Tiresias: Black-Box Failure Prediction in Distributed Systems

Williams, A.W.; Pertet, S.M.; Priya Narasimhan
Carnegie Mellon University

Appeared in: IEEE International Parallel and Distributed Processing Symposium, 2007 (IPDPS '07)

Presented by:
Ignacio Laguna
Dependable Computing Systems Lab (DCSL)

PURDUE UNIVERSITY

Current Approaches on Fault-Tolerant Systems

- Typical chain of events in fault manifestations:

Performance Problems (e.g., network congestion, memory leaks) → Errors → Application-level Failures → Failure Propagation (may impact other components or systems)

- Current approaches on fault-tolerant systems:
  - Recovery is triggered after failures (i.e., reactive)
  - Impact of faults is not necessarily reduced
  - Symptoms of possible future problems are not considered

PURDUE UNIVERSITY

Motivation for Failure Prediction

- Take advantage of pre-failure indicators or symptoms
  - For example, performance problems or anomalies
- Being *proactive* rather than *reactive*
  - Initiate recovery faster (i.e., proactively)
- Reported facts in the literature:
  - *A fault manifests as increasingly unstable performance-related behavior before scaling into a failure*
  - Systems shows steady-state performance behavior in the non-faulty case

PURDUE UNIVERSITY

Key Contributions of the Paper

- Evaluate performance metrics to facilitate black-box failure prediction
  - Black-box—no knowledge of the application internals
  - Compare performance metrics under non-faulty and faulty conditions
  - Provide look-ahead time of application-level failures
- Empirical validation
  - Experiments performed in middleware evaluation test-bed
  - Various faults injected and explanation of adjustable parameters is provided

PURDUE UNIVERSITY

Outline

- System Model and Assumptions
- TIRESIAS' Failure-Prediction Framework
  - Data Collection
  - Anomaly Detection
  - Failure-Prediction Scheme
- Empirical Evaluation
- Future Work
- Summary

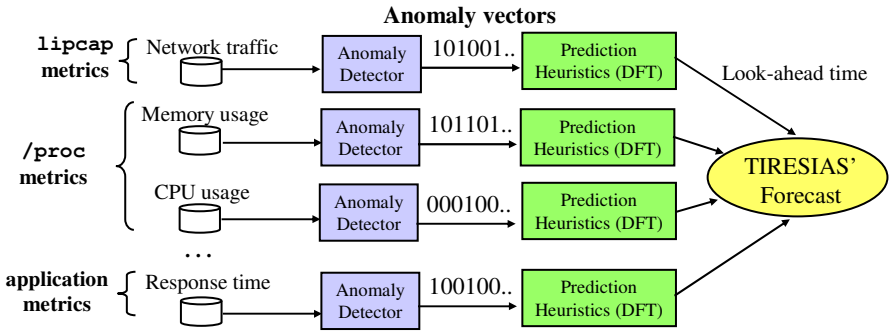
PURDUE UNIVERSITY

System Model and Assumptions

- Faults that lead to failures will affect performance metrics in an observable way
- Performance patterns can be exhibited leading up to failures
  - Even when they appear to be random
- TIRESIAS cannot predict a failure that is not preceded by a trend of anomalous behavior on the system's performance metrics
  - For example, value faults where the application produces a wrong result

PURDUE UNIVERSITY

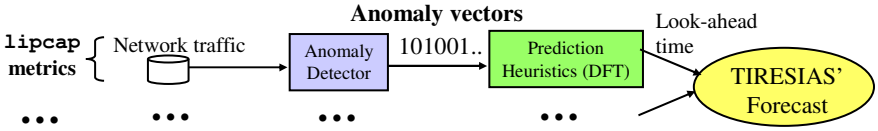
TIRESIAS' Failure-Prediction Framework



- Two-stage approach: anomaly detection followed by failure prediction
  - Anomaly detection phase—process time-series data of performance metrics with a straightforward threshold-based anomaly detector
  - Dispersion Frame Technique (DFT) is used as prediction heuristic—it looks for clustering patterns of anomaly points

PURDUE UNIVERSITY

TIRESIAS' Failure-Prediction Framework (Cont'd)



- Anomaly vectors—a binary sequence of 1's and 0's
  - Indicates whether each data-point of the metric's time-series is seen as anomalous or normal
- DFT seeks patterns of behavior that might indicate instability
  - If a pattern if found, a warning of a imminent failure is fired
- A prediction does not reveal the root cause of a problem (there is no diagnosis) and does not guarantee that a failure will occur

PURDUE UNIVERSITY

Data Collection for Anomaly Detection

- Data is collected under non-faulty runs as well as under injection of faults
- Resource usage is retrieved from `/proc` every 5 seconds in Linux machines
  - CPU usage (%)
  - Available memory (bytes)
  - Context-switch rate (per second)
- Network traffic is monitored by the use of the `libcap` library
  - Timestamps are recorded every time a packet appears on the wire
  - Capturing mechanism does not add overhead to the network traffic

PURDUE UNIVERSITY

Anomaly Detection Scheme

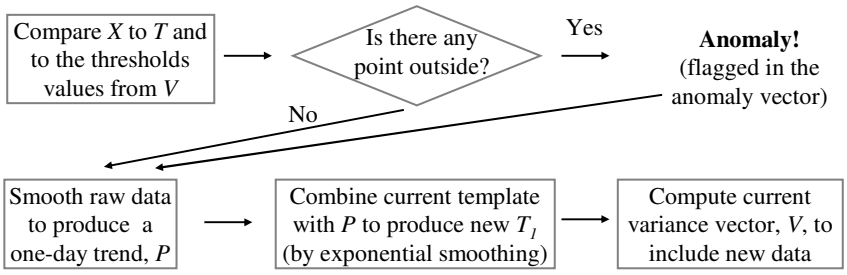
- Any anomaly-detection scheme can be used in TIRESIAS
  - Paper contribution is not the anomaly-detection scheme, but its use for failure prediction
- Algorithm developed for network-related failures
- Algorithm based on computing the mean of the data and looking at a three-standard-deviations ($\pm 3\sigma$) rule
- Model is based in:
  - *Template* — a model of time-varying expected/normal system behavior
  - *Envelopes* — represent tolerance limits for metric's normal behavior

PURDUE UNIVERSITY

Anomaly Detection Scheme

- Four vectors of data used for network-traffic analysis:
  - *X*, current raw network-traffic captured data
  - *P*, smoothed traffic data (the trend of the data avoiding extreme discontinuities)
  - *T*, template that describes normal network-traffic behavior
  - *V*, variance template to calculate standard deviations (thresholds)

- Steps to detect anomalous network-traffic points:



PURDUE UNIVERSITY

Prediction Heuristics: The Dispersion Frame Technique (DFT)

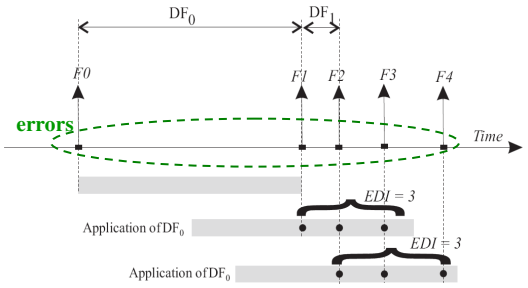
- Dispersion Frame Technique (DFT) originally developed to predict failures in hardware devices (e.g., hard disks)
  - A set of heuristic rules developed from empirical studies (e.g., error-logs)
  - Check the relationship between anomaly occurrences by examining how closely they occur in time
  - First time it is applied to time-series error-logs that are output by an anomaly detector
  - Applied for prediction of hard-drive failures with a success rate of 93.7%

- DFT uses two concepts:
  - *Dispersion Frames* (DF) — inter-arrival time between successive error events (i.e., successive anomalies of the same metric)
  - *Error Dispersion Indices* (EDI) — the number of errors (or anomalies) in one half of a DF

PURDUE UNIVERSITY

Dispersion Frame Technique (DFT)

A highly related group of anomalies exhibits a high EDI — a sign of instability



***Dispersion Frames* (DF)** — inter-arrival time between successive error events

***Error Dispersion Indices* (EDI)** — the number of errors (or anomalies) in one half of a DF

- Heuristic clustering rules for prediction:
  - (1) When two consecutive indices from the successive application of the same DF exhibit EDI ≥ 2
  - (2) When two consecutive indices from two successive DFs exhibit EDI ≥ 2
  - (3) Four monotonically decreasing DFs, and at least one DF = ½ the previous one
  - (4) DF < 10 minutes

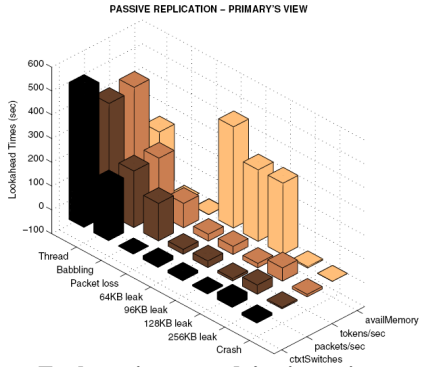
PURDUE UNIVERSITY

Empirical Evaluation

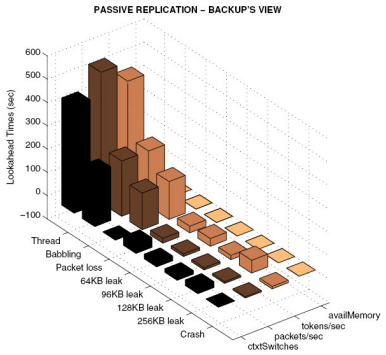
- Experiments run in a distributed environment
  - 4 nodes (850MHz processor, 512MB RAM, Linux 2.4.18)
- Simple two-tier distributed client-server application
  - One client and a dual-redundant server
  - The client sends a request to the server, which returns 32 Kb of data to the client.
- Each experiment covers 45, 000 round-trip client invocations, and runs for 15 minutes
- A total of 12 different faults injected
  - At the primary replica: memory leaks of 64Kb, 96Kb, 128Kb, 256Kb
  - At the backup: thread-leak, babbling node, packet loss and abrupt crash

PURDUE UNIVERSITY

Look-ahead Times for Passive Replication



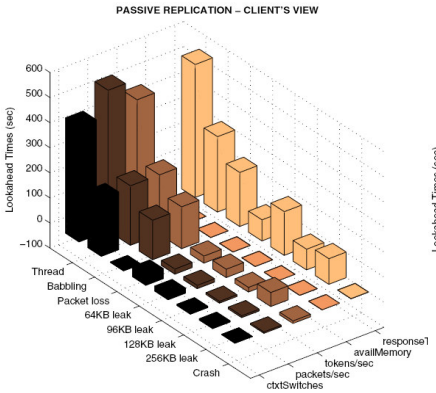
Faulty primary node's viewpoint



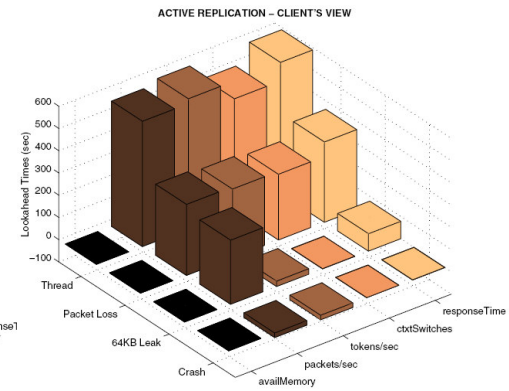
Non-faulty backup node's viewpoint

- Failure prediction can be performed on a node other than the faulty one
- CPU usage was not stable enough for consistent failure-prediction
- Failure rate does affect look-ahead time
  - The faster the failure rate, the smaller the look-ahead time

Look-ahead Times for Non-faulty Client Node's Viewpoint



Passive replication



Active replication

- A false positive rate of 2.5% was found (5 false positives out of 200 runs)

Future work

- Extend TIRESIAS for root-cause analysis (diagnosis) and proactive recovery
- Investigate false positive rate more accurately by gathering more empirical data
- Employ alternative anomaly-detection algorithm along with DFT
- Develop a distributed DFT to correlated failures across the nodes

PURDUE UNIVERSITY

Summary

- TIRESIAS: a framework for black-box failure prediction by the analysis of performance metrics
- It exploits clustering rules that were originally develop for single-node hardware failures, to predict failures in distributed environments
- Experiments with different fault injections showed that failures can be predicted with look-ahead time to allow proactive recovery

PURDUE UNIVERSITY