Joint Laboratory
for Petascale Computation

On the feasibility of message logging in hybrid hierarchical FT protocols

Tatiana V. Martsinkevich, Franck Cappello

UNIVERSITÉ PARIS-SUD 11

informatics mathematics

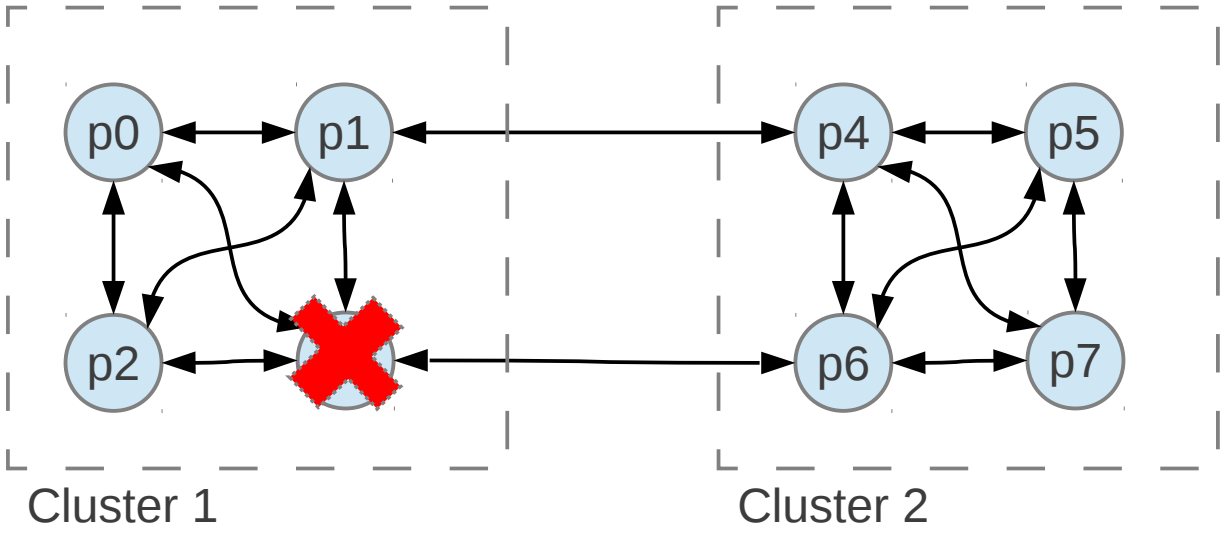
NCSA

FT protocols for large scale

- Number of cores on one CPU and number of CPU grows
- Can expect frequent hardware failures
- What fault tolerance protocol to use in large scale systems?
  - Checkpoint/restart, message logging, etc. protocols don't scale well as is
- For message passing applications hybrid protocols are the most promising
  - Hierarchical rollback-recovery protocols

Hybrid hierarchical FT protocol

- Divide processes into clusters
- Coordinated checkpointing inside the cluster
- Message logging for inter-cluster communication



Is message logging feasible?

- How much memory is available for logging?
- What if there is not enough memory?

Joint Laboratory for Petascale Computation

Memory requirenments of scientific applications

- Barcelona Supercomputing Center*

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

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

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

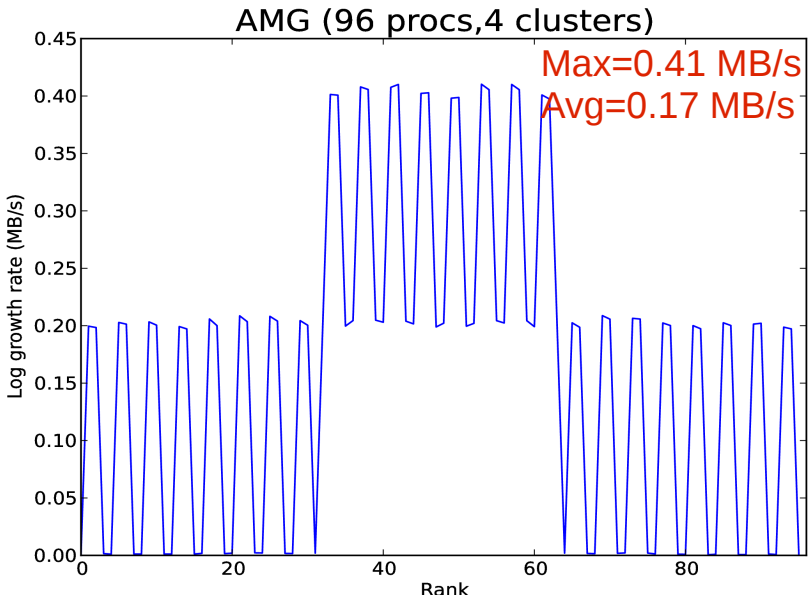
Memory requirenments of scientific applications(2)

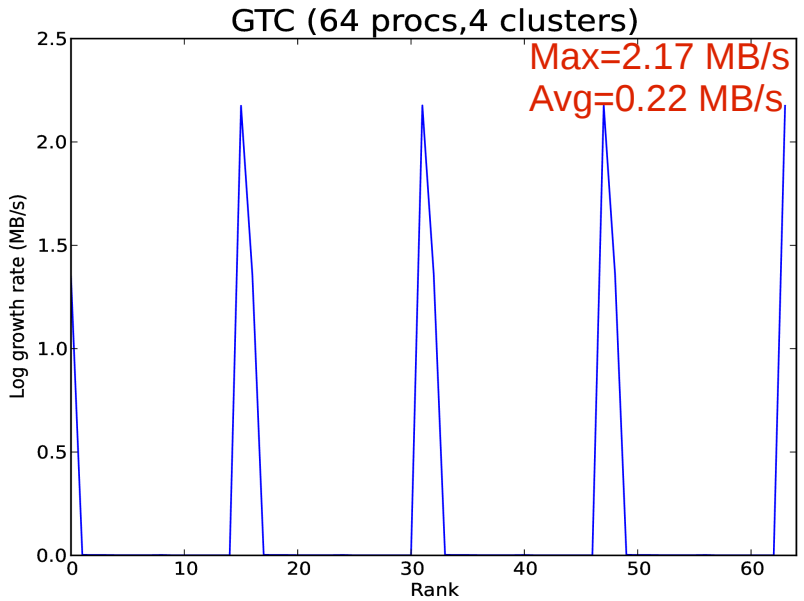
- NERSC-8 mini-applications

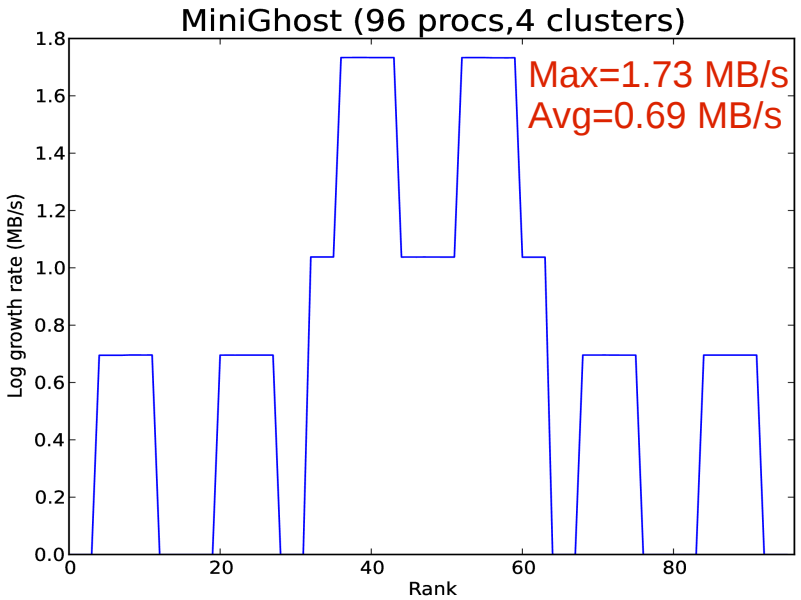
Application	nprocs	Total memory (GB)	Memory per-proc (GB)
AMG	96	100	1.04
AMG	49,152	51,200	1.04
AMG	960,000	1,000,000	1.04
GTC	64	32	0.50
GTC	19,200	10,240	0.53
MiniFe	144	96	0.67
MiniFe	49,152	32,768	0.67
MiniGhost	96	90	0.94
MiniGhost	49,152	47,104	0.96

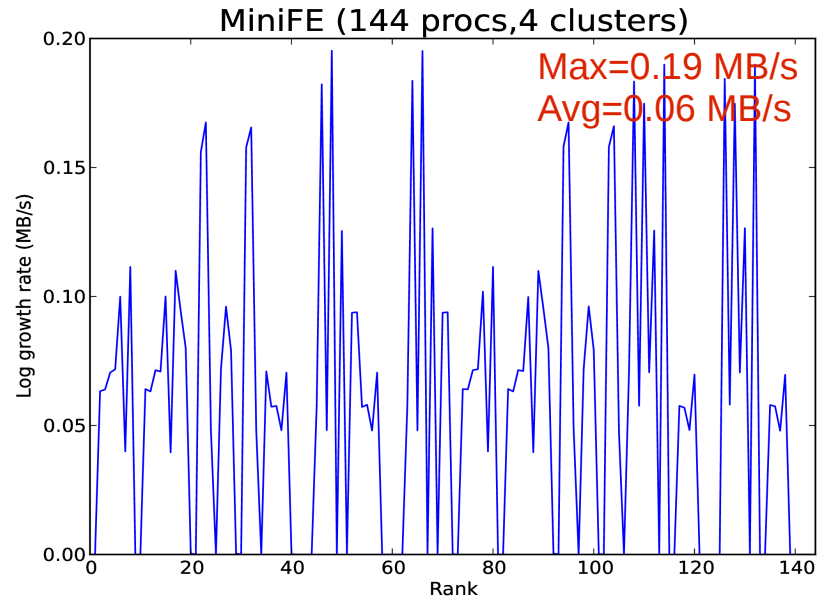
~0.5-1GB per process

Log growth rate









Clustering tool: T. Ropars, A. Guermouche, B. Ucar, E. Meneses, L. V. Kale, and F. Cappello. On the Use of Cluster-Based Partial Message Logging to Improve Fault Tolerance for MPI HPC Applications., *Euro-Par'11*

Log growth rate vs available memory

- Top 10 supercomputers from the top500 list
  - Average 1GB per core, 8-16GB per processor
- Example: GTC, MPI rank per core
  - Max log growth rate 2.17 MB/s
  - Assume memory quota for logs – 0.5GB per core

After ~4 mins will run out of memory

- Note: with OpenMP and shared memory will be better

When memory is not enough

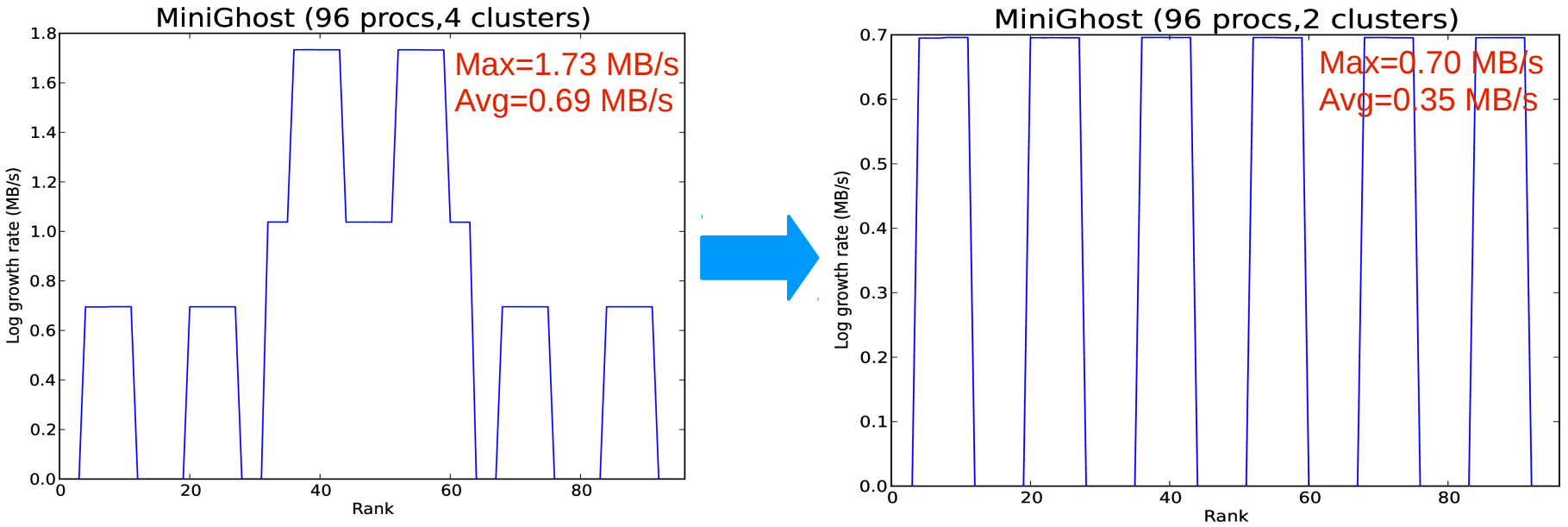
- Change checkpointing frequency in cluster
- Change clustering
- Flush logs

Strategy #1: Checkpointing frequency

- Many apps are self-synchronizing
  - → Assume chkp in the end of an iteration but the iteration is too long
  - → Different chkp frequency in clusters will introduce jitter
- This approach works if:
  - Log growth rate is even among clusters
  - Min period between chkp() < time to use up memory for logging

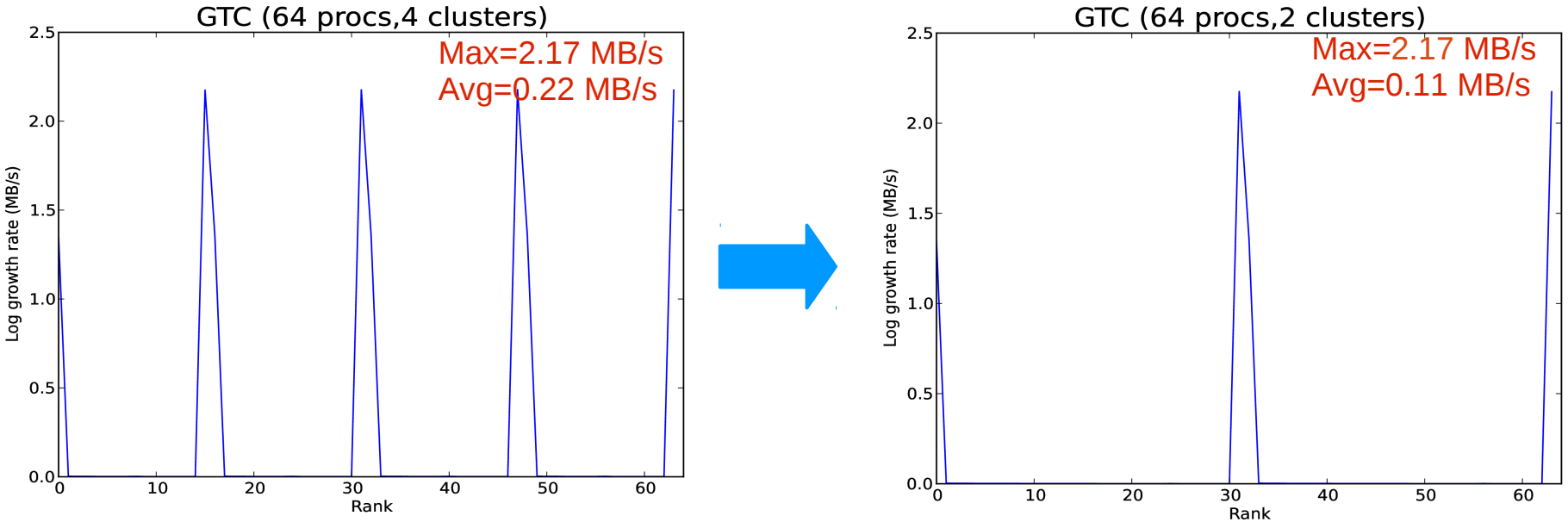
Strategy #2: Clustering

- Restart from the last chkp with new clustering
- Decreasing number of clusters
  - ➔Less data to log
  - ➔More computations to loose in failure



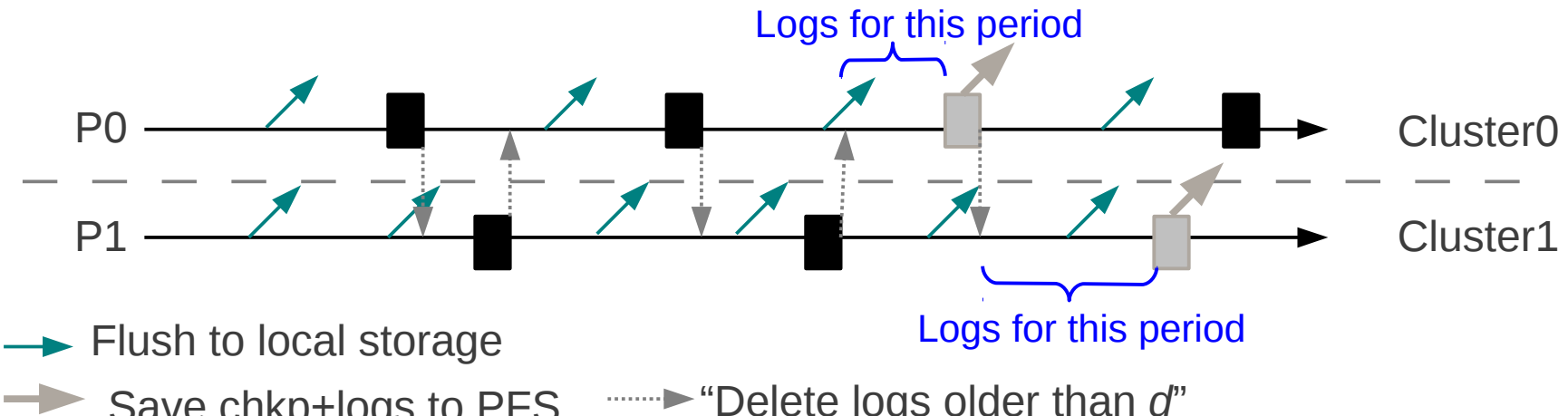
Clustering(2)

- Need clustering tool that not only minimizes average amount of logged data, but considers max log growth rate



Strategy #3: flushing logs

- Monitor log growth rate and schedule flushing?
- Decrease amount of logs to flush to the max
  - After chkp tell others to delete logs older than date *d*
  - Self-synchronizing apps eventually delete all logs
- Hierarchical approach?
  - Flush to local storage
  - Save to PFS with every *n*-th checkpoint



What to choose?

- Change checkpointing period so that it is the same for all procs
- Flushing logs when log growth rate is moderate
- Dynamical re-clustering for apps with regular communication pattern
- Combine re-clustering and log flushing?
  - With good clustering will need to additionaly save only small amount of data

Conclusions and future plans

- Memory requirements of some applications put constraints on message logging
- Several possible approaches but no universal approach

TODO:

- Study behaviour of apps on larger scale
  - Using of shared memory?
- New clustering tool?
  - more even log growth rate among clusters?

Joint Laboratory for Petascale Computation

Thank you

On the feasibility of message logging in hybrid hierarchical FT protocols

Tatiana V. Martsinkevich, Franck Cappello

UNIVERSITÉ PARIS-SUD 11

Inria informatics mathematics

NCSA