24).
- These are related to the mapping.
- Size files aggregators 14MiB-20MiB.

Size (MiB) of the zones that make up the checkpoint files in applications with I/O Full (Mpich)



Time spent by the checkpoint and by the application without checkpoint and with checkpoint with different mapping and I/O buffer size (BT.B.16.MPI.IO FULL)(Mpich)

Aggregator Nº:	1	2	4	1	2	4
Aggregator Size:	32MiB	32MiB	32MiB	64MiB	64MiB	64MiB
Mapping:	1nx16p	2nx8p	4nx4p	1nx16p	2nx8p	4nx4p
Time (s)	Time (s)	Time (s)	Time (s)	Time (s)	Time (s)	Time (s)
App	70.77	86.49	122.47	70.21	89.57	235.38
Ckpt	51.09	26.76	23.24	47.01	27.17	25.52
App+Ckpt	122.09	111.55	145.38	118.77	114.69	262.2

Agenda

- Introduction
- Objective
- Model of the Checkpoint Behavior
- Experimentation
- Conclusions

Conclusions

- Our model describes the behavior of the checkpoint size based on the mapping (number of processes and nodes), the number I/O aggregators and collective buffering technique.

- By analyzing the coordinated behavior of the checkpoint generated in the user layer by the DMTCP library, we identified the impact of the I/O strategy parameters on the different zones of the checkpoint file.

- Therefore, at large scale, the configuration of the aggregators and these I/O components that impact on the size of the checkpoint could be significant.

- With this model, the size of the checkpoint files of applications with I/O can be estimated with few resources, analyzing what happens in a node with few processes and the size can be known when the number of nodes changes, the number of processes and/or the configuration of the aggregator processes. In this way, the size of the stable storage necessary to save the files generated by the checkpoint can be previously known.

Thank you!

Sandra Méndez
sandra.mendez@uab.es
sandra.mendez@bsc.es