An Online Controller Towards Self-Adaptive File System Availability and Performance

Xin Chen[1], Benjamin Eckart[1], Xubin He[1],

and

Christian Engelmann[2], Stephen Scott[2]

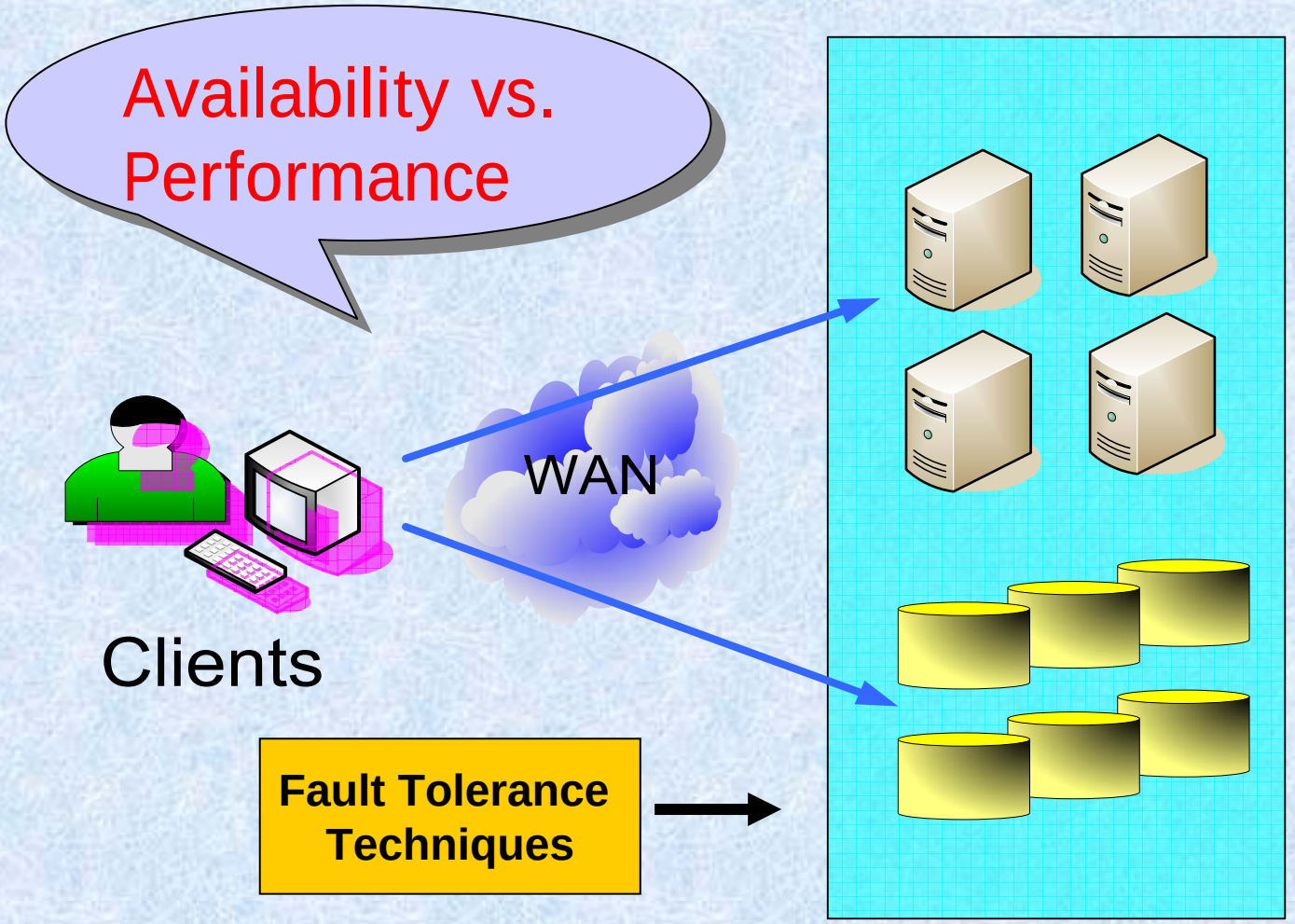
[1] Department of Electrical and Computer Engineering
Tennessee Technological University

[2] Computer Science and Mathematics Division
Oak Ridge National Laboratory

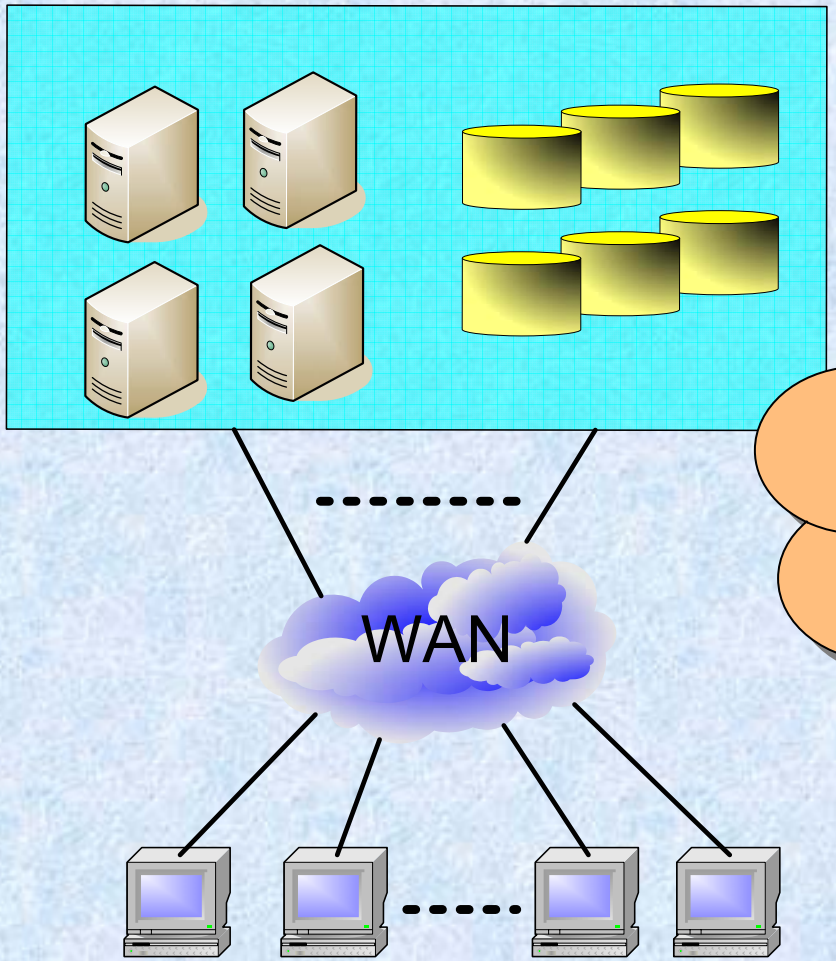
Outline

1. Introduction
   - Motivations
   - Contributions
2. Replication strategy
3. Mathematical models
   - Performance model
   - Availability model
4. Controller design
5. Conclusions and future work

Motivation – Demands for High Availability and Performance



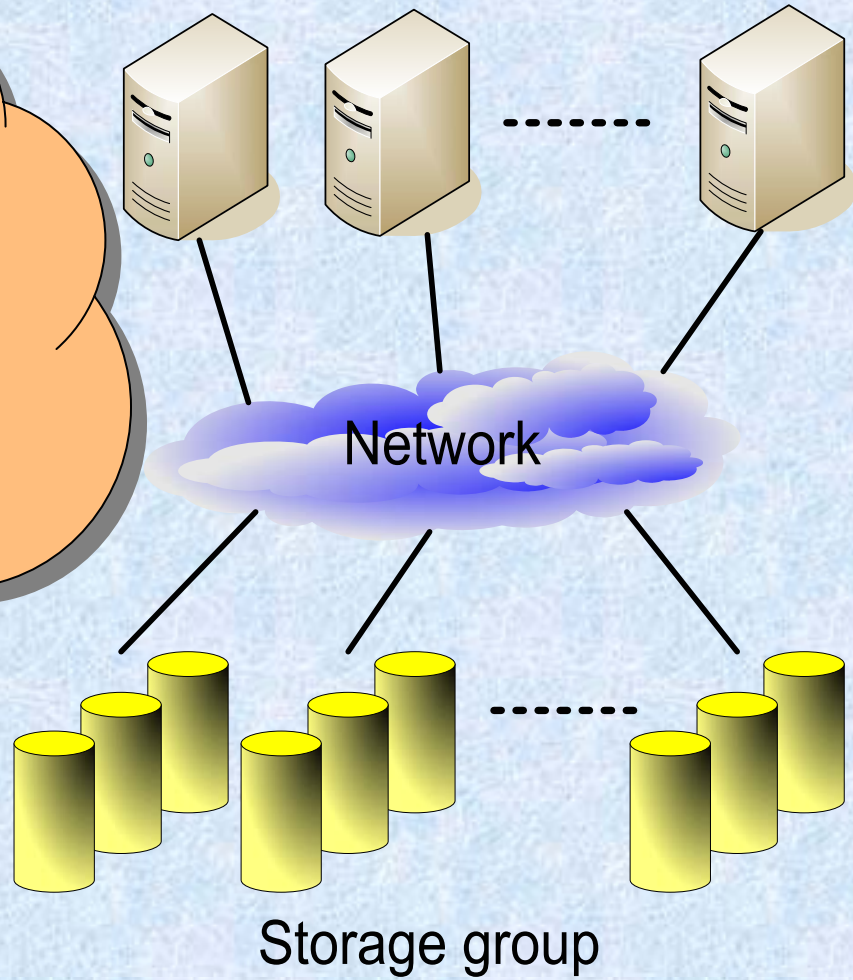
Motivation – A Strong correlation between workloads and failure rate



Recent study [1] on failures in peta-scale file systems concludes that a higher system workload has a strong correlation to a higher failure rate.

Motivation – The importance of online controlling

With the increasing size and complexity of large scale file systems, manually tuning a system to achieve optimal online system availability and performance is impractical, difficult, and most likely error-prone [2].



Contributions

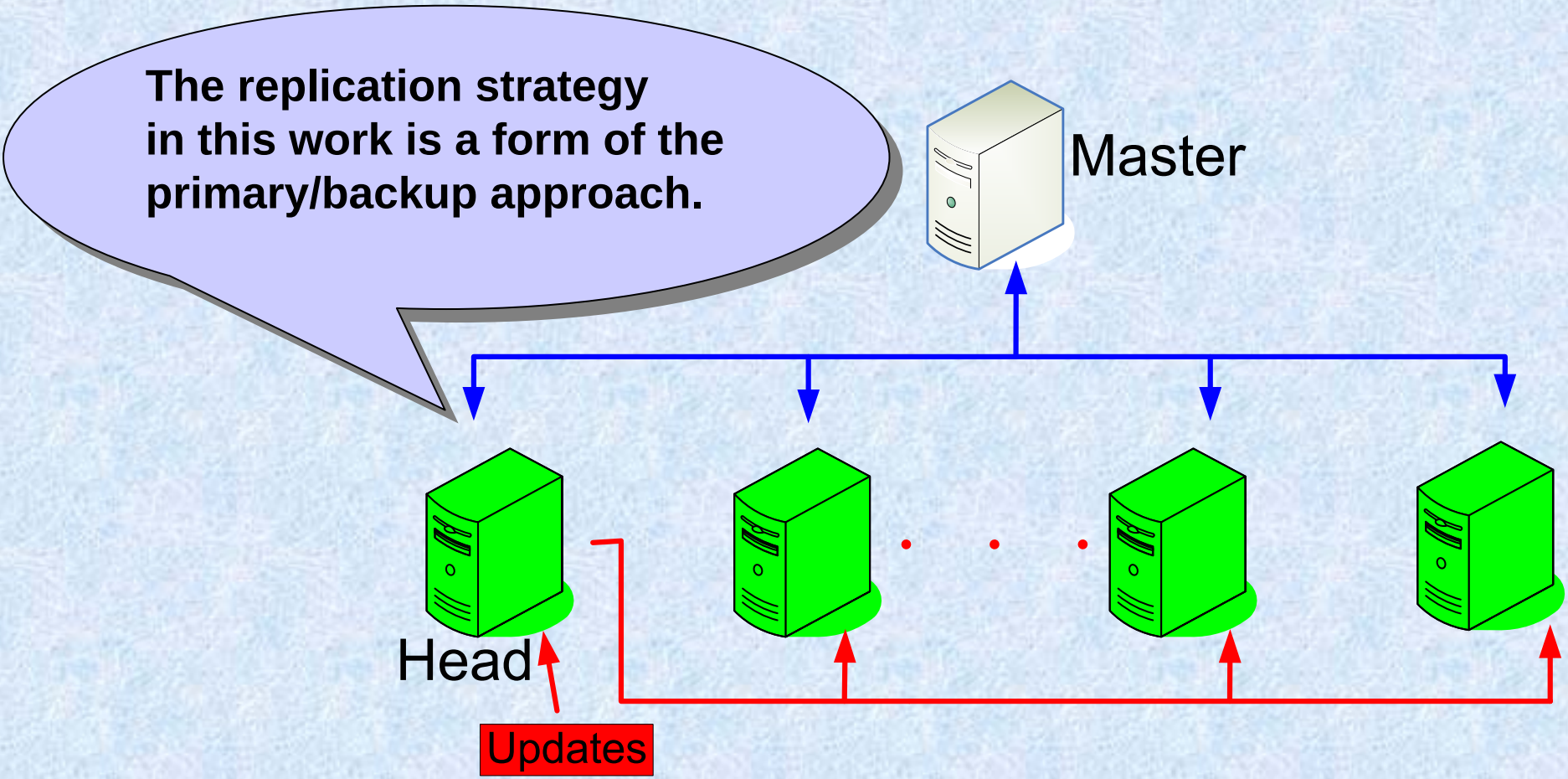
- Propose two mathematical models based on a replication strategy in a distributed file system to explore the correlation among availability, performance, and workloads.

- Propose an online controller that will help system achieve a runtime optimal performance and availability via dynamically tuning the system replication policy.

Outline

1. Introduction
   - Motivations
   - Contributions
2. Replication strategy
3. Mathematical models
   - Performance model
   - Availability model
4. Controller design
5. Conclusions and future work

Replication Strategy

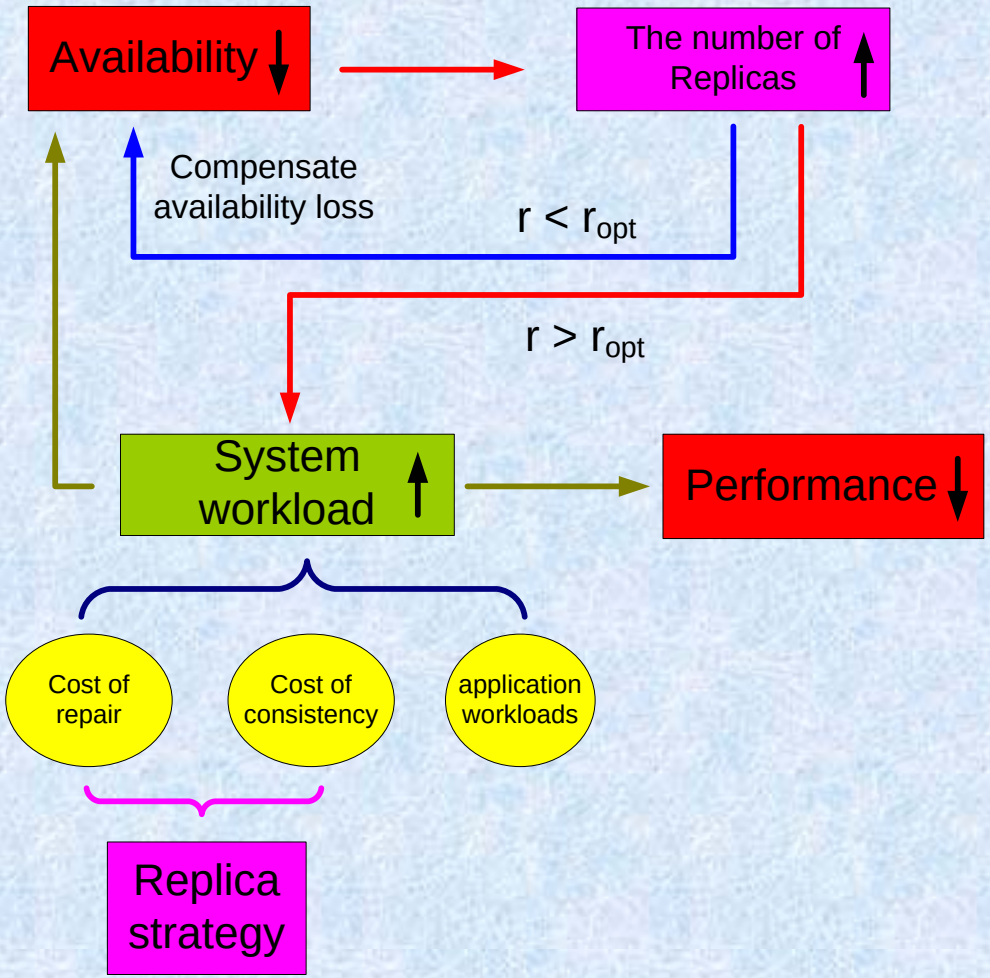


Outline

1. Introduction
   - Motivations
   - Contributions
2. Replication strategy
3. Mathematical models
   - Performance model
   - Availability model
4. Controller design
5. Conclusions and future work

The Analytical Model

- $r < r_{opt}$
- $r > r_{opt}$



Two Questions

- How much performance will be sacrificed as a result of increasing the number of replicas?

- What's the runtime system availability given a replica number *r* and workloads?

Assumptions

- The probability of a node to be unavailable is treated as an independent variable;
- System workloads only consist of update operations in this model;
- The system is fail-stop;
- The resource of the system, in terms of number of nodes, is constant.

Performance Model

In this model, we concentrate on the effect of the number of replicas on the system performance with update intensive system workloads, and conclude that the system will suffer great performance degradation with more replicas in some cases given the resource constraint.

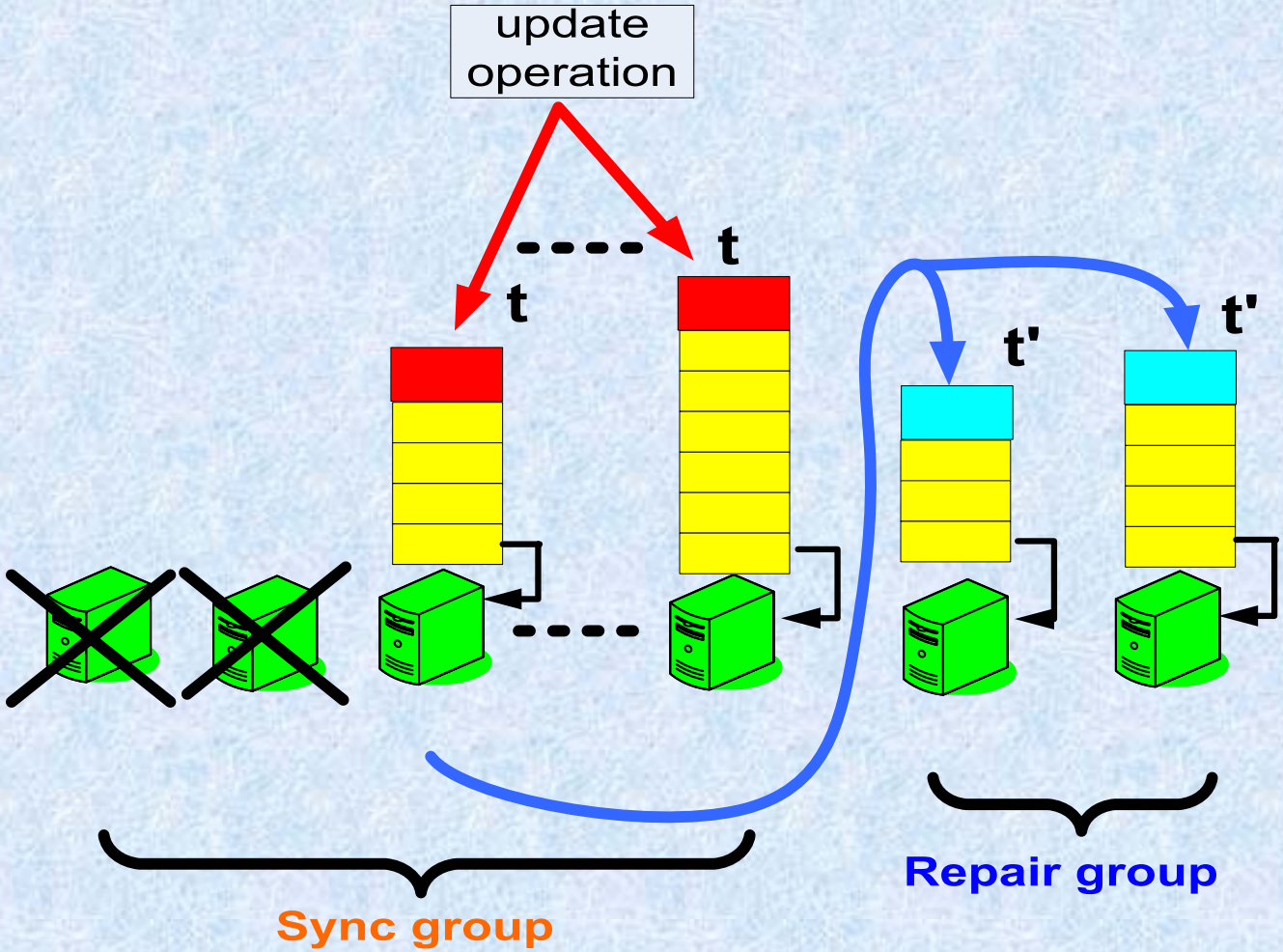
Performance Model -- Definitions

- $C_r(i)$: denotes the cost of repair overhead;
- $C_s(i)$: denotes the cost of synchronization overhead;
- $C(i)$: denotes the cost of an update operation i on the head node;
- $C_{total}(i)$: denotes the total cost of an update operation i in the system;

$$C_{total}(i) = C(i) + C_s(i) + C_r(i) \quad (1)$$

$$C_s(i) = t_{max} \in \{t_1, t_2, ..., t_n\}, (n \in \{1, 2, ..., r\})$$

$$C_r(i) = t_{max} \in \{t'_1, t'_2, ..., t'_m\}, (m \in \{1, 2, ..., r-1\})$$



Analysis of ba

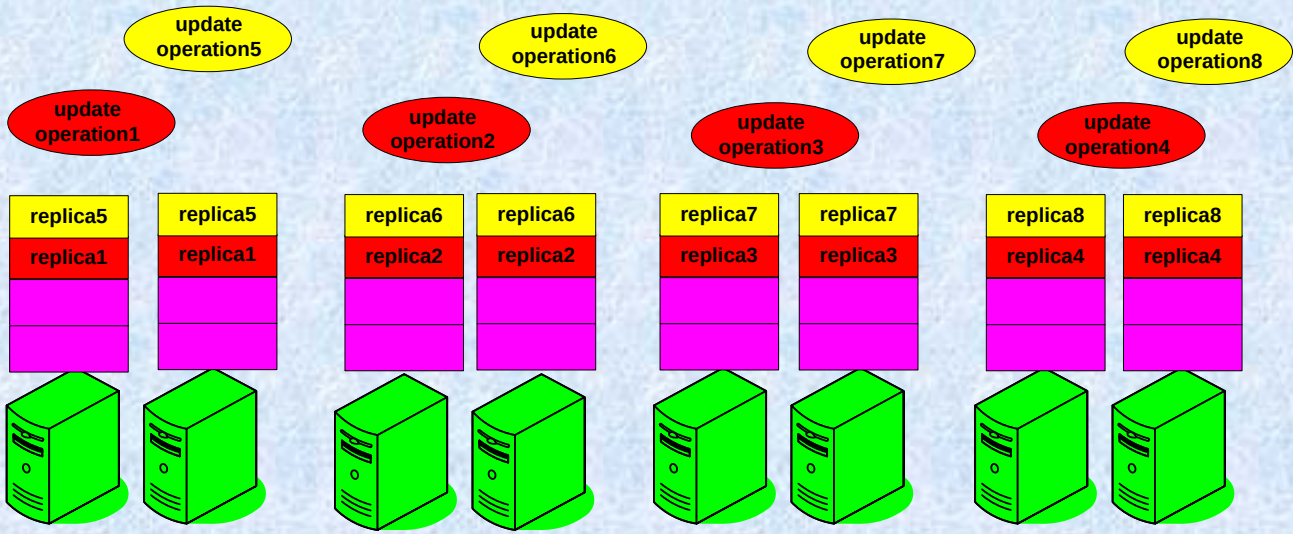
$$C = \beta + max(C(1), C(2), ..., C(8)) \quad (2)$$
$$= \beta + \alpha$$

Eight update operations are issued simultaneously;

Only eight nodes are in the system;

There are two replicas associated with each operation;

No failure occurs during all updates in this case.



$$C' = \beta + max(C(1), ..., C(4)) + max(C(5), ..., C(8)) \quad (3)$$
$$= \beta + \alpha + \alpha = \beta + 2\alpha$$

Relative execution time increase percentage

$$C = \beta + max(C(1), C(2), ..., C(8)) \quad (2)$$
$$= \beta + \alpha$$
$$C' = \beta + max(C(1), ..., C(4)) + max(C(5), ..., C(8)) \quad (3)$$
$$= \beta + \alpha + \alpha = \beta + 2\alpha$$

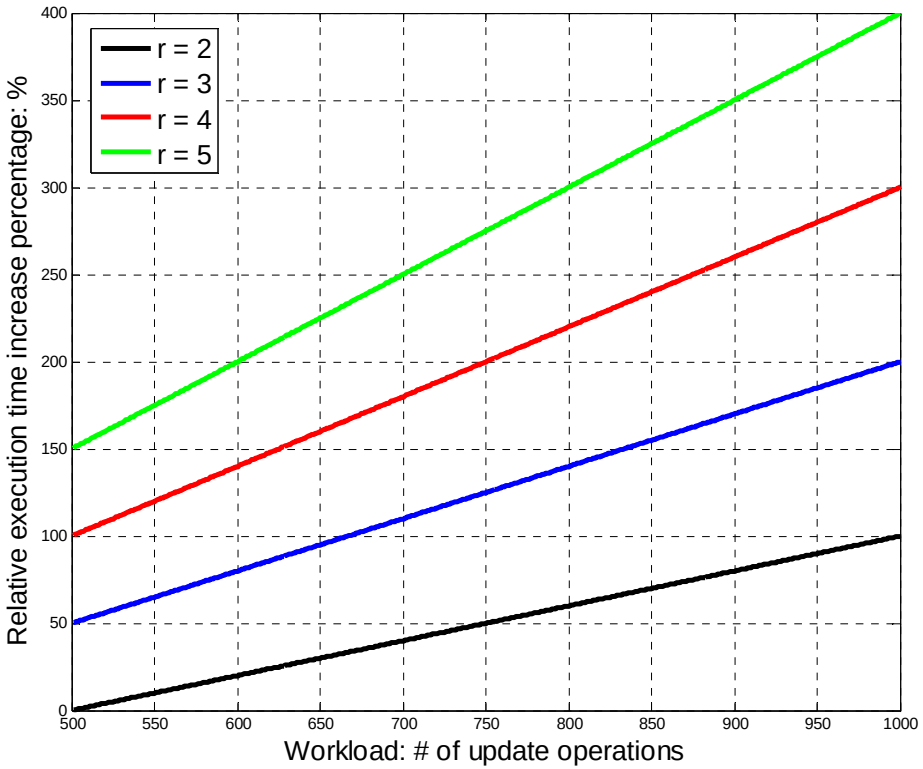
$$\gamma = \frac{C' - C}{C} \times 100\% \quad (4)$$
$$= \frac{\alpha}{\beta + \alpha} \times 100\%$$

n : the number of nodes in the system;

r: the number of replicas;

w: the number of update operations

$$\Gamma(w,r) = \lceil \frac{r \times w}{n} \rceil - 1, (\beta = 0)$$



Summary

- **System performance incurs significant degradation with more replicas in a period of high updating activities;**

Availability Model

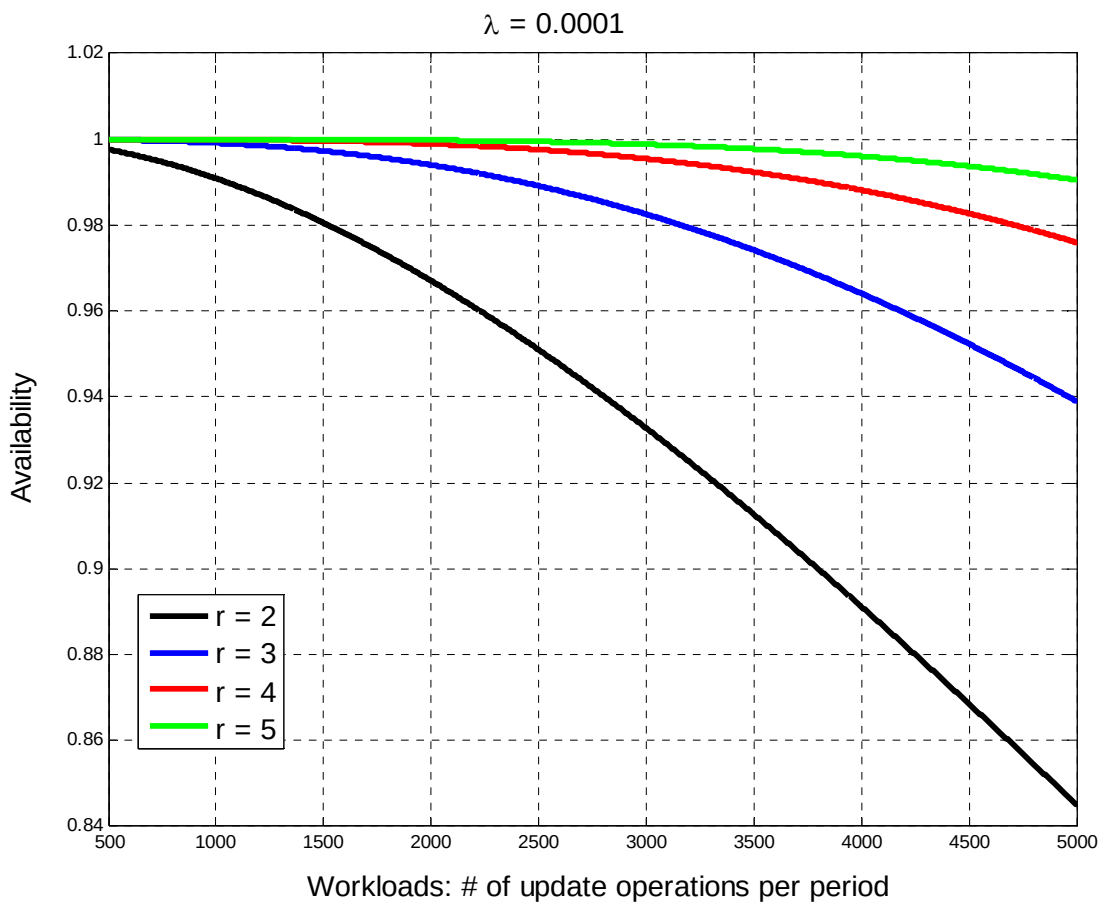
- A widely used failure growth model, Goel-Okumoto model [3] assumes that the failure arrival process is a non-homogeneous *Poisson* process: the failures experienced by time t follows a Poisson distribution [4].

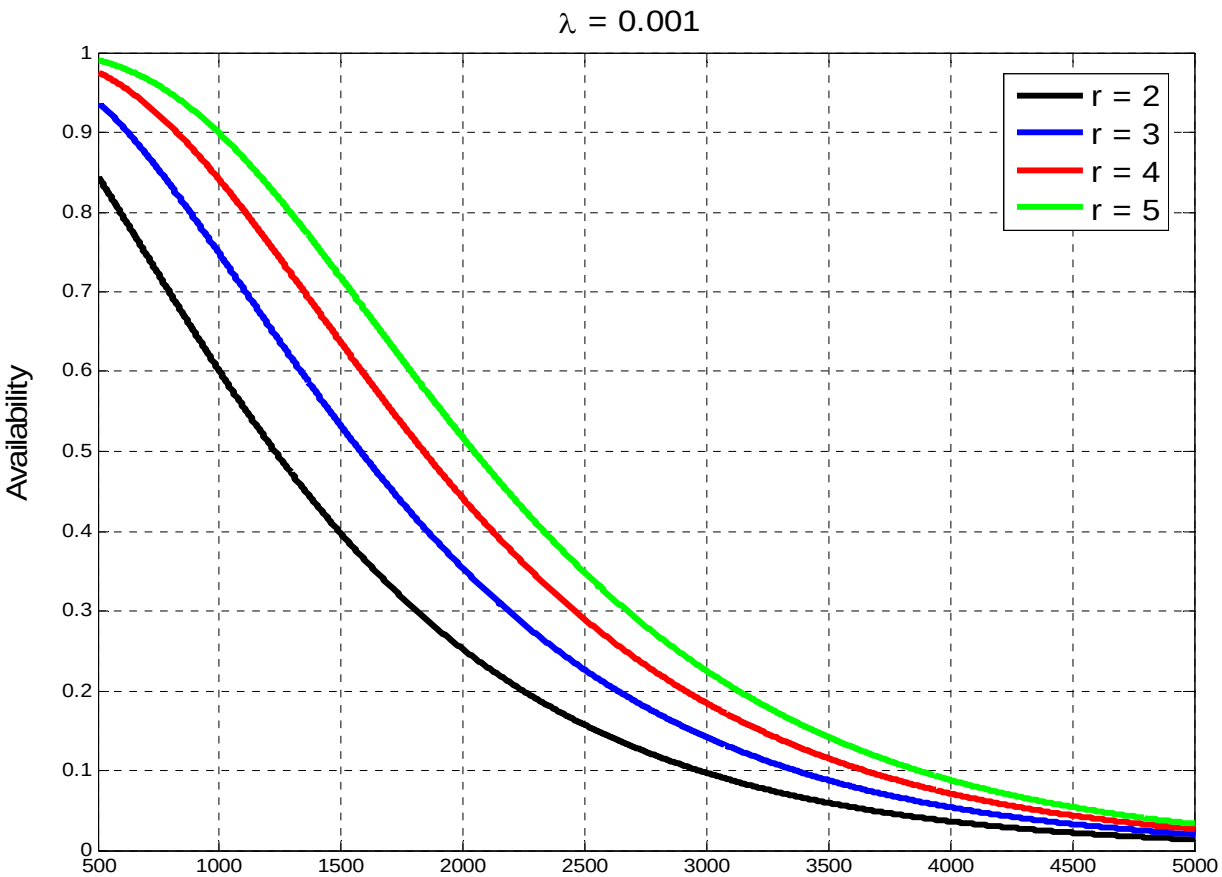
- In our model, the failures experienced by workloads follows a Poisson distribution.

$$p(w) = 1 - e^{-\lambda w}$$

Availability and workloads

$$A(w) = 1 - (1 - e^{-\lambda w})^r$$




λ = 0.001
Availability
Workload: # of update operations per period
r = 2
r = 3
r = 4
r = 5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
500
1000
1500
2000
2500
3000
3500
4000
4500
5000

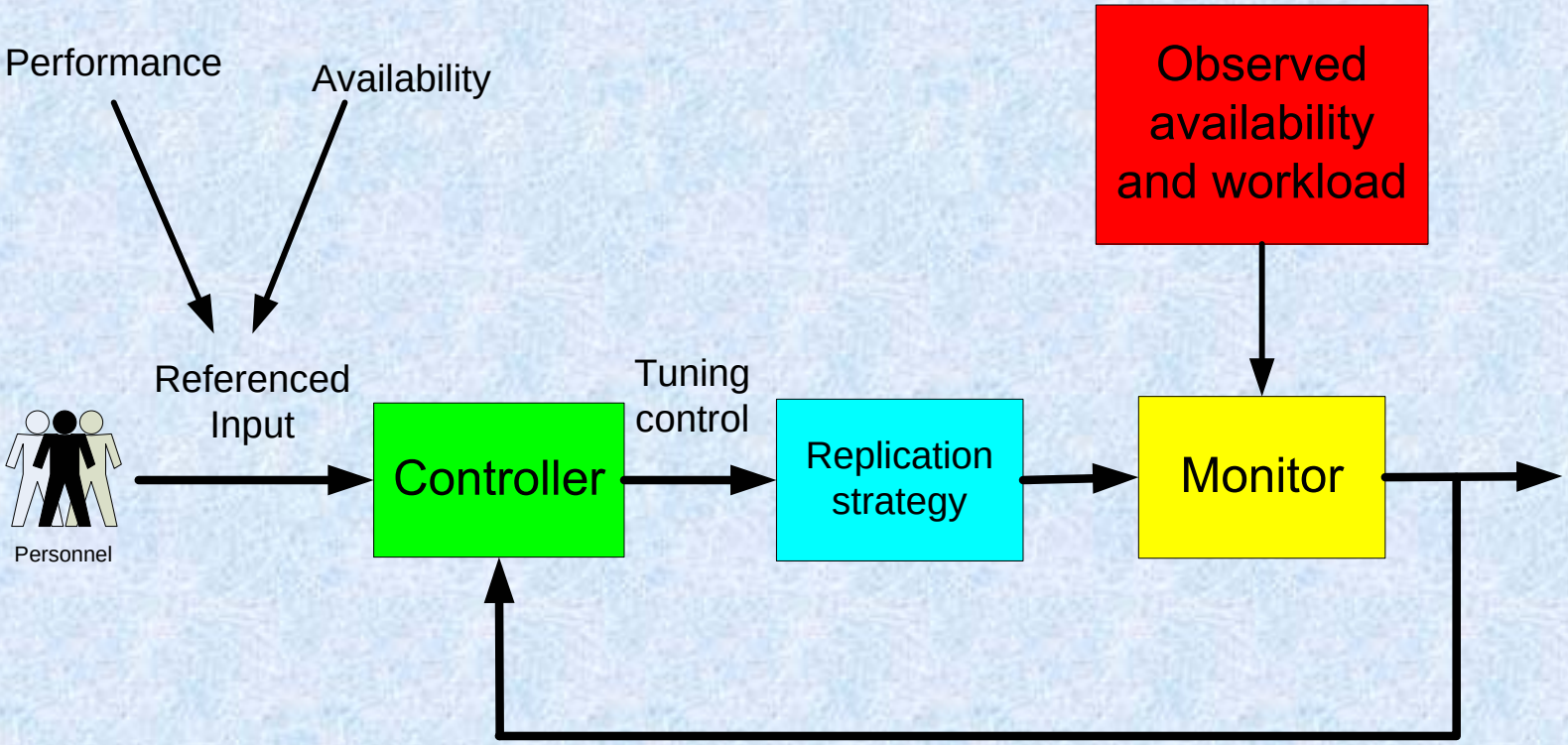
Summary

- **System availability can have an expected improvement with more replicas when the failure rate is low.**
- **When failures are more frequent, however, replication may not be able to help the system to achieve the desired availability without sacrificing too much performance.**

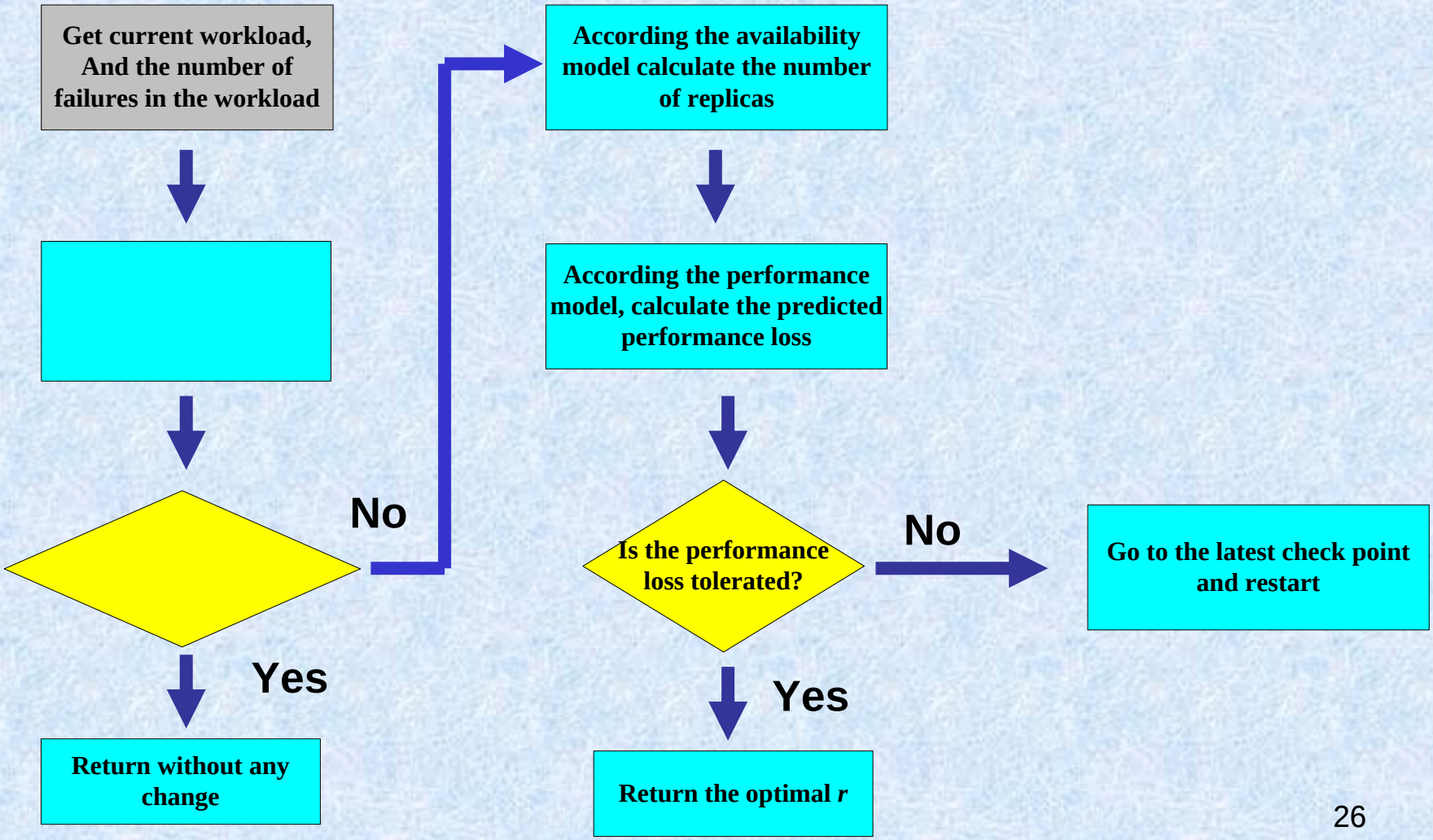
Outline

1. Introduction
   - Motivations
   - Contributions
2. Replication strategy
3. Mathematical models
   - Analytical model
   - Performance model
   - Availability model
4. Controller design
5. Conclusions and future work

Controller Design



The algorithm of controller



Outline

1. Introduction
   - Motivations
   - Contributions
2. Replication strategy
3. Mathematical models
   - Performance model
   - Availability model
4. Controller design
5. Conclusions and future work

Conclusions

- Propose two mathematical models based on a replication strategy in a distributed file system to explore the correlation among availability, performance, and workloads.

- Propose an online controller that will help system achieve an runtime optimal performance and availability via dynamically tuning the system replication policy.

Future work

- Validate the proposed models via simulation and failure traces;
- Implement such a controller in a parallel file system.

References

1. Bianca Schroeder and Garth A. Gibson. **A large scale study of failures in high-performance computing systems.** *DSN 06, 2006*
2. Chee-Wei Ang and Chen-Khong Tham. **Analysis and Optimization of Service Availability in a HA Cluster with Load-Dependent Machine Availability.** *IEEE Trans. Parallel and Distributed Systems, 18(9):1307–1319, Sept. 2007.*
3. Jeff Tian, Sunita Rudraraju, and Zhao Li. **Evaluating Web Software Reliability Based on Workload and Failure Data Extracted from Server Logs**. *IEEE Trans. Software Eng., 30(11):754–769, 2004.*
4. Michael Grottke. **Software Reliability Model Study**. *2001*, http://www.statistik.wiso.unierlangen.de/lehrstuhl/migrottke/SRModelStudy.pdf.

Acknowledgement

- National Science Foundation under grant #CNS-0720617
- Oak Ridge National Laboratory
- Center for Manufacturing Research of TTU

Thank You !