Reducing Waste in Extreme Scale Systems through Introspective Analysis

Leonardo Bautista-Gomez, Ana Gainaru, Swann Perarnau,
Devesh Tiwari, Saurabh Gupta,
Christian Engelmann, Franck Cappello and Marc Snir

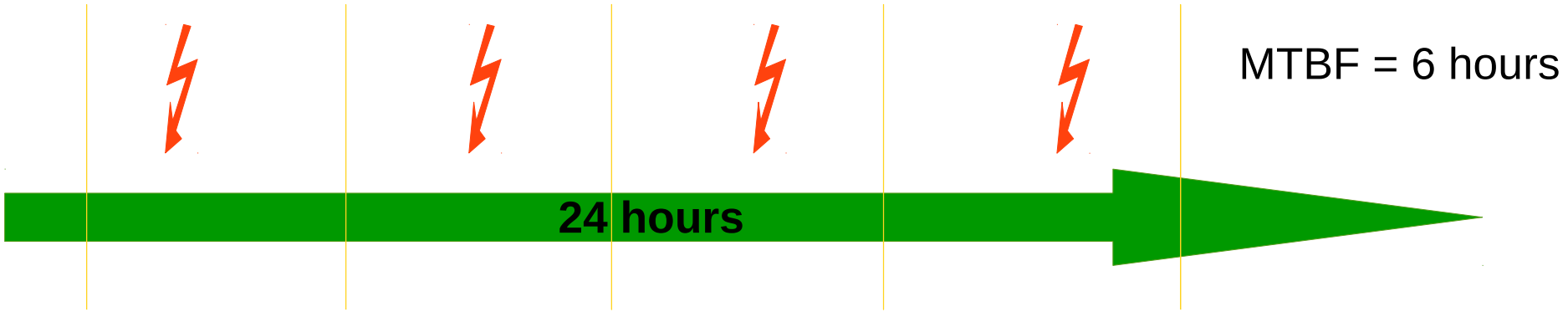
Outline

- Failure Analysis of Petascale Systems
- Monitoring and Reacting System
- Checkpointing Waste Reduction
- Conclusion

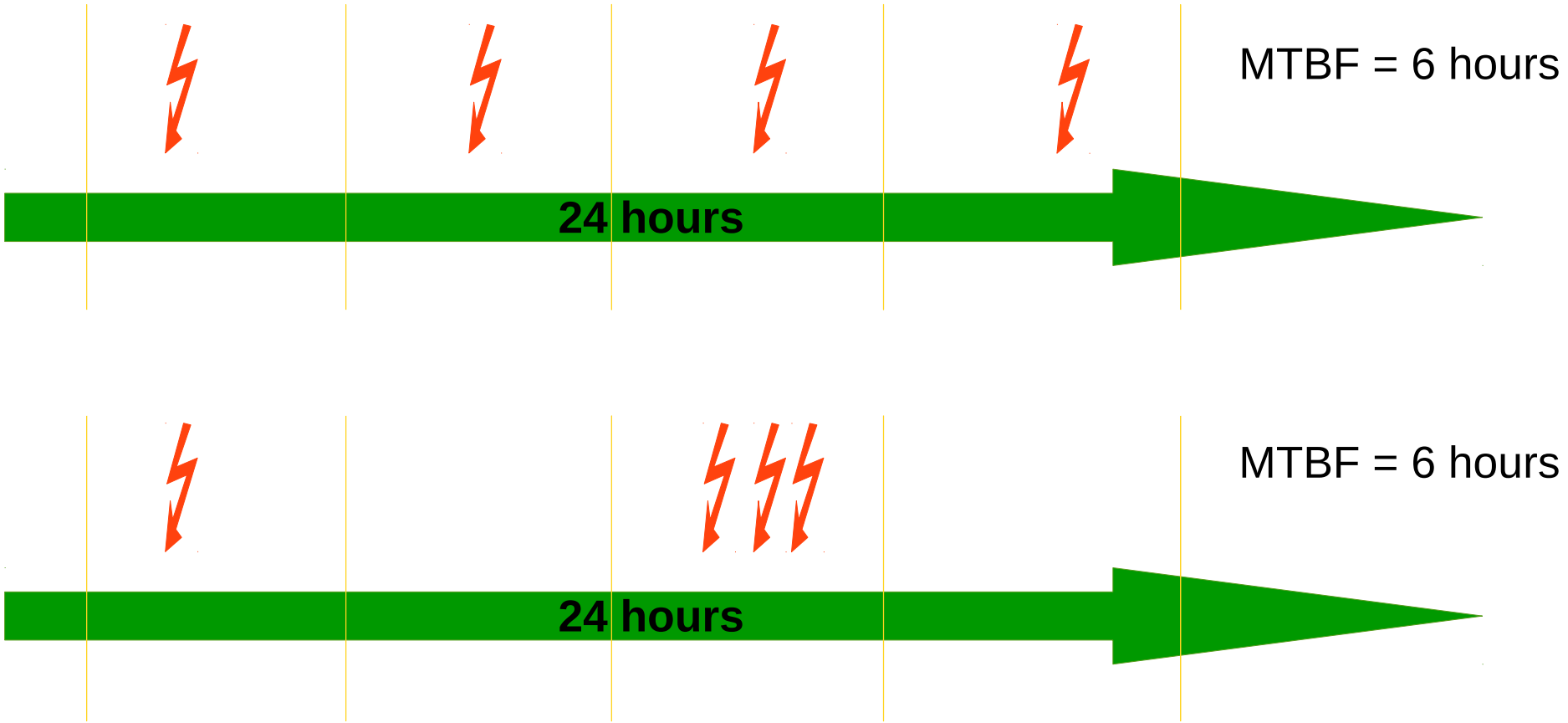
Outline

- **Failure Analysis of Petascale Systems**
- Monitoring and Reacting System
- Checkpointing Waste Reduction
- Conclusion

Failure Distribution



Failure Distribution



Defining Failure Regimes

$$px = \frac{x_i * 100}{\sum x_i} \qquad pf = \frac{f_i * 100}{\sum f_i}$$

Defining Failure Regimes

Number of segments with i failures

$$px = \frac{x_i * 100}{\sum x_i}$$

Percentage of time spent in regime i

Number of failures in x_i segments

$$pf = \frac{f_i * 100}{\sum f_i}$$

Percentage of failures occurring in regime i

Defining Failure Regimes

Number of segments with i failures

$$px = \frac{x_i * 100}{\sum x_i}$$

Percentage of time spent in regime i

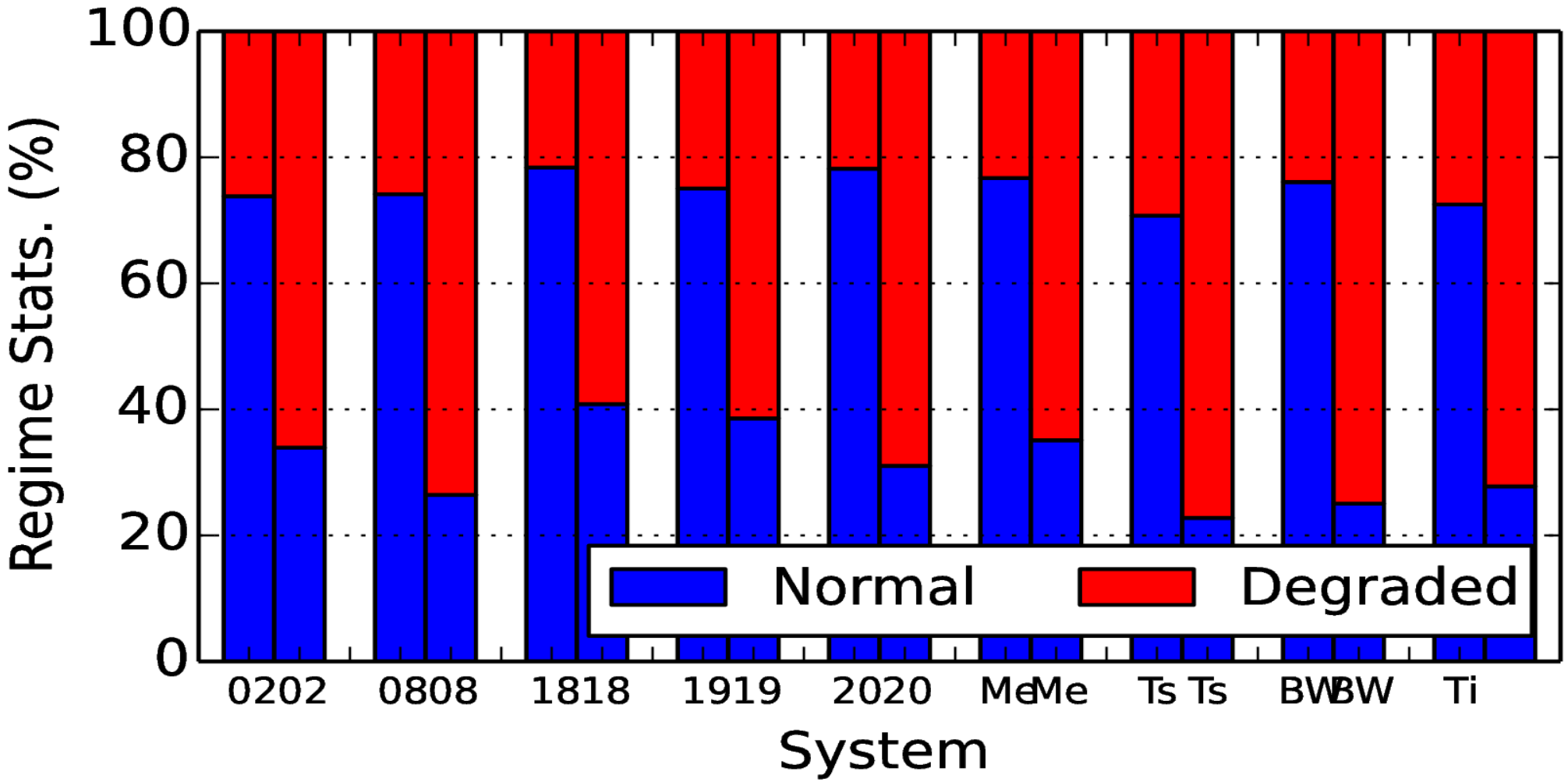
Number of failures in x_i segments

$$pf = \frac{f_i * 100}{\sum f_i}$$

Percentage of failures occurring in regime i

System	LANL02	LANL08	LANL18	LANL19	LANL20	Mercury	Tsubame 2	BlueWater	Titan
Normal reg. px	73.81	74.15	78.36	75.05	78.19	76.69	70.73	76.07	72.52
Normal reg. pf	33.92	26.42	40.84	38.58	31.05	35.10	22.78	25.05	27.77
Normal reg. pf/px	00.46	00.36	00.52	00.51	00.40	00.46	00.32	00.33	00.38
Degraded r. px	26.19	25.85	21.64	24.95	21.81	23.31	29.27	23.93	27.48
Degraded r. pf	66.08	73.58	59.16	61.42	68.95	64.90	77.22	74.95	72.23
Degraded r. pf/px	02.52	02.85	02.73	02.46	03.16	02.78	02.64	03.13	02.63

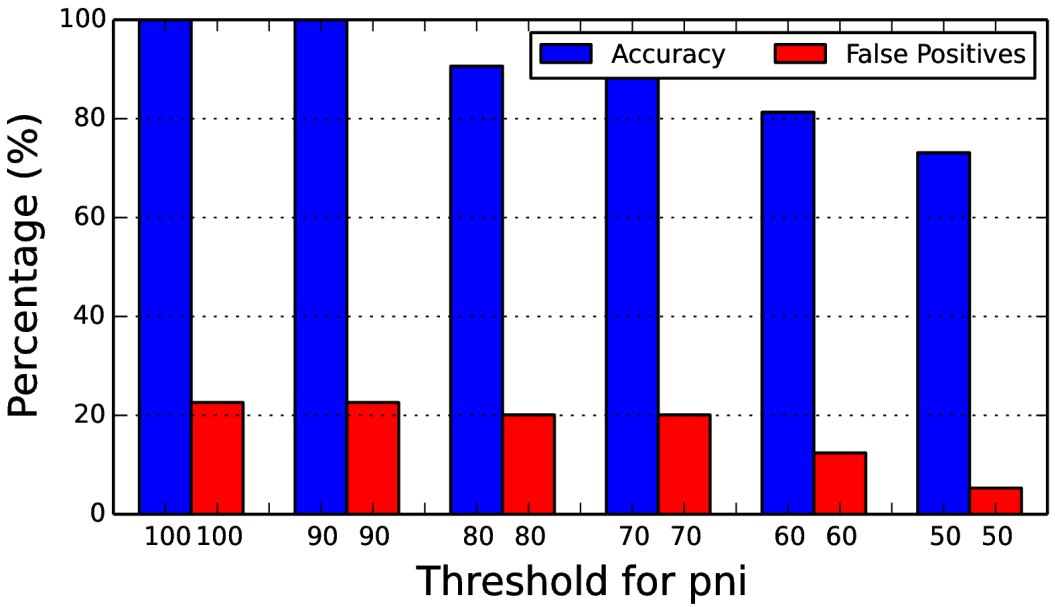
Failure Regime Statistics



Failure Regime Detection

- **Most** failures occur in degraded regime (Regime change?) .
- Some types of failures occur **only** in normal regimes.
- Pn_i = Probability of a type of failure to occur in normal regime.

Tsubame 2.5		LANL systems	
Failure type	pn_i	Failure type	pn_i
SysBrd	100%	Kernel	100%
GPU	55 %	Memory	61%
Switch	33 %	Fibre	100%
OtherSW	100.0 %	OS	49%
Disk	66 %	Disk	75%

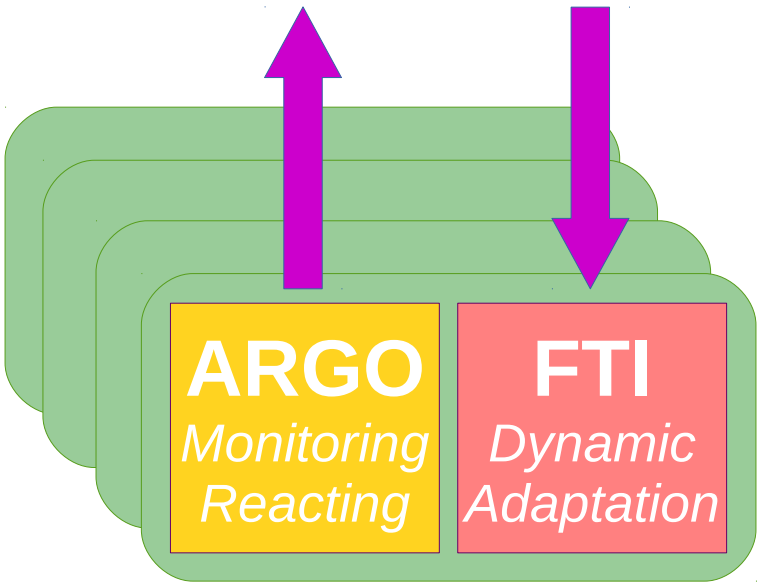


Outline

- Failure Analysis of Petascale Systems
- **Monitoring and Reacting System**
- Checkpointing Waste Reduction
- Conclusion

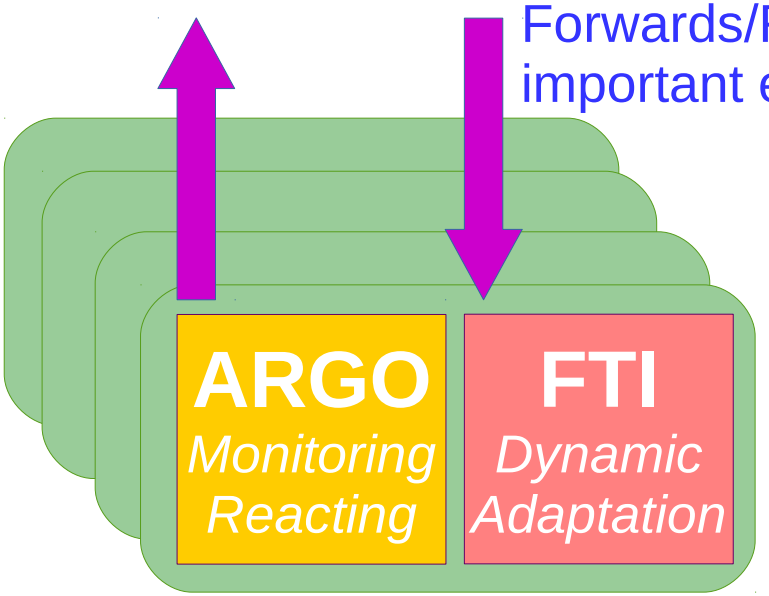
Monitoring and Reacting

Beacon Notification System (publish/subscribe)



Local monitoring and filtering

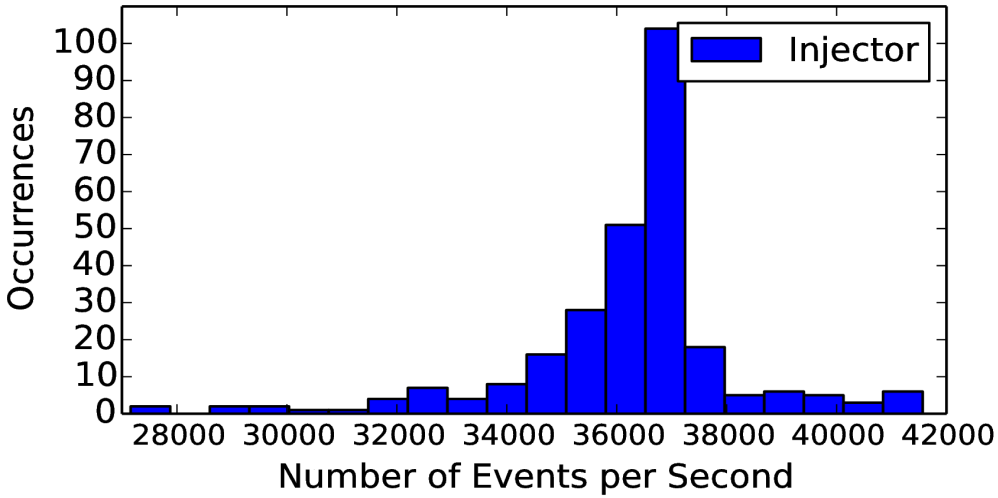
Forwards/Receives important event

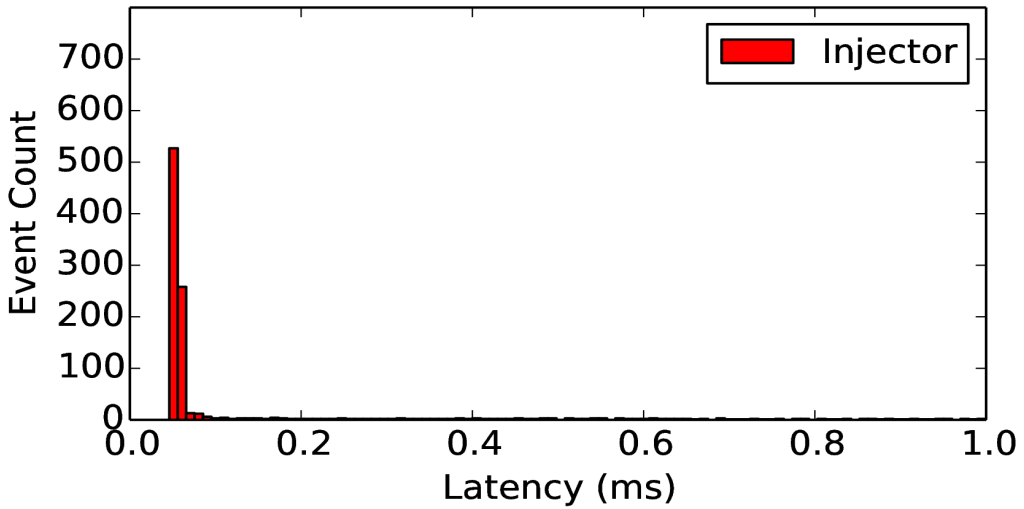


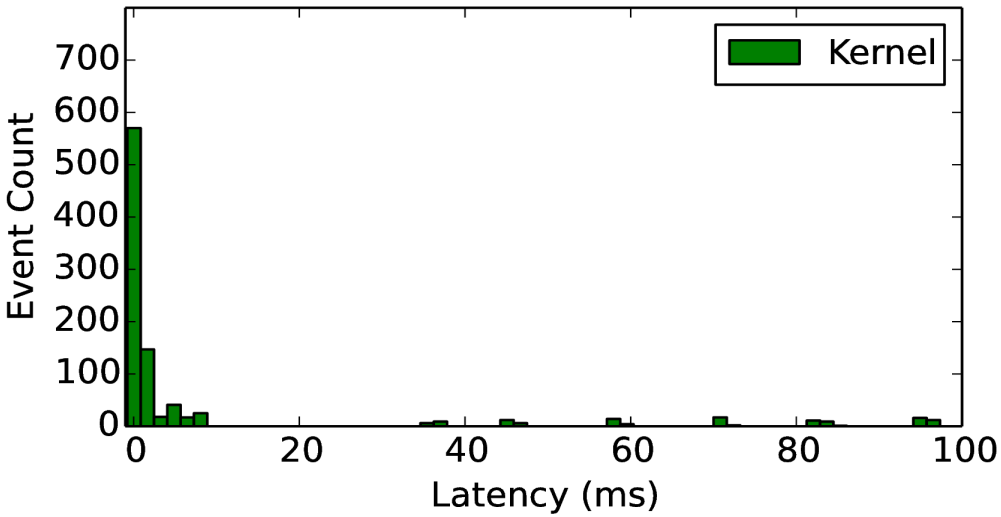
Checkpoint interval adapting

Event Traversal Latency

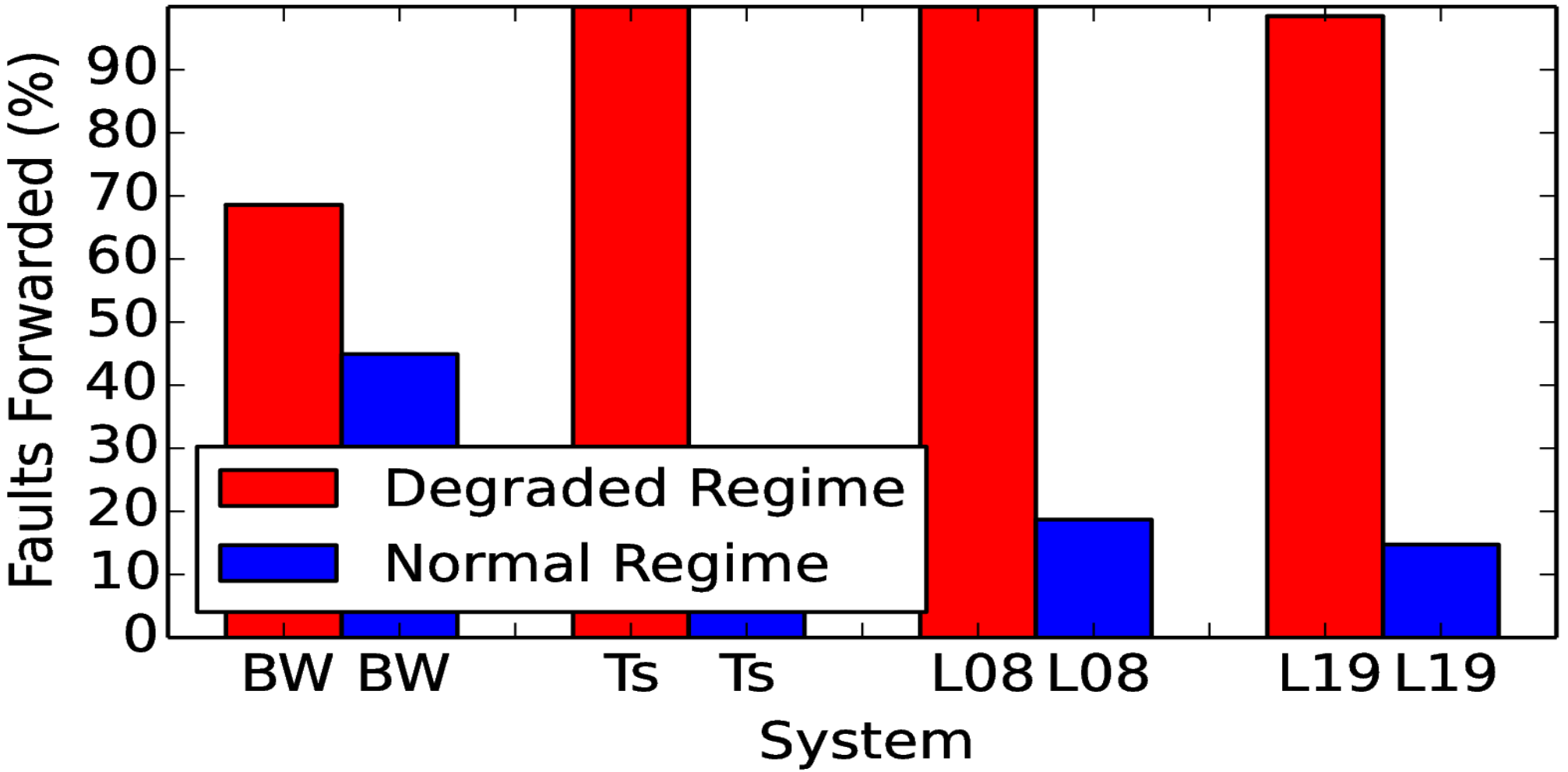
- Injecting a 1000 events burst.
- Most events analyzed in < 1ms.
- Event analysis << MTBF







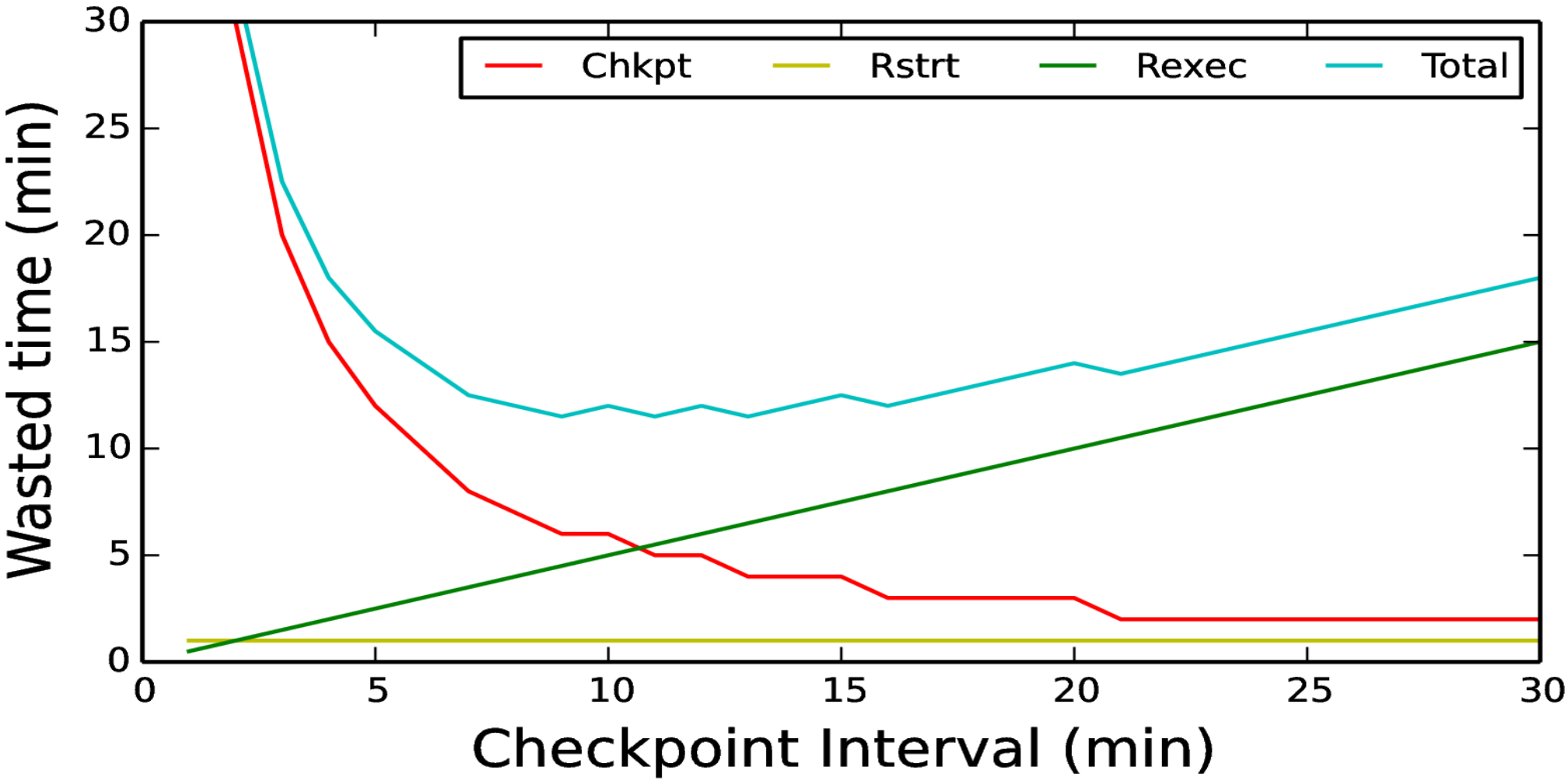
Events Forwarded



Outline

- Failure Analysis of Petascale Systems
- Monitoring and Reacting System
- **Checkpointing Waste Reduction**
- Conclusion

Checkpoint Overhead



Dynamic Checkpointing

Algorithm 1 Dynamic Checkpoint Interval

```
procedure FTI_SNAPSHOT
    addLastIterationLengthToList(IL)
    if updateGailIter == currentIter then
        GAIL = compute Global Average Iteration Length
        IterCkptInterval = wallClockCkptInterval/GAIL
        if updateRoof > expDecay*2 then
            expDecay = expDecay*2
        end if
        updateGailIter = currentIter + expDecay
    end if
    if nextCkptIter == currentIter then
        FTI_Checkpoint
        nextCkptIter = currentIter + IterCkptInterval
    else
        received = checkForNewNotifications(noti)
        if received then
            endRegimeIter, IterCkptInterval = decodeNotification(noti)
        end if
    end if
    if endRegimeIter == currentIter then
        IterCkptInterval = wallClockCkptInterval/GAIL
        endRegimeIter = -1
    end if
currentIter = currentIter+1
end procedure
```

Checkpoint Model

$$T_{waste}^{total} = \quad (Ck \; + Rt \; + Rx \;)$$

Checkpoint Model

$$T_{waste}^{total} = \sum_{i=1}^{R} (Ck_i + Rt_i + Rx_i)$$

Checkpoint Model

$$T_{waste}^{total} = \sum_{i=1}^{R} (Ck_i + Rt_i + Rx_i)$$

$$T_{waste}^{total} = \sum_{i=1}^{R} ((\frac{Ex \times px_i}{\alpha_i}) \times \beta \ + \frac{Ex \times px_i}{\alpha_i} (\mathrm{e}^{\frac{\alpha_i + \beta}{M_i}} - 1) \times (\epsilon(\alpha_i + \beta) + \gamma))$$

Checkpoint Model

$$T_{waste}^{total} = \sum_{i=1}^{R} (Ck_i + Rt_i + Rx_i)$$

$$T_{waste}^{total} = \sum_{i=1}^{R} \left(\left(\frac{Ex \times px_i}{\alpha_i} \right) \times \beta + \frac{Ex \times px_i}{\alpha_i} \left(e^{\frac{\alpha_i + \beta}{M_i}} - 1 \right) \times \left(\epsilon (\alpha_i + \beta) + \gamma \right) \right)$$

Total failure free execution time

Fraction of time spent in regime *i*

Average fraction of lost work

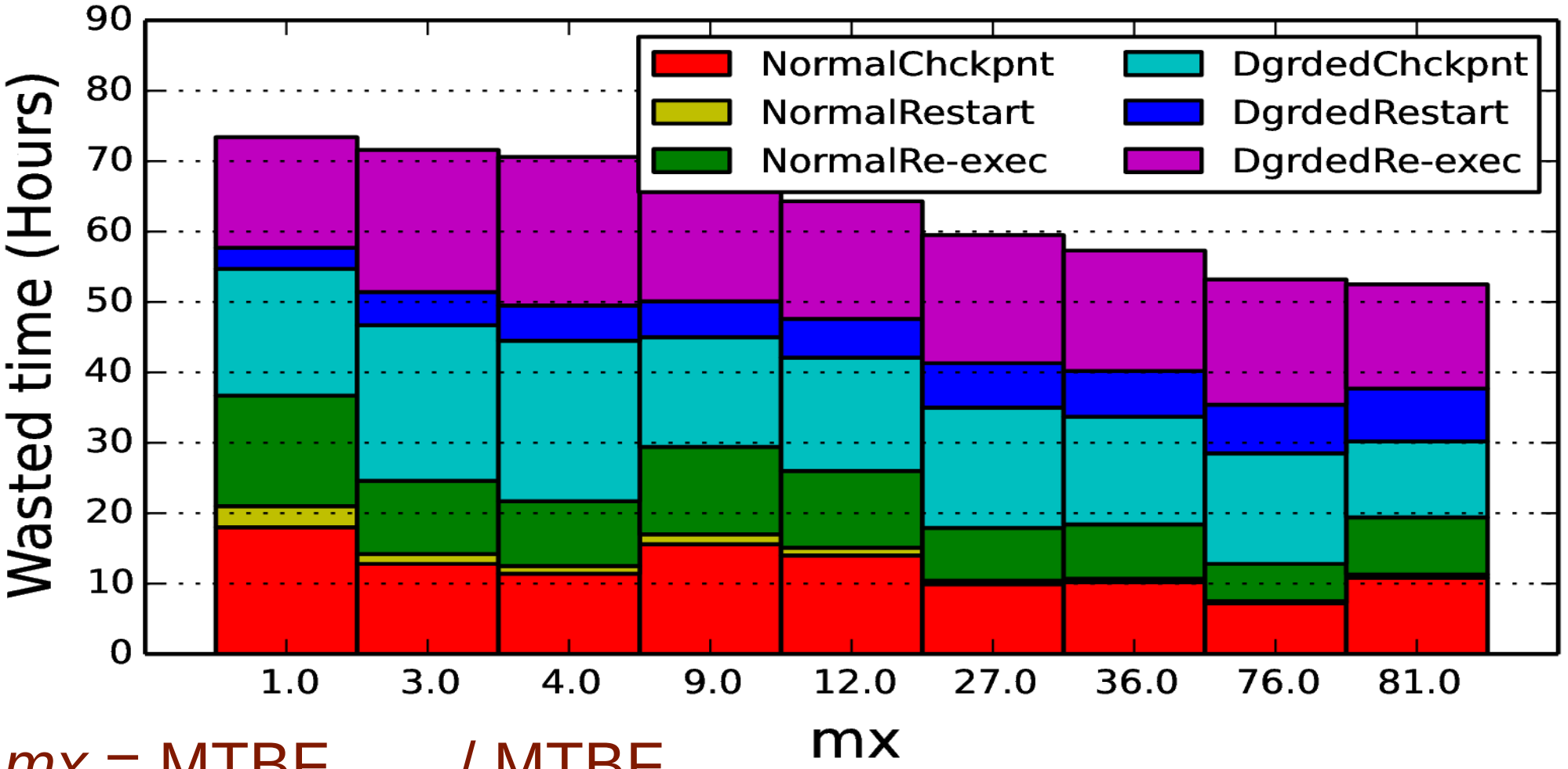
Checkpoint interval in regime *i*

Checkpoint cost

MTBF for regime *i*

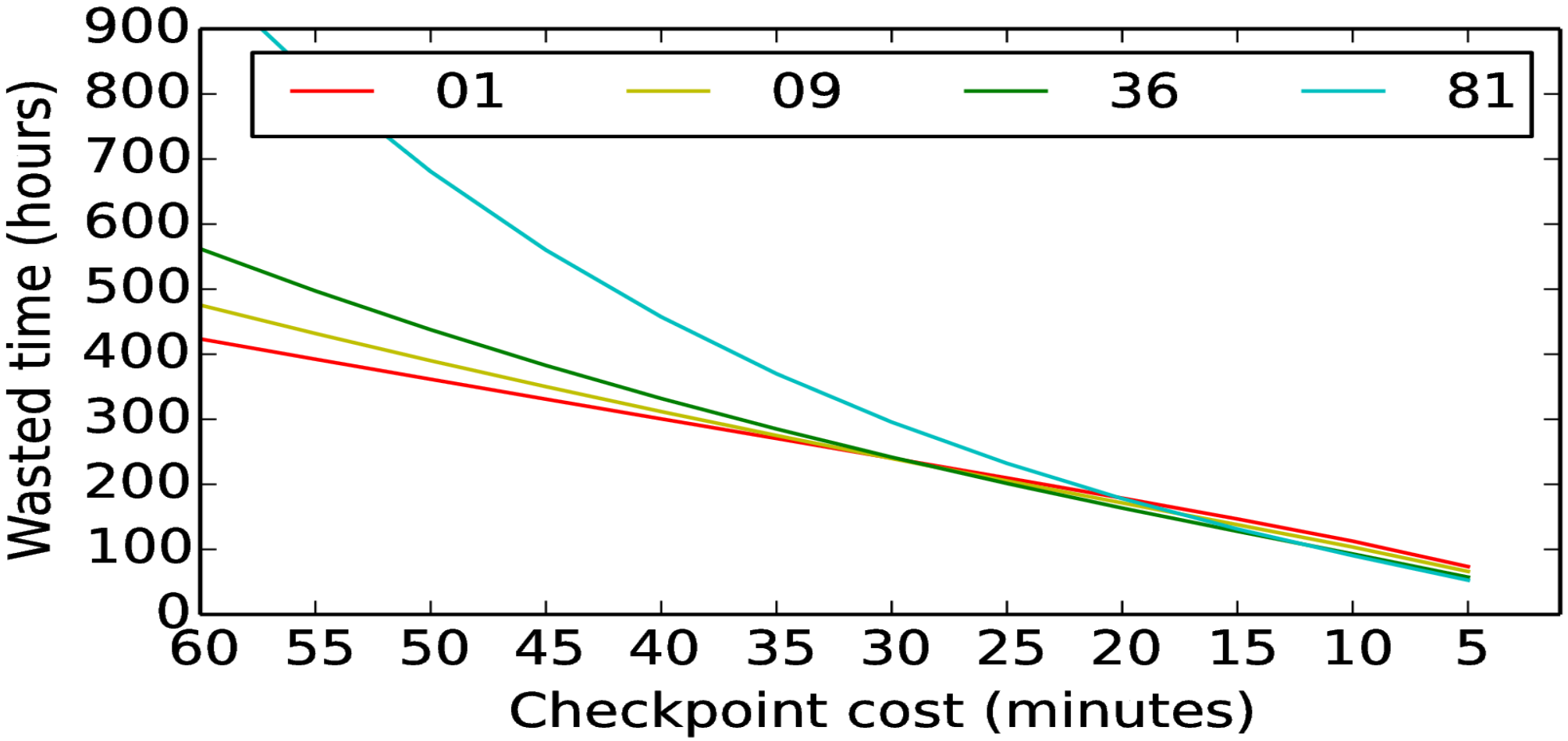
Restart cost

Waste Reduction



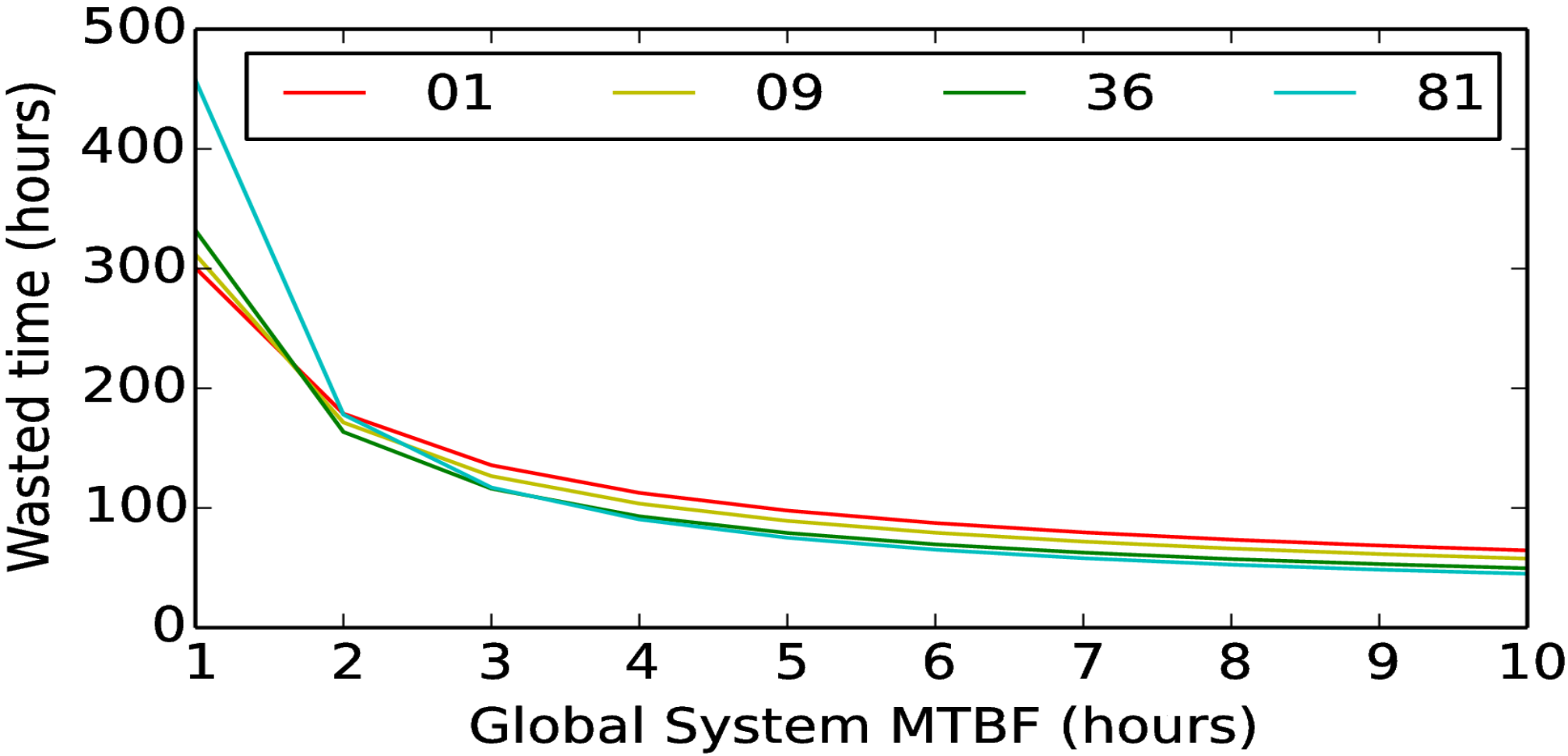
$mx = MTBF_{normal} / MTBF_{degraded}$

Waste for Different Ckpt.



Wasted time vs Checkpoint cost for systems with different *mx* (MTBF = 8 hours).

Waste for Different MTBF



Wasted time vs MTBF for systems with different *mx* (Checkpoint cost = 5 minutes).

Conclusions

- Most systems show some level of temporal failure correlation.
- Introspective Analysis can be done on-the-fly for a low cost.
- Significant gains can be achieved trough dynamic adaptation.

Thank you!

Questions?