On the feasibility of message logging in hybrid hierarchical FT protocols

Tatiana V. Martsinkevich, Franck Cappello

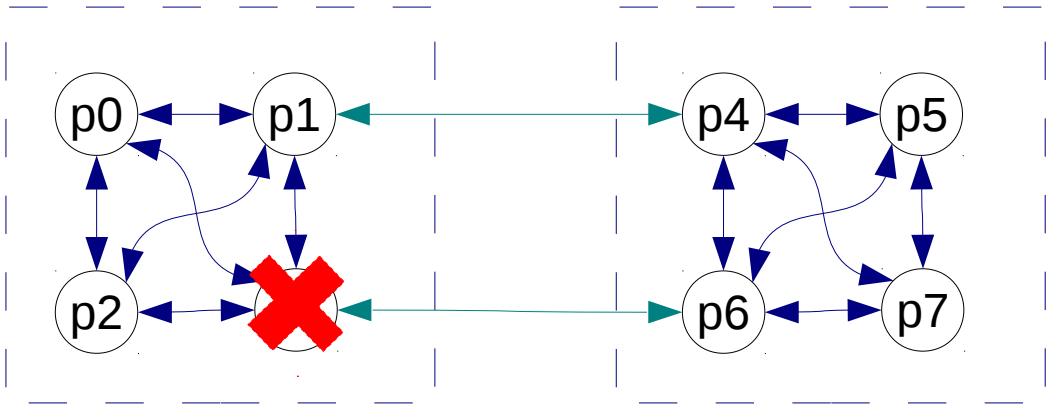
What FT protocols have a future?

- **Currently used: app level coordinated checkpointing**
  - **Everyone access PFS to get the checkpoint**
  - **Everyone has to re-execute → waste of energy**
- **On large scale may be not feasible**

Can we do better?

Hybrid FT protocols(1)

- **Divide processes into clusters**
  - **Coordinated checkpointing inside the cluster**
  - **Message logging for inter-cluster communication**
- **Advantages**
  - **Restart only part of execution → less load on PFS & save energy**
  - **Can (potencially) use idle PEs for something else**



Hybrid FT protocols(2)

- **Existing hybrid FT protocols: SPBC* etc.**
  - **Low overhead in failure-free execution**
  - **Recovery as fast as failure-free or even faster**
- **Message logging in hybrid protocols**
  - **We only have so much memory to use!**

➔ **Top 10 supercomputers from the top500 list have in average 1GB of RAM per core**

(*) *"SPBC: Leveraging the Characteristics of MPI HPC Applications for Scalable Checkpointing".* T.Ropars, T.Martsinkevich, A. Guermouche, A. Schiper, F.Cappello ACM/IEEE SC13

Memory requirenments of scientific applications

- **Tendency:**
  - **~300MB per core**
  - **Doubling # of procs doesn't halve memory footprint**

Application	#Procs.	Avg. mem. footprint (per-proc.)	Max. mem. footprint (per-proc.)	Footprint reduction (w/2x procs.)	Est. tot. footprint (pessimistic)
MILC	64	0.30 GB	0.31 GB	-33%	19.20 GB
	32	0.45 GB	0.48 GB	-38%	14.40 GB
	16	0.73 GB	0.80 GB	N/A	11.68 GB
GADGET2	128	0.52 GB	0.68 GB	-32%	66.56 GB
	64	0.77 GB	1.00 GB	-42%	49.28 GB
	32	1.32 GB	1.83 GB	N/A	42.24 GB
WRF311	64	0.22 GB	0.29 GB	-19%	14.08 GB
	32	0.27 GB	0.34 GB	-23%	8.64 GB
	16	0.35 GB	0.41 GB	N/A	5.60 GB
SOCORRO	64	0.23 GB	0.24 GB	-12%	14.72 GB
	32	0.26 GB	0.28 GB	-24%	8.32 GB
	16	0.34 GB	0.35 GB	N/A	5.44 GB

* Milan Pavlovic et al. Can Manycores Support the Memory Requirements of Scientific Applications? *ISCA'10 Proceedings of the 2010 international conference on Computer Architecture*

Approaches to limited memory

- **Change checkpointing frequency in cluster**
  - **Logs are flushed with the chp()**
- **Flush part of logs to dedicated logger nodes**
- **Change clustering**
  - **Less clusters but bigger size → less to log**

Approaches to limited memory

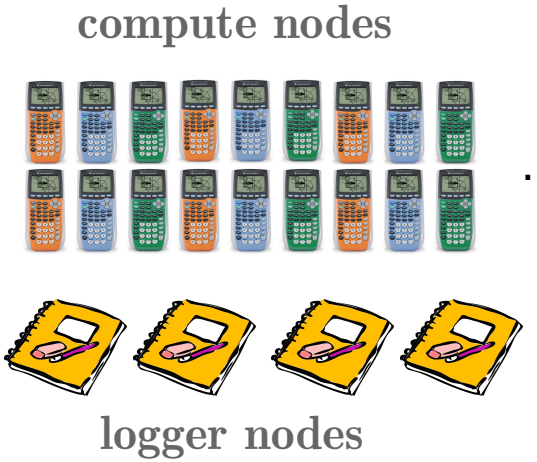
- **Change checkpointing frequency in cluster**
  - **Logs are flushed with the chp()**
- **Flush part of logs to dedicated logger nodes**
- **Change clustering**
  - **Less clusters but bigger size → less to log**

Case study

- **Applications:**
  - ***POP2*: ocean component of CESM**
  - ***CM1*: model to study atmospheric phenomena (thunderstorms)**
- **256 PEs (32 nodes)**
- **Platform: GRID5000**
  - **Node x 2 Intel Xeon CPUs (2.27GHz) x 4 cores, 16GB RAM**
  - **Infiniband-40G**

Dedicated loggers

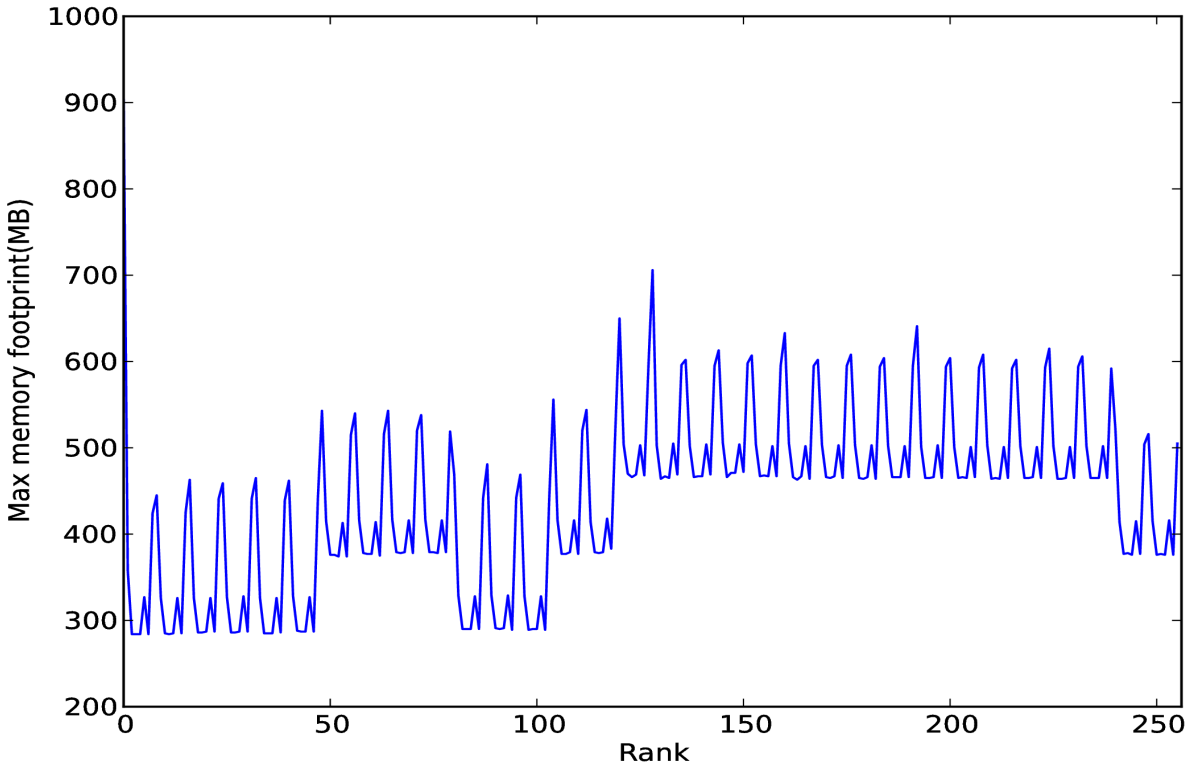
- 4 nodes, each with
  - 16GB of RAM
  - 8 logger MPI ranks



- If compute rank runs out of memory
  - Flush part of log to logger's memory
    - Free enough memory to run for another 10sec with current log growth rate

Case study 1: POP2

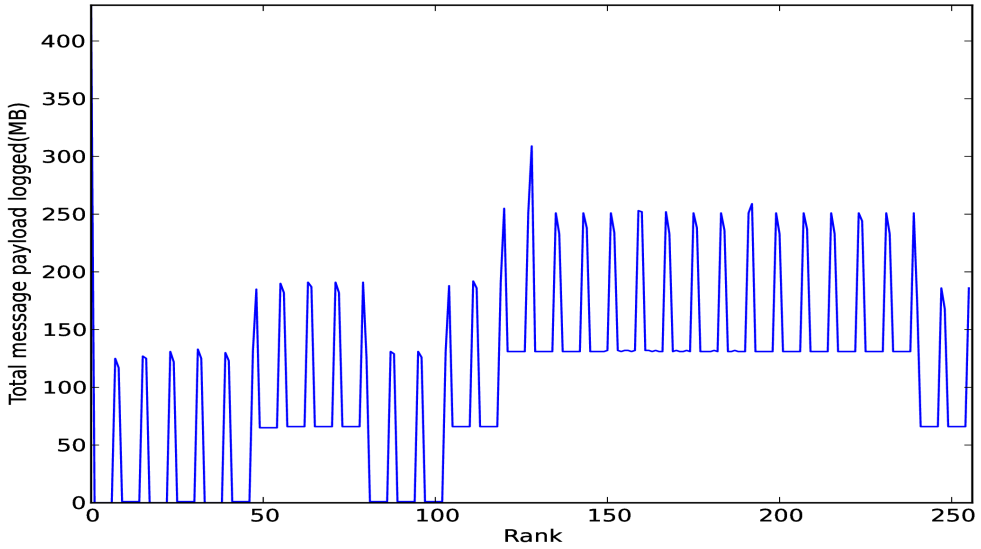
maximum memory footprint

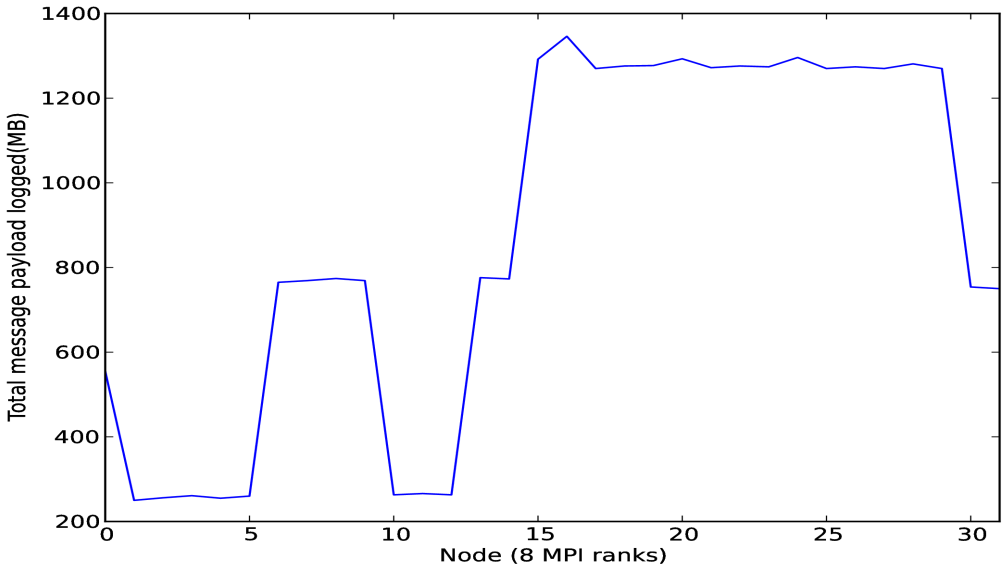


- **Simulate 10 days (~5 mins of execution)**
- **Max memory: ~900MB**
- **Avg memory: ~440MB**

(High memory utilization may be due to the initialization stage)

POP2: TOTAL LOG SIZE



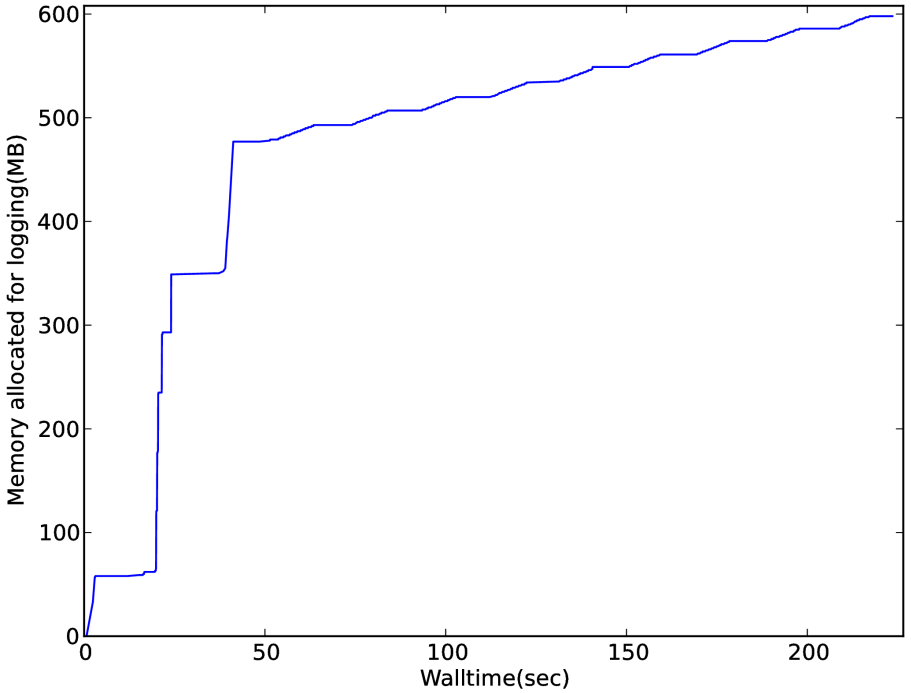


8 clusters (32 PEs per cluster)

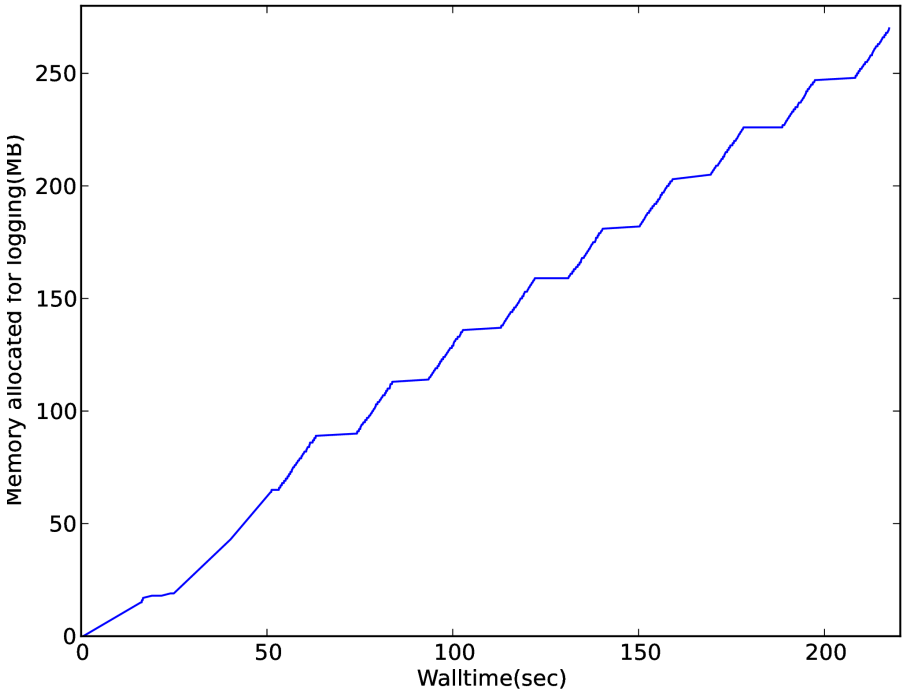
Average log per rank: 109MB

- **Max log per rank: 430MB**
- **Min log per rank: 0MB**

Memory allocated for logging: no limit



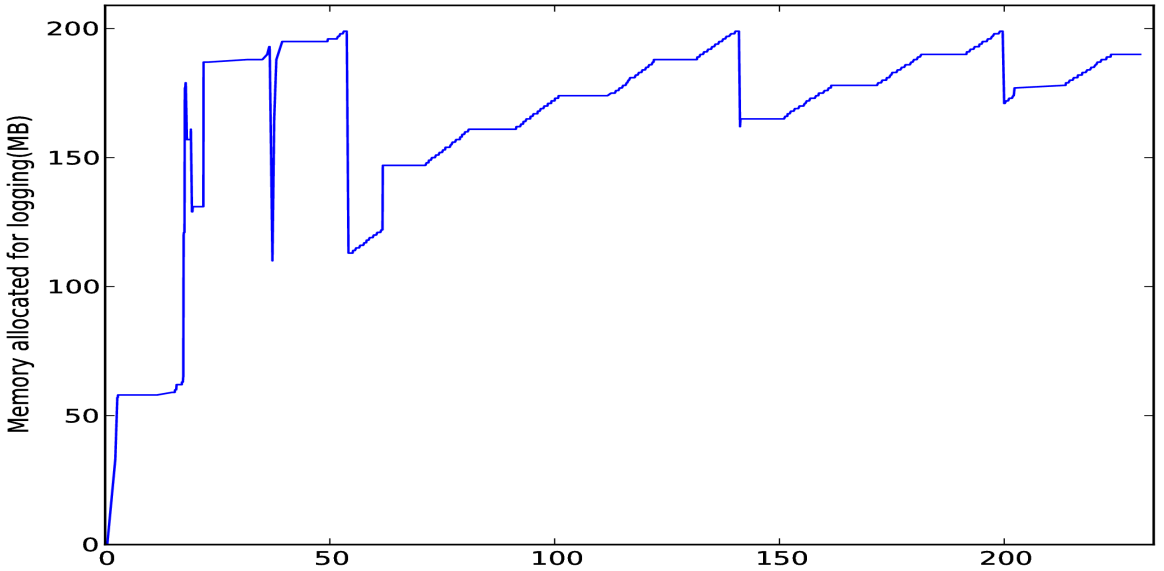
Rank 0
Total logged payload: 430MB



Rank 56
Total logged payload: 182MB

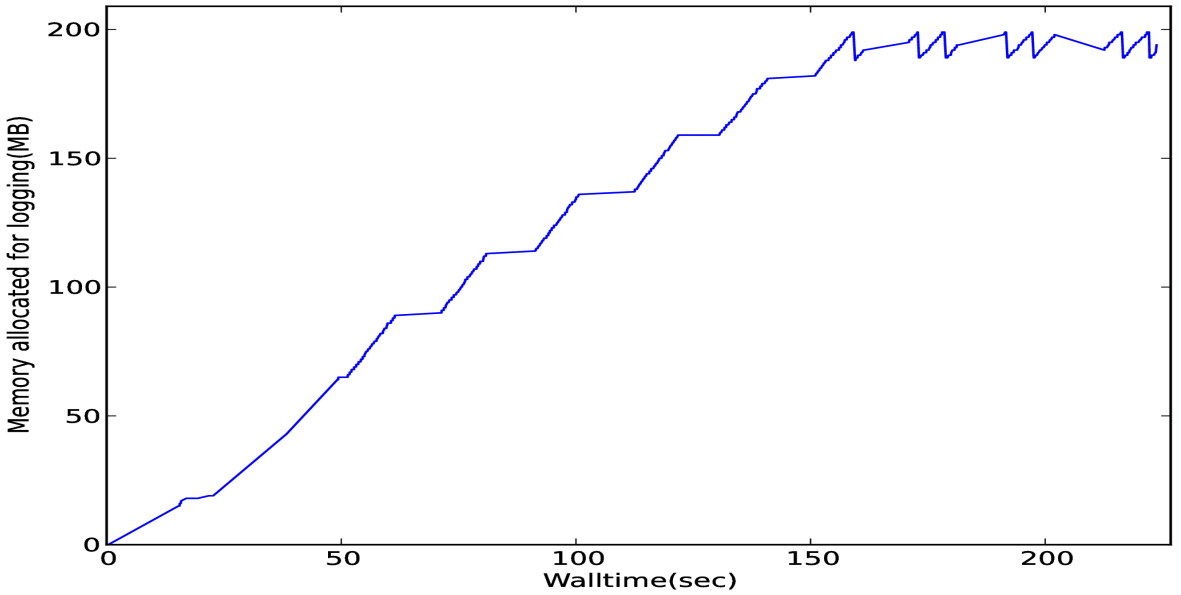
<u>Note</u>: besides logging the message payload, need memory to log determinants and for other bookkeeping stuff

Memory allocated for logging: Max 200MB



Rank 0

Total logged payload: 430MB



Rank 56

Total logged payload: 182MB

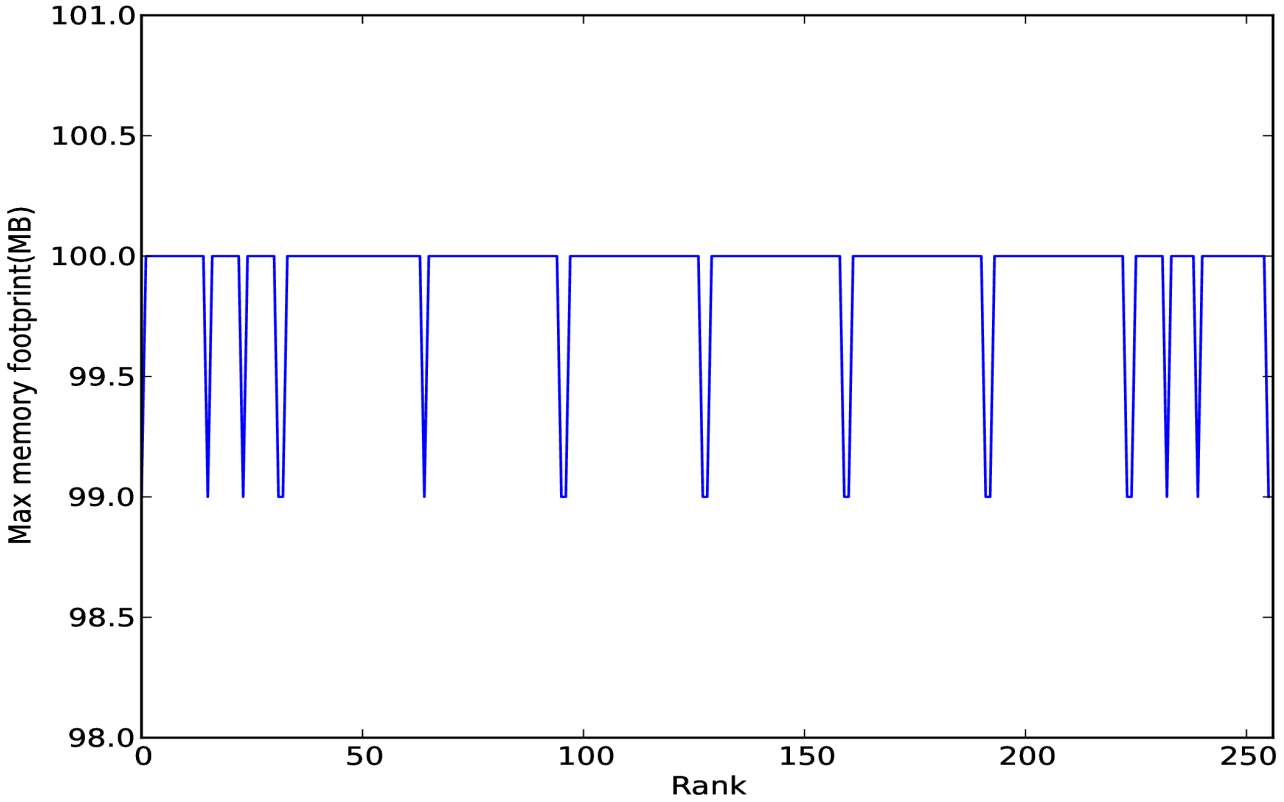
Memory limit vs Execution time

Mem_limit (MB)	Execution time (sec)	Total dumped (MB) (% of total logged)	# ranks who dumped
100	255	9398 (33%)	154
200	226	1665 (6%)	31
300	224	139 (0.05%)	2
400	222	30 (0.01%)	1
No limit	222	-	-

Dumping ~30% of logs to loggers' memory delayed execution by ~15%

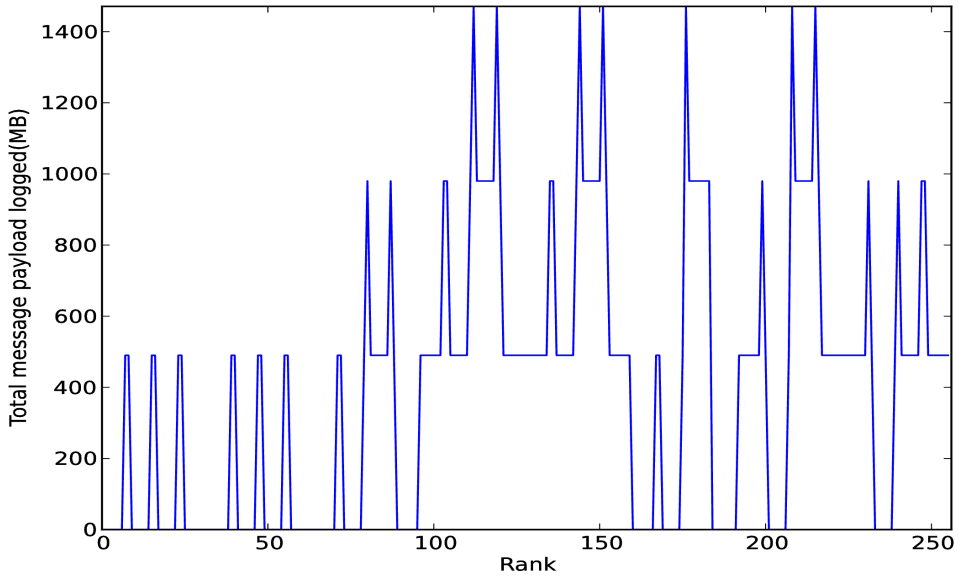
Case study 2

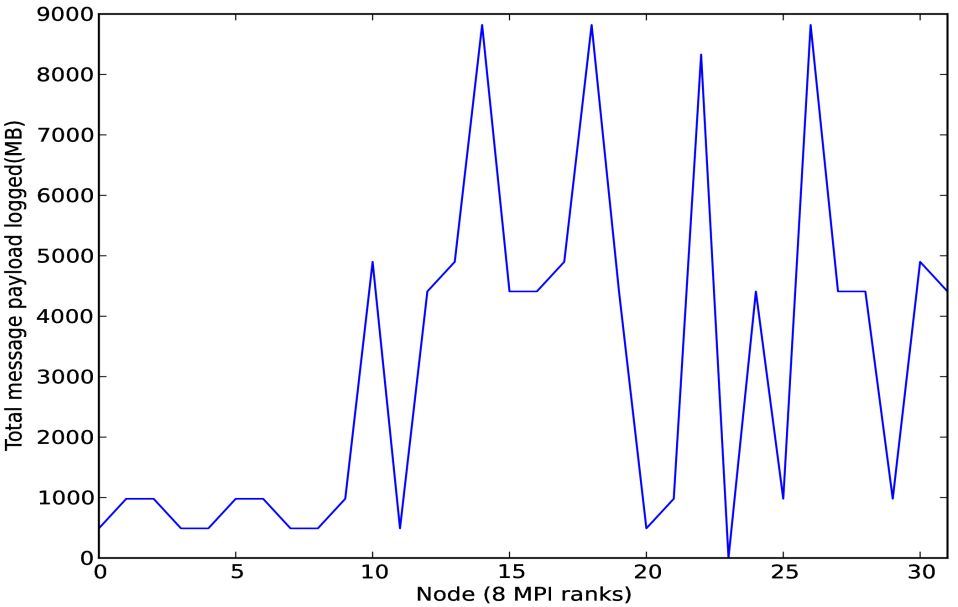
CM1: memory footprint



- **Simulate 6 mins (~4 mins of execution)**
- **Max memory: ~100MB**
- **Avg memory: ~99.9MB**

CM1: TOTAL LOG SIZE



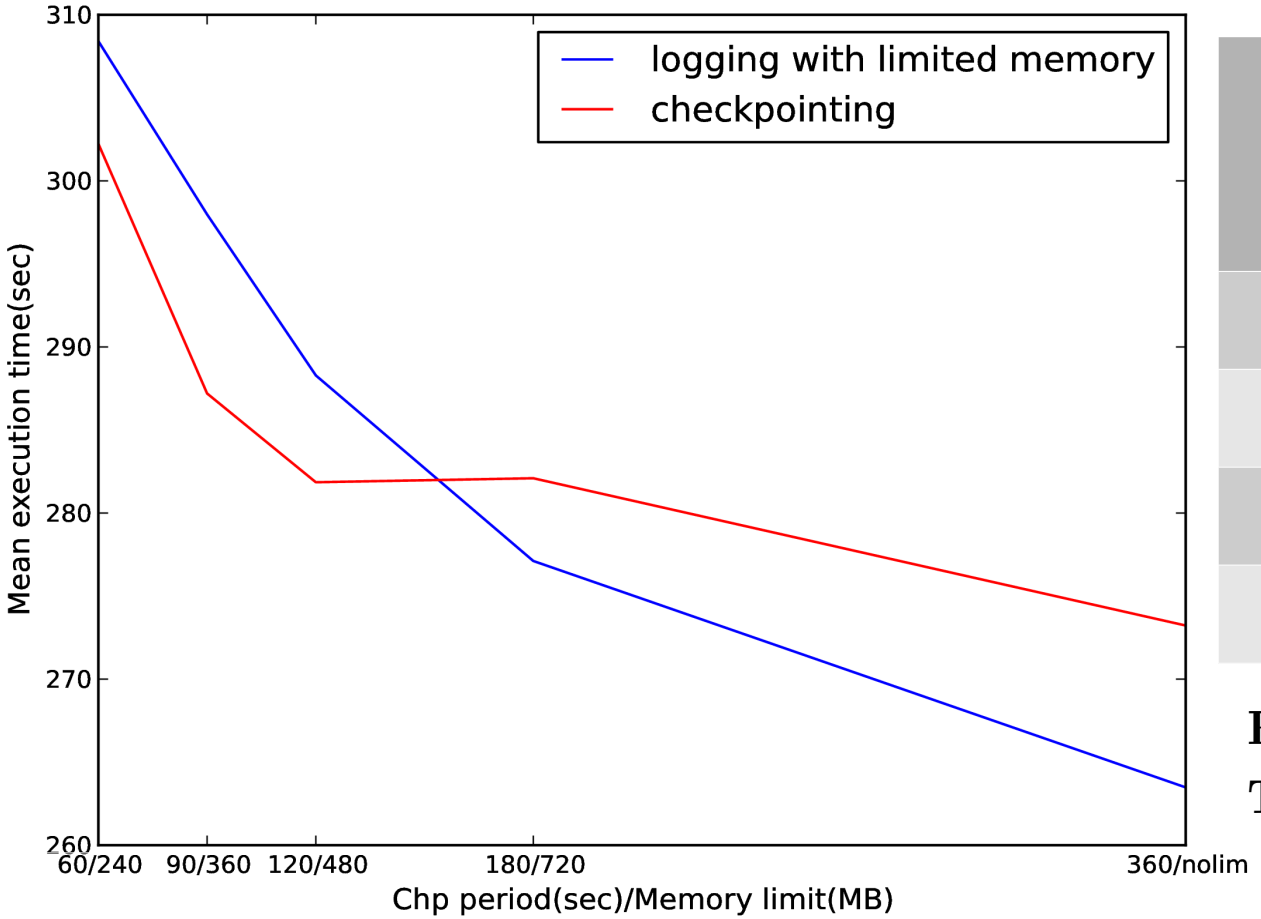


9 clusters (16-32 PEs per cluster)

Average log per rank: 394MB

- **Max log per rank: 1470MB**
- **Min log per rank: 0MB**

Message logging vs Checkpointing



Mem_limit (MB)	Total dumped (MB) (%of total logged)
240	64700 (64%)
360	46580 (46%)
480	28460 (28%)
720	15910 (15%)

Restart file size: ~6MB

Total written to PFS: ~1536MB

Conclusions

- **Caught between two fires:**
  - **App with small memory footprint but big log growth rate**
  - **App with large footprint but reasonable log growth rate (if ignore the init stage)**
- **Keep hope alive:**

sometimes still do better than just chp() frequently

What can be done

- **Intensive communication during initialization stage**
  - → **Chp() after the initialization**
- **Find optimal (chp period, memory limit)**
- **Change clustering ?**
  - – **Graph partitioning algorithm that minimizes maximum log size (per rank) ?**