Joint Laboratory for Petascale Computation

Signal Analysis for Modeling the Normal and Faulty Behavior of Large-scale HPC Systems

Ana Gainaru, Franck Cappello, Bill Kramer

Motivation

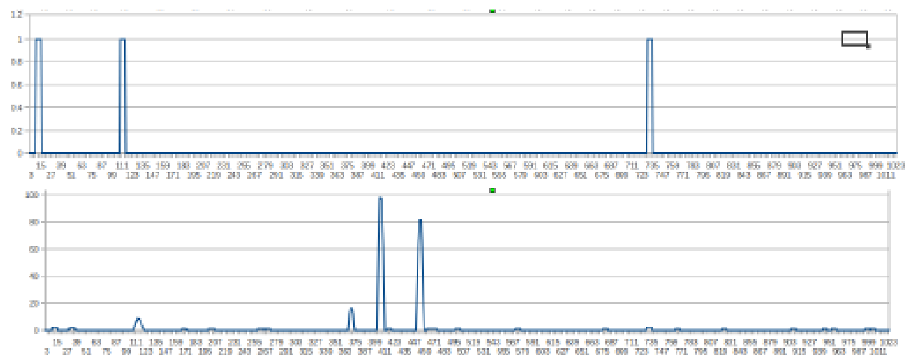
- Log files give useful information about hardware, application, user actions
  - Systems generate events
- Data mining on raw logs
  - Detection of abnormal behaviors
  - Root cause analysis
  - Event prediction
- Observation
  - Different error types present different distributions
  - Analyze behavior differences in all events

Signal analysis

- Signal analysis
  - Occurrences of each event types are considered as time series
    - Different event types become different signals
  - Variation in signal's normal behavior identifies suspicious events that could represent failures
- Advantages
  - Easy to shape and characterize different behaviors
    - Related work do not make a difference between different events and analyze all of them in the same way
  - Does not require filtering
  - Does not have cross correlation limitation

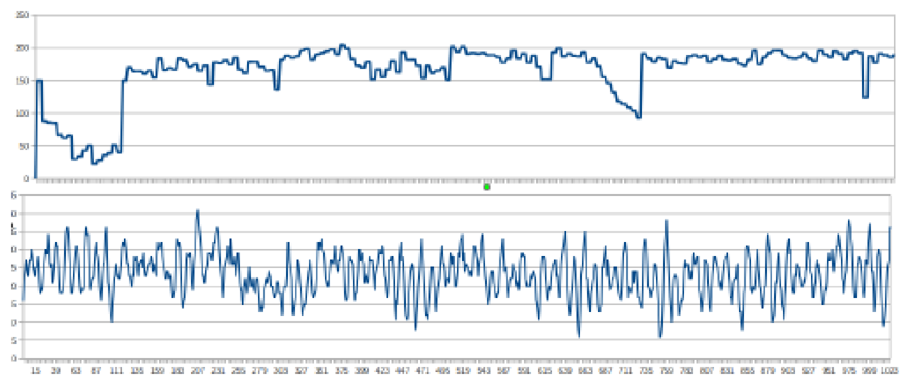
Signal analysis

Silent signal
characteristic of error events. PBS errors



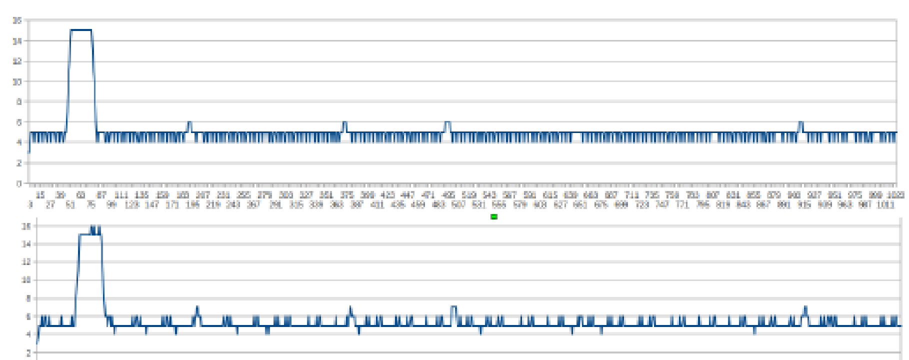
(a)

Noise signal
typically Warning messages: Memory errors corrected by ECC



(b)

Periodic signals
daemons, monitoring



(c)

Table of contents

- Extracting signals
  - Pre-process
  - Identifying periodic, silent and noise events
  - Anomaly detection
  - Correlations
- Results
  - Anomaly detection
  - Filtering
  - Prediction

Pre-processing

- Looking for frequently occurring messages with similar syntactic patterns
  - HELO: Hierarchical Event Log Organizer
- Signal extraction
  - Use a sampling rate (different depending on the signal)
  - Map number of events for each sample

Template	Event type
failed to configure resourcemgmt subsystem err = 10	Processor cache error
psu failure	PSU failure
component state change: component * is in the * state *	State change in a component

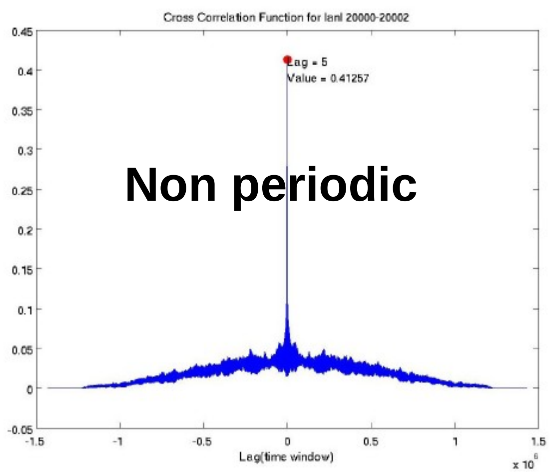
Table II
EXAMPLES OF TEMPLATES AND THEIR EVENT TYPES

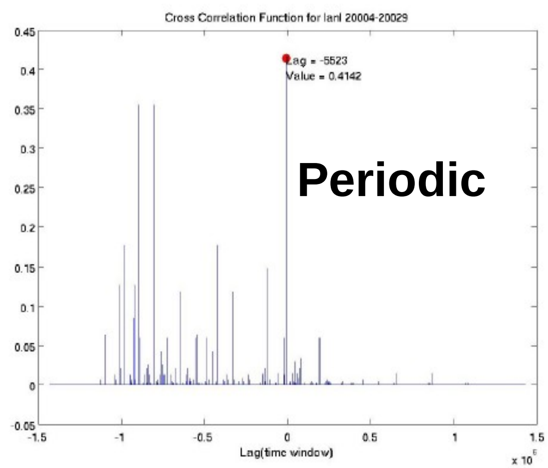
Signals analysis workflow

- First step: identify periodic signal
  - Consider all signals are periodic
  - **Establish the best sampling rate**
    - Nyquist Theorem: A signal with frequencies of max B hertz, is determined by giving its ordinates at a series of points spaced 1/(2B) units apart.
    - start with a low sampling rate (mean time between 2 occurrences of the event type)
    - increase the rate until it exceeds the minimum time lag OR a periodic signal if found

Signals analysis workflow

- **Determine if the extracted signal is periodic**
  - Auto-correlation function
    - Compute the similarity between a signal and itself for different time lags
  - If similarity over a threshold, signal is periodic
    - Threshold fixed automatically by Pearson’s significance test





Examples of auto-correlation graphs

Signals analysis workflow

- For periodic signals (after auto-correlation):
  - **Compute the frequencies**
    - Compute the power spectrum of the signal
    - Filter the power spectrum to remove noise

- Second step: **identify silent and noise signals**
  - Fix sampling rate, all signals same number of samples
  - Transform the signal to the wavelet domain
  - Use wavelet shrinkage and thresholding to eliminate the noise from our signal.
    - Extract the main features of the signal

Signal statistics

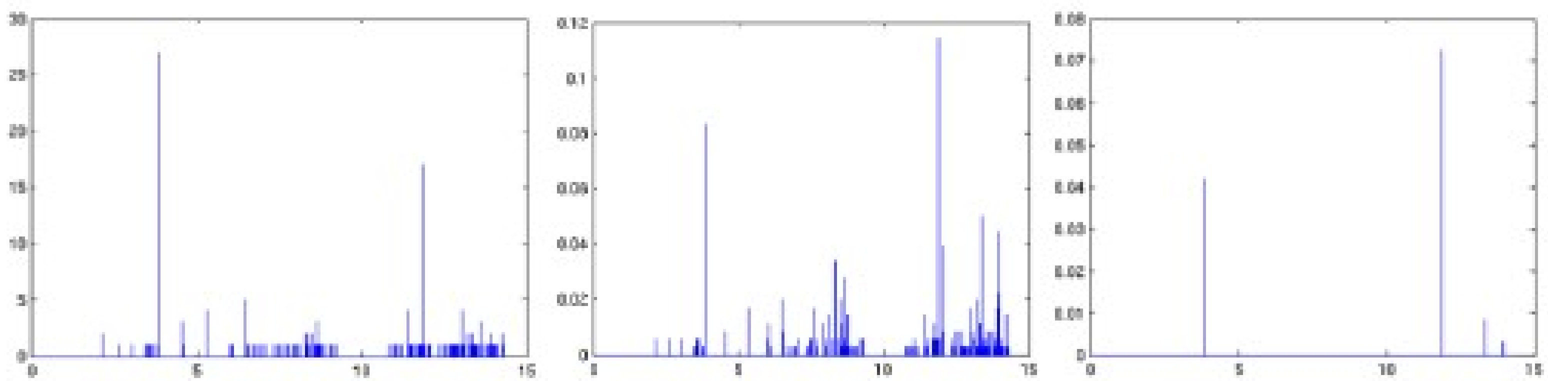
System	Mercury	LANL
Periodic signals		
Number	11	2
Percentage	2.7%	3.8%
Correct frequencies	90.9%	100%
Silent signals		
Number	338	39
Percentage	82.6%	73.6%
Noise signals		
Number	60	12
Percentage	14.7%	22.6%

Table III
STATISTICS FOR DIFFERENT SIGNAL TYPES

Anomaly Detection

- **Detect changes in the structure of the normal behavior of the signal signal**

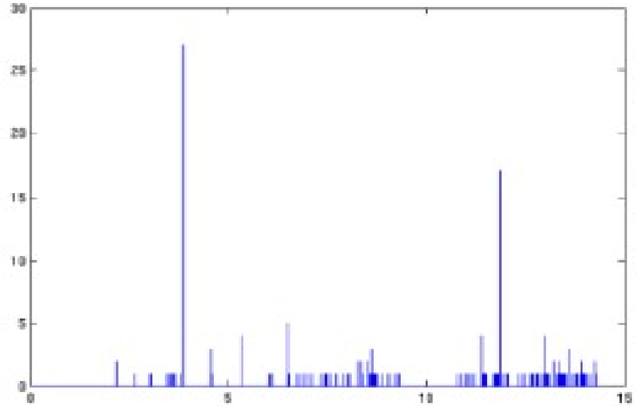
- Changes in frequency:
  - Use the moving average technique
    - Replace each value with an average of neighboring values
    - Highlight high rates
  - Filter the resulting signal to remove the normal behavior

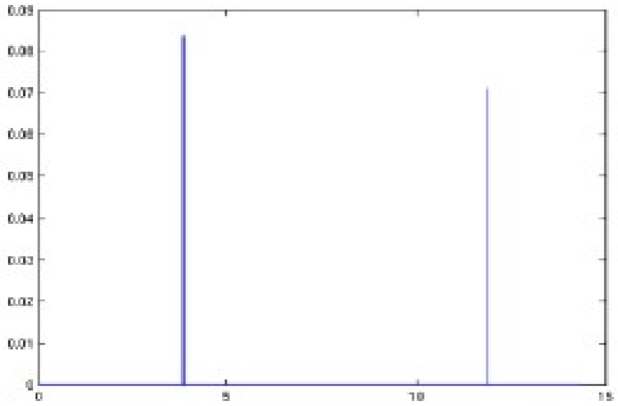


Anomaly Detection

- **Detect changes in the structure of the normal behavior of the signal signal**

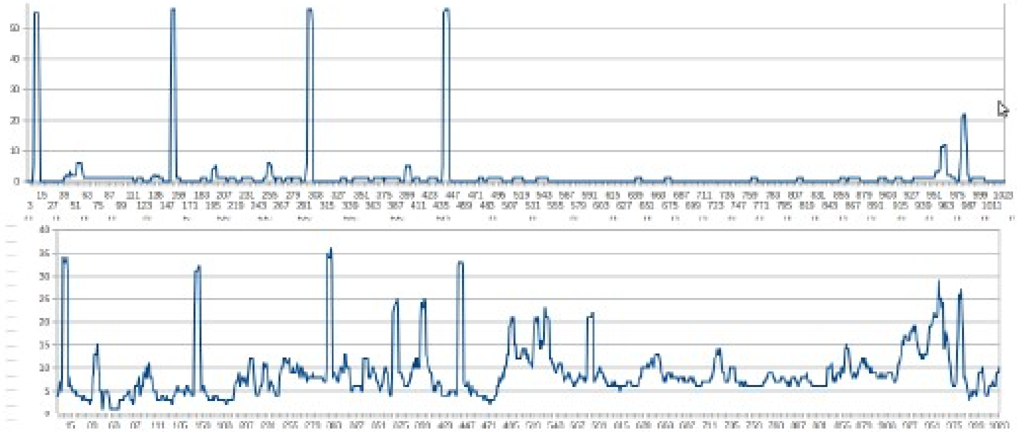
- Changes in intensity:
  - 2 filters:
    - One putting emphasis on large values
      - Enhance the difference between peaks and the normal behavior
    - One on abnormal small values
      - Enhancing the deviation of time units with decreasing intensities from normal.





Event correlation

- **Detect correlations between signals**
  - Cross-correlation function



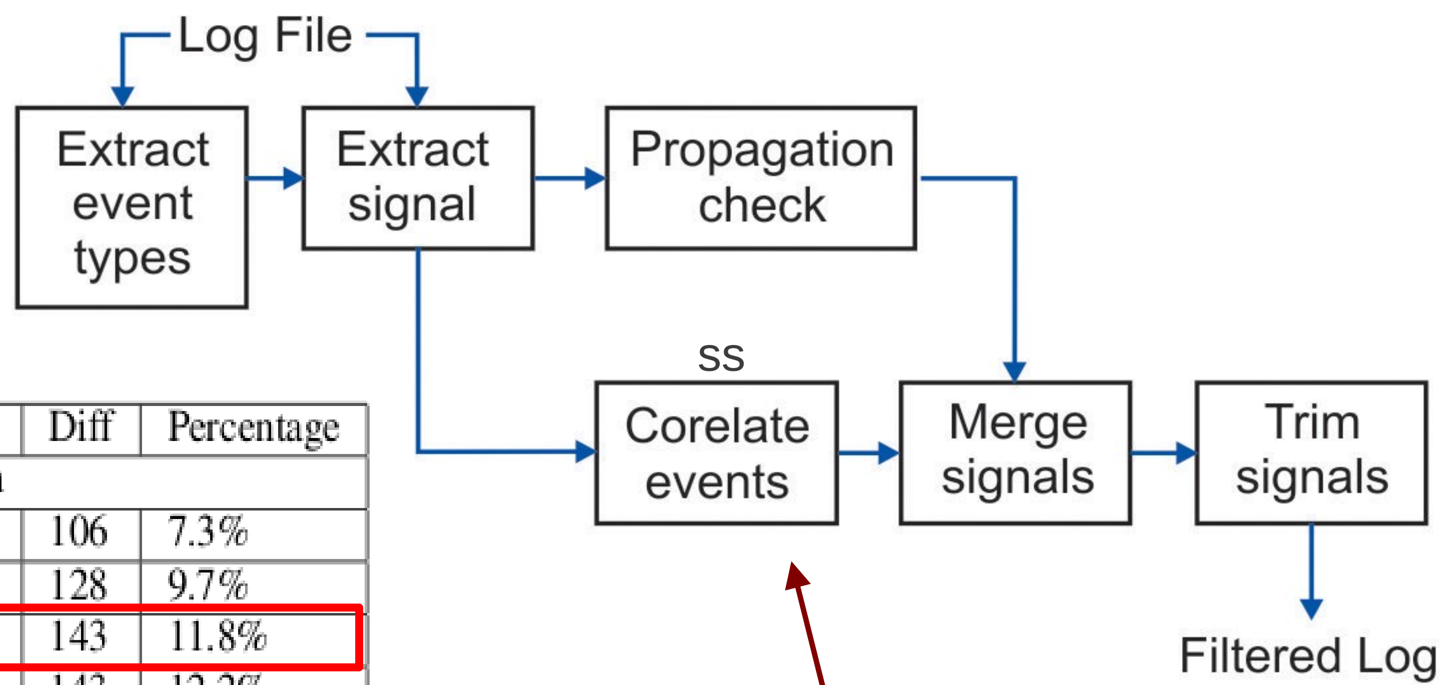
  - Normal behavior of the signals may be different
    - similarities are masked
- **Focus only on anomaly correlations**

System	Mercury	LANL
Average lag (seconds)	45	23
Correlated events (percentage)	68%	71%

Experiments

- Filtering analysis
  - Necessary step in almost all data mining methods
  - Use signal analysis to improve the process
- Anomaly detection
  - LANL trace (both system and failure log)
- Event prediction
  - We are interested in precision and recall
  - And also in the time interval between prediction and event

Filtering events



Time window	T+	T	Diff	Percentage
LANL system				
160	1,450	1344	106	7.3%
200	1,323	1194	128	9.7%
240	1,212	1068	143	11.8%
280	1,171	1028	143	12.2%
Mercury				
160	12,872	12,189	682	5.3%
200	12,732	12,006	726	5.7%
240	12,644	11,720	923	7.3%
280	12,591	11,596	994	7.9%

Table V
FILTERING RESULTS

Filter events on different locations only if:

- They they have a propagation behavior
- They are correlated

Anomaly detection

- Detect anomalies in the signal
  - Necessary step since the correlation is based on it
  - The step is offline
    - We have all the data to extract the normal behavior
  - We only use the LANL trace
    - Failures are identified by sys admins in the failure log

Precision	87%
Recall	51%
Average lag (before)	107 time units; aprox 18min

	Facilities	Hardware	Human Error
Precision	86.4%	88.1%	80.8%
Recall	38%	61.2%	9.6%
Average lag	329s	456s	65s

	Network Error	Software	Unknown
Precision	83.2%	92.1%	90.6%
Recall	52.8%	55.1%	26%
Average lag	608s	389s	198s

Anomaly detection results

Failure prediction

For event prediction we use 1/10 of the LANL log for training and prediction is done on the rest

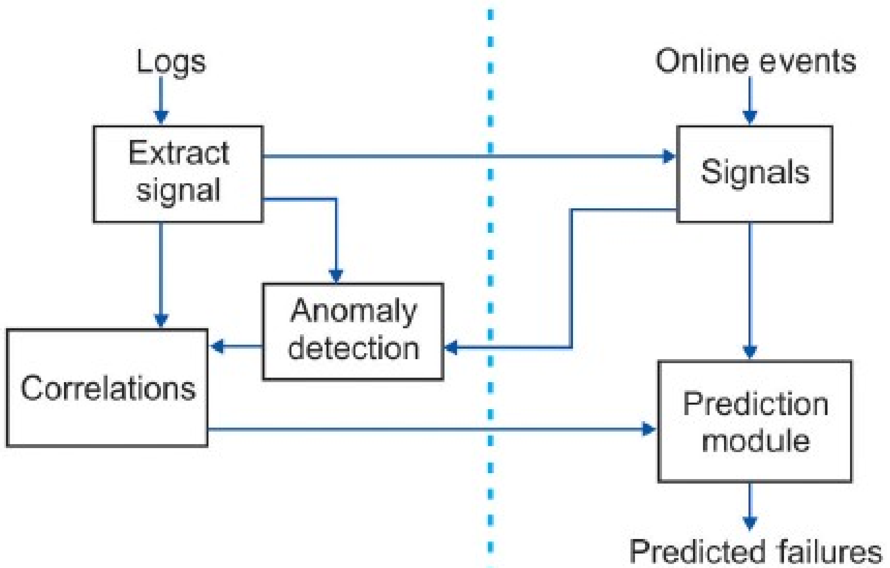
Distribution of failure types in LANL logs

Facilities	Hardware	Human Error
1.55%	61.57%	0.64%
Network Error	Software	Unknown
1.8%	23.02%	11.38%

Prediction methodology

Offline

Online



int La

Prediction results

Precision	93%
Recall	43%
Average lag (before)	32 seconds

	Facilities	Hardware	Human Error
Precision	89.2%	93.8%	80.8%
Recall	38%	45.1%	9.2%
Average lag	68s	45s	15s

	Network Error	Software	Unknown
Precision	91.2%	93.7%	91.6%
Recall	42.8%	41.1%	23.4%
Average lag	48s	5s	19s

Conclusions

- Apply signal analysis concepts for log analysis
  - Different event types may have different normal behaviors
  - Faults affect event types in a different way
  - Fault correlation is affected by this model
    - Gives a higher number of correlations
- Apply the model for fault prediction
  - Precision of 91% and recall of 43%
  - Prediction 30 seconds before event

Future work

- We do not consider the analysis time
  - Using more efficient methods of storing the correlations
  - Optimizations in the signal analysis modules
- Extract tuples of n correlating events
  - Collaboration with INRIA Grenoble
  - Analyze correlation's lifespan
- Combine the prediction module with a checkpointing strategy

Thank you