Root Cause Analysis (RCA) for Future Networks :

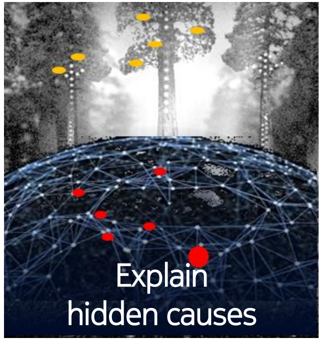
Technology, Vision and Challenges

Armen Aghasaryan, Bell Labs, Paris-Saclay

Shonan Seminar on Causal Reasoning in Systems
24-27 June, 2019

Management of Future Networks with Root Cause Analysis
Introduction

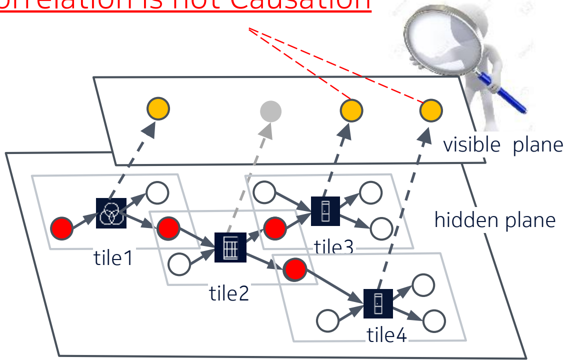
The digital era calls for an unprecedented need of real-time monitoring and Root Cause Analysis (RCA) of complex systems deployed at a global network scale



RCA duality character: RCA is applied to methodically identify and correct the *root causes* of events, as opposed to simply addressing their *symptomatic result*.

RCA impact perspective: “Root cause” may be described as the point in a causal chain where applying a corrective action or intervention would prevent the problem from occurring.

Correlation is not Causation

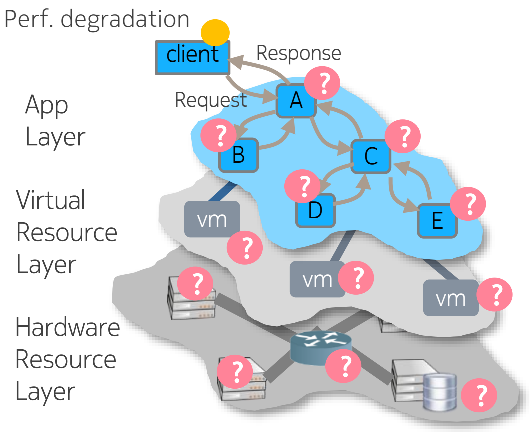


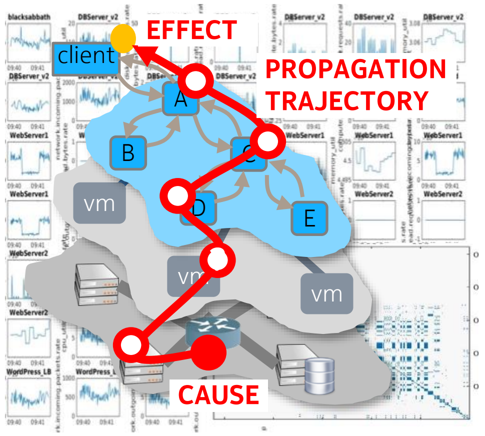
Model-based approach

- Model explains the occurrence, manifestation, and propagation fault situations
- Model learning techniques:
  interventional / observational / knowledge-based

NOKIA

RCA for Cloudified Networks





High volume of measurements and alarms to be handled at different layers

- Relevant alarm definitions and measurements not trivial to identify

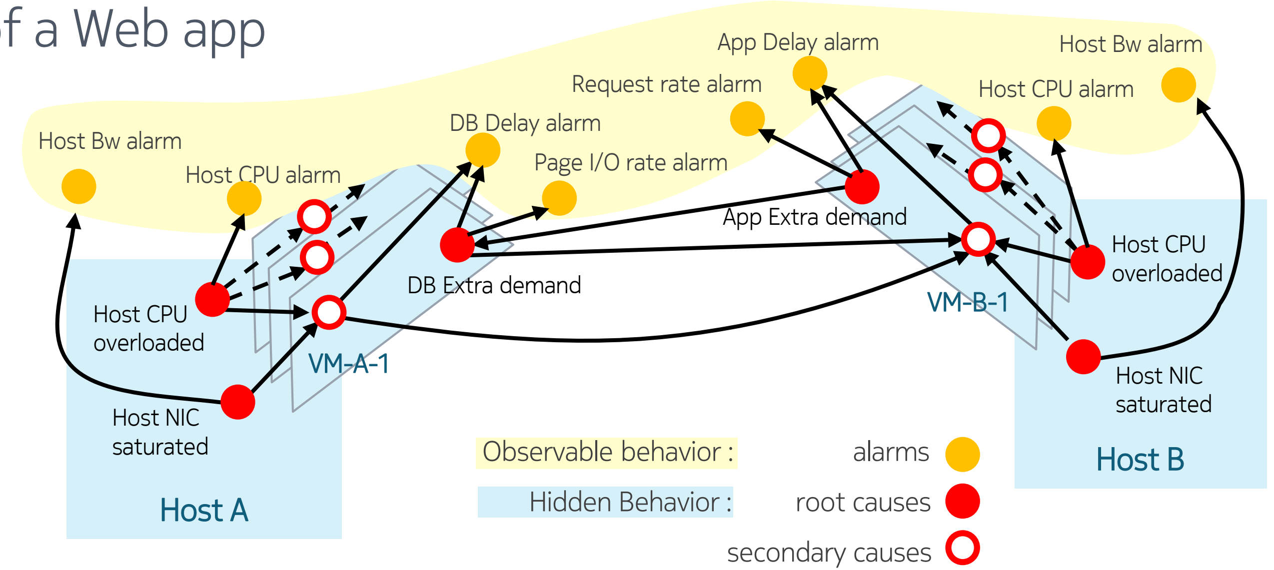
Ambiguity in the interpretation of observations by the human operator

- Single fault may produce multiple alarms, a given alarm can be caused by different fault conditions
- Number of possible failures grows exponentially in the future (terminals – IoT, network components – 5G)

Delayed understanding of faults leads to negative impact on businesses

RCA for Cloudified Networks

Example of a Web app



Symptoms: (web) application performance degradation - generation of infrastructure and application level alarms

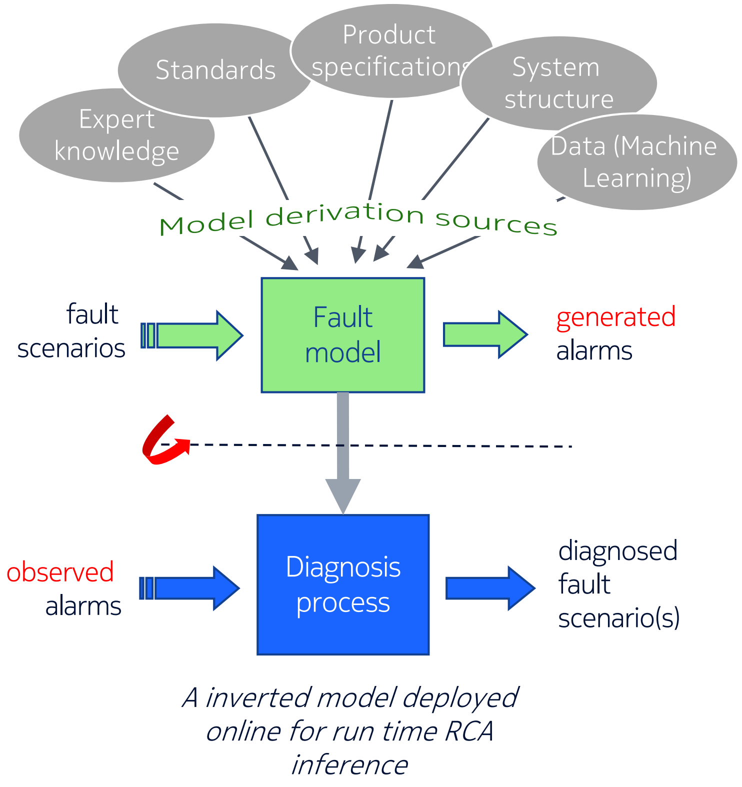
Multiple possible underlying causes

- User request peak or permanent increase (App extra demand)
- Underlying resource overload by other tenants (host network interface, host CPU saturated)
- Remote fault propagation via app dependencies (e.g. BD delay)
- Remote host failures, etc.

Fault model-based approach

Setting up the RCA design as a model-based fault diagnosis using a “model inversion” approach

- Fault model replicates the fault behavior of the network
  - representation of initial faults, fault propagation, and alarm generation
  - different types of models representing the interplay between observed variables and hidden states
    - Labelled automata, Petri Nets, Pattern matching, Hidden Markov models, Bayesian Networks, …
- Diagnosis algorithms inverse the model
  - without reconstructing the global state (to avoid state space explosion)

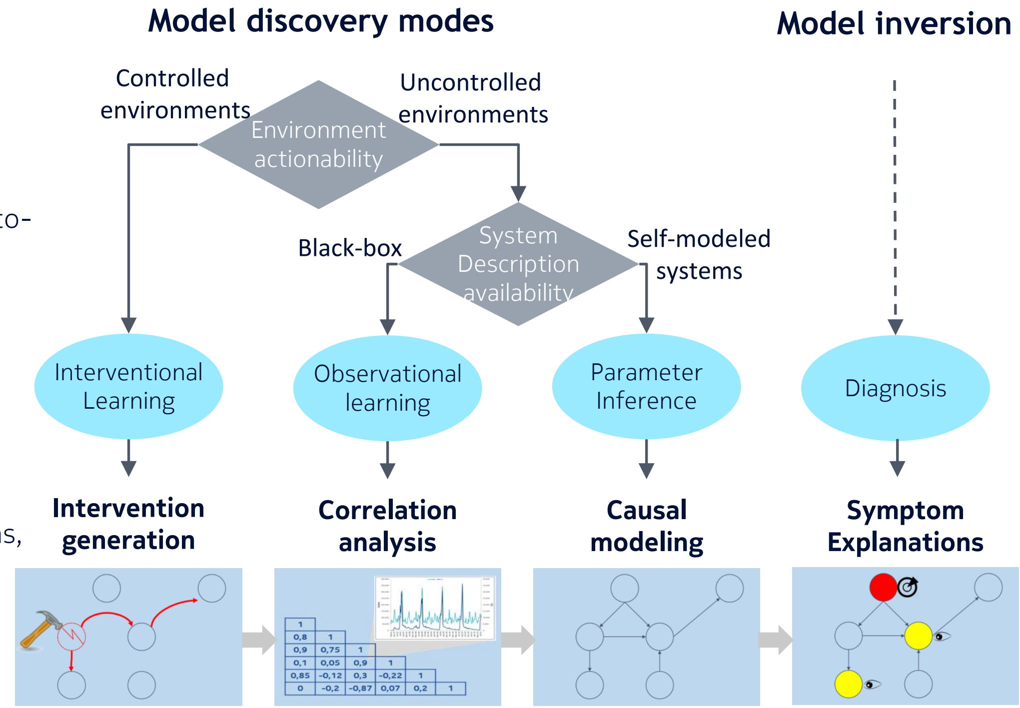


Logical view of the RCA design process

Framework : from model discovery to fault diagnosis

- **Automated (fault) model discovery modes**
  - Interventional learning: stimulus-based App Profiling Sandbox
  - Observational learning: lagged correlation analysis & pattern learning
  - Self-modelled data/systems, e.g. perf. logs allowing auto-extraction of a Bayesian Network structure

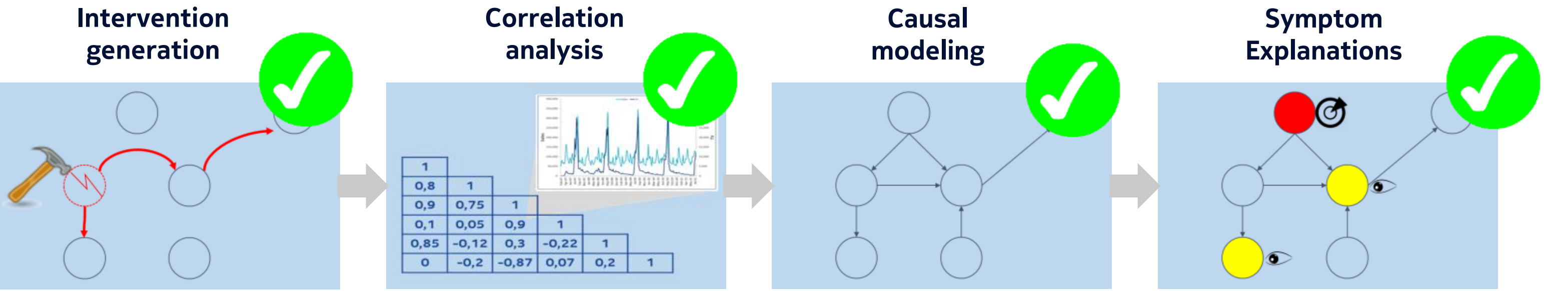
- **Model Inversion: from symptoms to causes**
  - Identification of possible causes - provides the most likely explanations given the observed alarms.
  - Using *Bayesian network inference*, tile-based trajectory composition / Petri Net unfolding, Viterbi-like algorithms, etc.



NOKIA

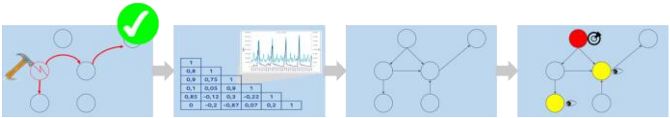
Interventional Learning

Stimulus-based App profiling



NOKIA

Stimulus-based App Profiling Sandbox



Goal: Automated fault model discovery

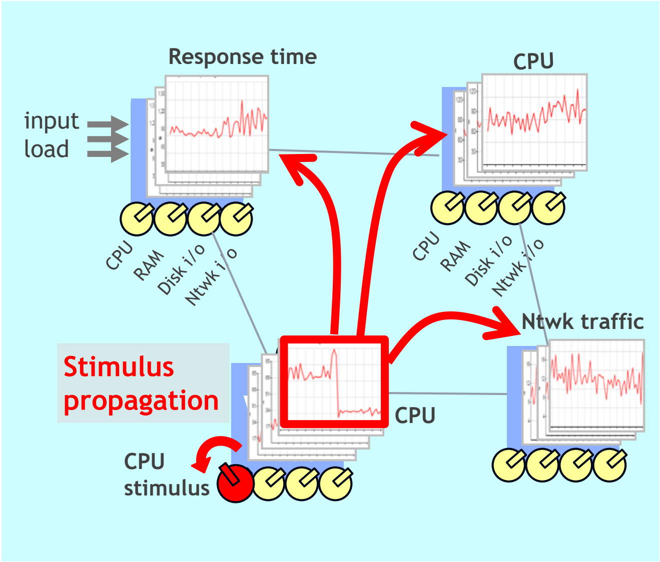
Aproach: Stimulus creation framework for virtualized distributed apps

- Interventional learning of causal Tiles via systematic resource perturbations and their impact analysis
- Algorithm for causal chaining of fault trajectories

Intervention Generation

“Stimulus” creation: system-level fine-tuned control of available computing resources on a given app node (CPU, RAM, Disk I/O, Ntwk I/O)

Discovery of causalities: analysis of the reactions of other app nodes to the current stimulus

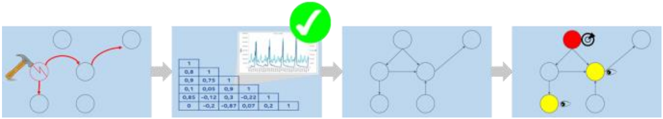


Methodical elucidation of causal dependencies

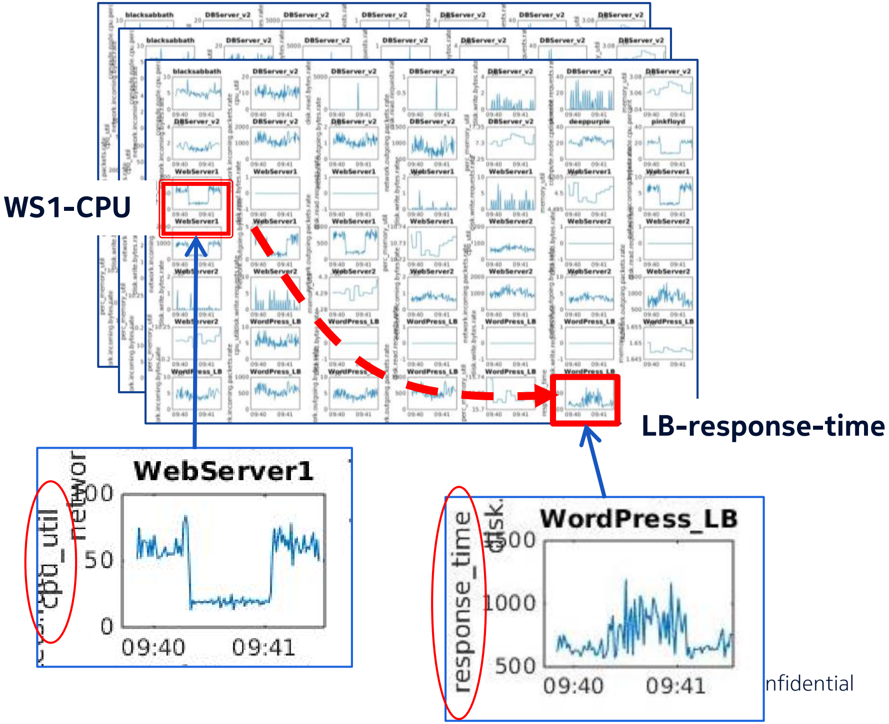
NOKIA

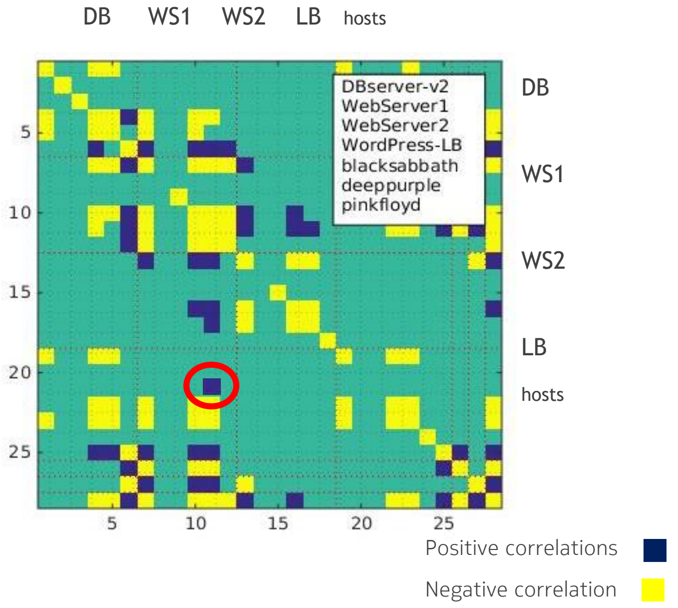
Stimulus-based modeling

Correlation analysis



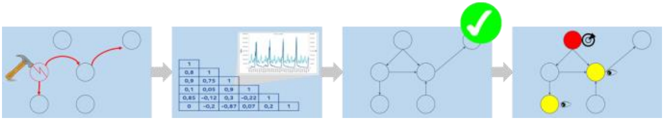
One datasheet per stimulus





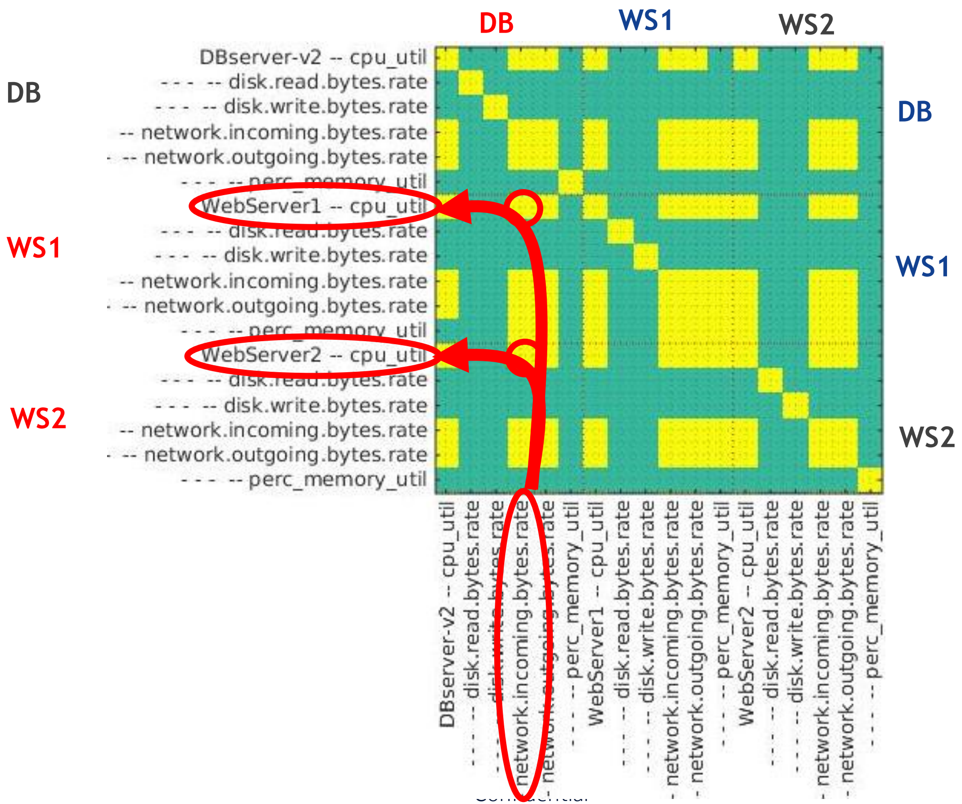
nfidential

NOKIA



Stimulus-based modeling

Causal Modeling: Interpretation of correlations given the stimulus



Current stimulus is on DB : network Bw down

NOKIA

Stimulus-based modeling
Tile extraction

Hidden faults (primary or propagated)

- we stimulate (and know) the ground truth
  - Network Bw, CPU, Memory, Disk I/O
  - Input Load

NETW↘
CPU↘
MEM↘
DISK IO↘

NETW↗
CPU↗
MEM↗
DISK IO↗
LOAD↗

Hidden states – alert enablers

- enable alarm generation
- defined by OpenStack & proprietary meters

network.outgoing.bytes.rate↘
network.incoming.bytes.rate↘
cpu_util↘
⋮

network.outgoing...↗
network.incoming...↗
cpu_util↗
⋮

Visible events - alarms

- defined on the basis of the existing meters

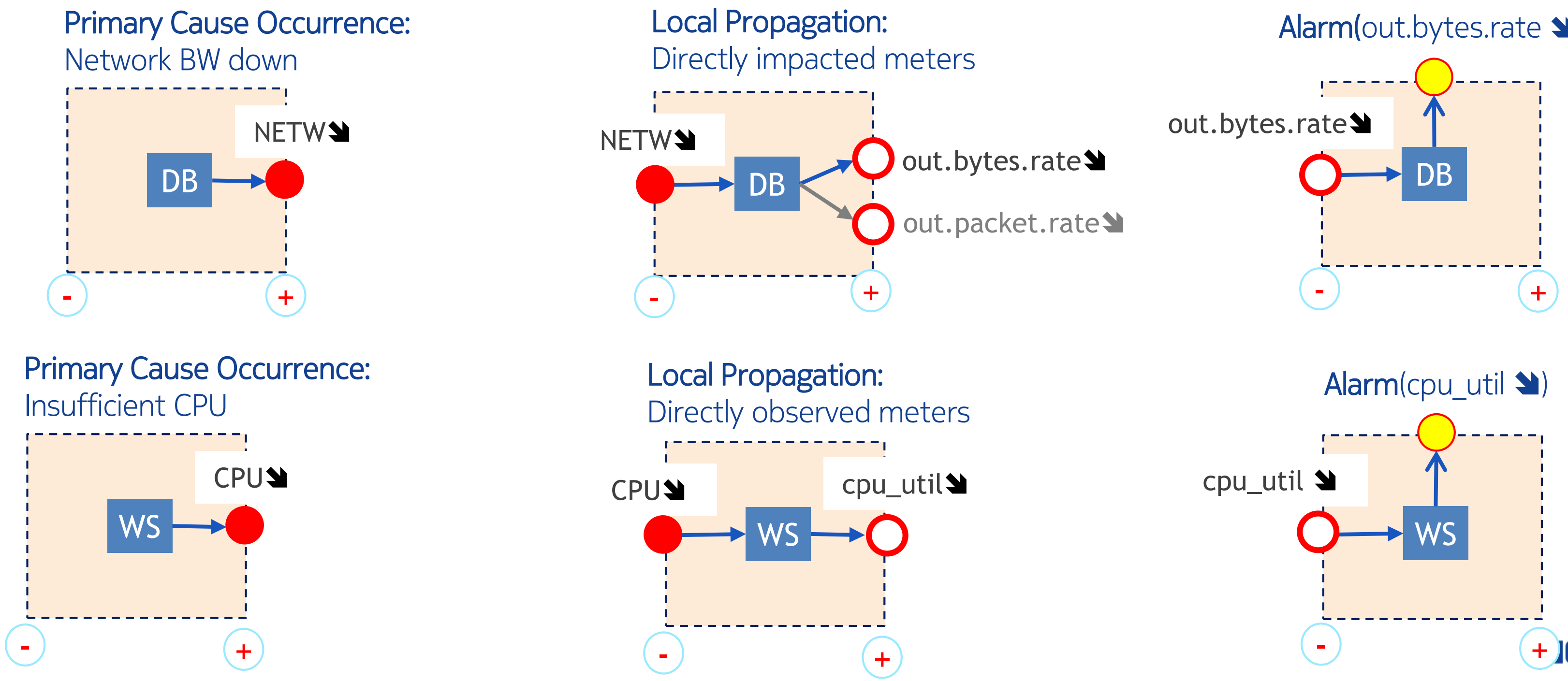
Alarm (network.outgoing.bytes.rate ↘)
Alarm (cpu_util↘)
⋮

Alarm (network.outgoing...↗)
Alarm (cpu_util ↗)
⋮

Stimulus-based modeling

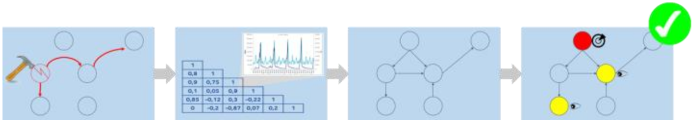
Tile extraction : example of stimulus on DB network bandwidth down

resourceType	resourceID	meterType	CorrCoeff	resourceType	resourceID	meterType
DB	db_server_v2	network.outgoing.bytes.rate	0.87	WS	WebServer1	network.incoming.bytes.rate
DB	db_server_v2	network.outgoing.bytes.rate	0.85	WS	WebServer1	cpu_util

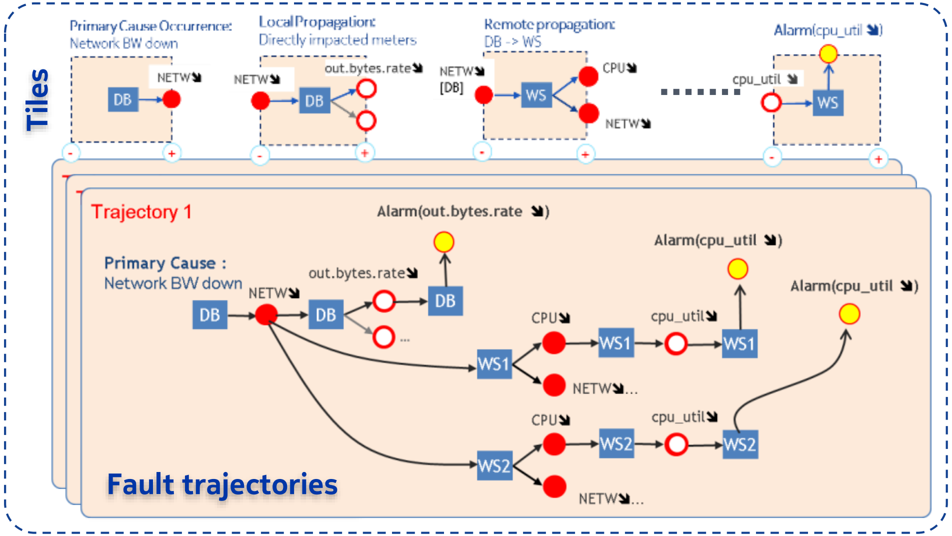


Stimulus-based modeling

Diagnosis



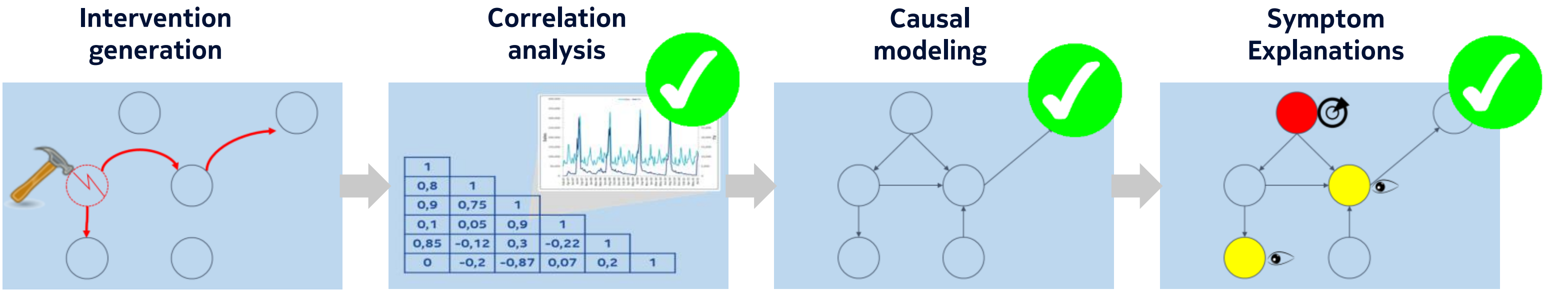
- **Interventional learning of causal Tiles:**
  stimulus-reaction transitions (tiles) represent causal relationships between resource stimuli and the reactions at local & remote nodes
- **Causality chaining via net unfolding:**
  the alarm flow guides the on-line composition of reusable tiles into hypothetical fault propagation trajectories
- **Fault trajectory inference & diagnosis**
  Iterative selection of the best explanations (trajectories) using Max likelihood estimation algorithms (Viterbi algorithm variants)



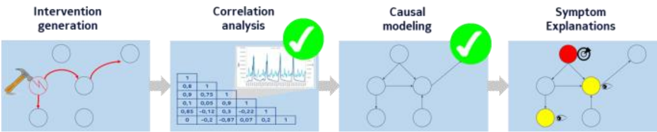
NOKIA

Observational learning

Correlation and Causality Analysis with CloudBand (CB) alerts



Causality inference from CB alert logs



Goal: automatic inference of correlation & causality rules from observed alert logs

Approach:

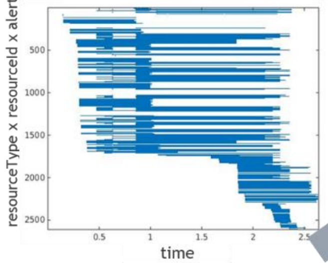
- Optimized computation of type-based correlations
- Causality inference based on time lagged correlations

Implemented by OpenStack Vitrage

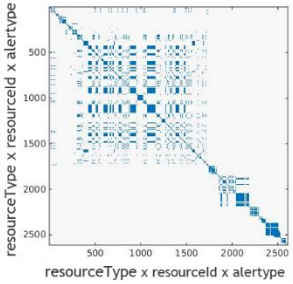
Presented at OpenStack Summit 2017, Boston

THE PLATFORM FOR NFV CLOUDBAND™

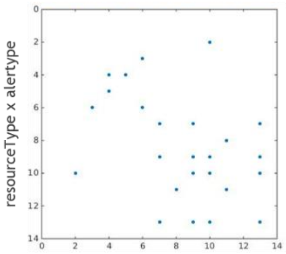
Alert time-series on distinct resources

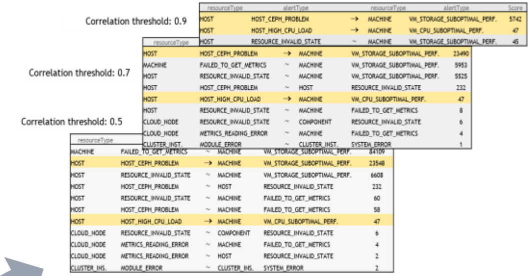


Instance-based correlations



Type-based correlations





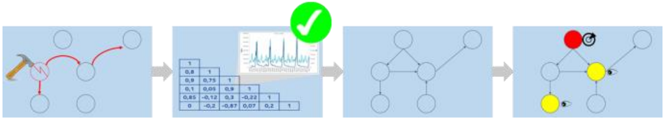
Inference of causal relationships

Data transformation pipeline

NOKIA

Causality inference from CB alert logs

Correlation Analysis



Datasets from CB:

- e.g. Cloud platform @Naperville: ~ 14K alarms, 1,5K distinct resources

CLUSTER_INSTANCE
CLOUD_NODE
OBJECT_STORAGE_NODE
COMPONENT
HOST
MACHINE

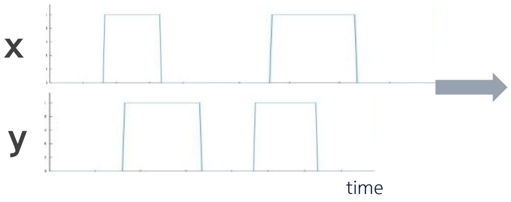
MODULE_ERROR
SYSTEM_ERROR
RESOURCE_INVALID_STATE
FAILED_TO_GET_METRICS
METRICS_READING_ERROR
HOST_CEPH_PROBLEM
HOST_HIGH_CPU_LOAD
VM_STORAGE_SUBOPT_PERF
VM_CPU_SUBOPTIMAL_PERF
VM_CPU_THRESHOLD

resourceType	resourceId	alertType	activeTimestamp	updateDate	inactiveTimestamp	clo
MACHINE	12345	VM_STORAGE_SUBOPTIMAL_PERF	05/08/2017 08:09	05/08/2017 08:20	05/08/2017 08:30	4

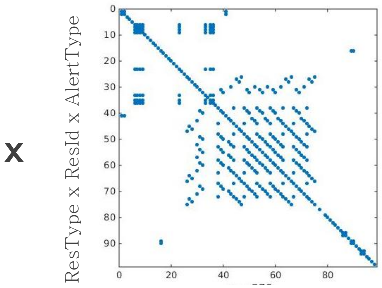
Computing correlations between alert time series:

- identified by <ResType x ResId x AlertType>
- using co-occurence measures: Jaccard, Pearson,...
- filtering according to topological constraints

Alert activation state



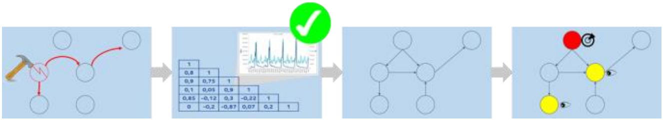
Corr(x,y) > threshold



ResType x ResId x AlertType

y

NOKIA

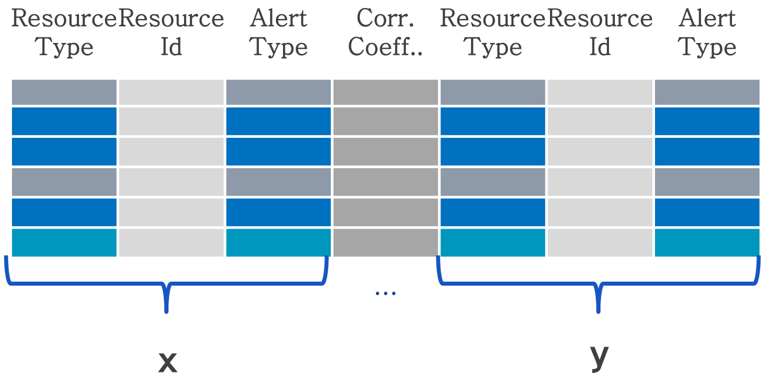


Causality inference from CB alert logs

Correlation Analysis (cont.)

Identify frequent co-occurrence patterns in terms of Types

- using Spark computing



Sparse representation of the Correlation matrix

Group by Types

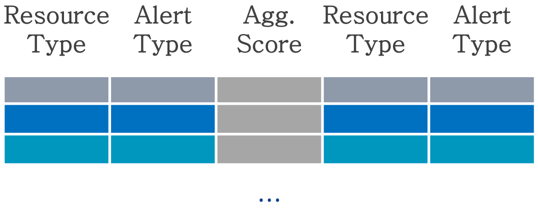
+

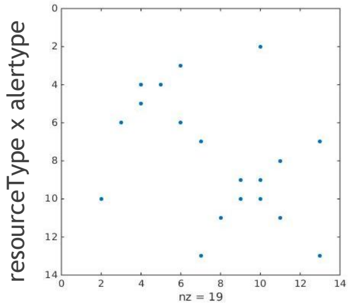
Aggregation functions (counter, mean)

+

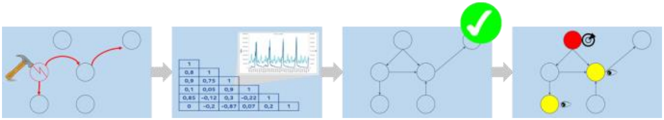
Threshold constraints

Inferred set of co-occurrence rules





resourceType x alertype



Causality inference from CB alert logs

Causal analysis

Transform co-occurrence rules into causality rules

Symmetric co-occurrence rules

Resource Type	Alert Type	Agg. Value	Resource Type	Alert Type
		…		

Lagged correlation Sensitivity tests

+

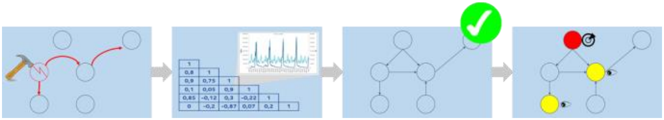
expert knowledge
on Resource & Alert Type
semantics (if available)

Suggested causality rules

Resource Type	Alert Type		Resource Type	Alert Type	Agg. Value
		→			
	…				
		~			

Maintain unresolved co-occurrences without causality

NOKIA



Causality inference from CB alert logs

Causality analysis (examples)

Correlation threshold: 0.9

resourceType	alertType		resourceType	alertType	Count	Mean
HOST	HOST_CEPH_PROBLEM	→	MACHINE	VM_STORAGE_SUBOPTIMAL_PERF.	5742	0.96
HOST	HOST_HIGH_CPU_LOAD	→	MACHINE	VM_CPU_SUBOPTIMAL_PERF.	47	0.98
HOST	RESOURCE_INVALID_STATE	~	MACHINE	VM_STORAGE_SUBOPTIMAL_PERF.	45	0.95

Correlation threshold: 0.7

resourceType						
HOST	HOST_CEPH_PROBLEM	→	MACHINE	VM_STORAGE_SUBOPTIMAL_PERF.	23490	0.86
MACHINE	FAILED_TO_GET_METRICS	~	MACHINE	VM_STORAGE_SUBOPTIMAL_PERF.	5953	0.72
HOST	RESOURCE_INVALID_STATE	~	MACHINE	VM_STORAGE_SUBOPTIMAL_PERF.	5525	0.77
HOST	HOST_CEPH_PROBLEM	~	HOST	RESOURCE_INVALID_STATE	232	0.77
HOST	HOST_HIGH_CPU_LOAD	→	MACHINE	VM_CPU_SUBOPTIMAL_PERF.	47	0.98
HOST	RESOURCE_INVALID_STATE	~	MACHINE	FAILED_TO_GET_METRICS	8	0.73
CLOUD_NODE	RESOURCE_INVALID_STATE	~	COMPONENT	RESOURCE_INVALID_STATE	6	0.82
CLOUD_NODE	METRICS_READING_ERROR	~	MACHINE	FAILED_TO_GET_METRICS	4	0.71
CLUSTER_INST.	MODULE_ERROR	→	CLUSTER_INST.	SYSTEM_ERROR	1	0.85

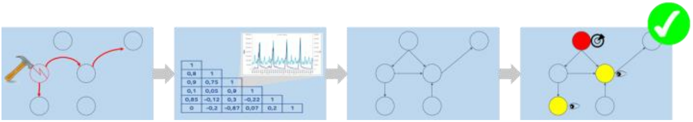
Correlation threshold: 0.5

resourceType						
MACHINE	FAILED_					
HOST	HOST_C					
HOST	RESOURCE_INVALID_STATE	~	MACHINE	VM_STORAGE_SUBOPTIMAL_PERF.	6608	0.75
HOST	HOST_CEPH_PROBLEM	~	HOST	RESOURCE_INVALID_STATE	232	0.77
HOST	RESOURCE_INVALID_STATE	~	MACHINE	FAILED_TO_GET_METRICS	60	0.64
HOST	HOST_CEPH_PROBLEM	~	MACHINE	FAILED_TO_GET_METRICS	58	0.56
HOST	HOST_HIGH_CPU_LOAD	→	MACHINE	VM_CPU_SUBOPTIMAL_PERF.	47	0.98
CLOUD_NODE	RESOURCE_INVALID_STATE	~	COMPONENT	RESOURCE_INVALID_STATE	6	0.82
CLOUD_NODE	METRICS_READING_ERROR	~	HOST	RESOURCE_INVALID_STATE	2	0.53
CLUSTER_INS.	MODULE_ERROR	→	CLUSTER_INS.	SYSTEM_ERROR	2	0.69

NOKIA

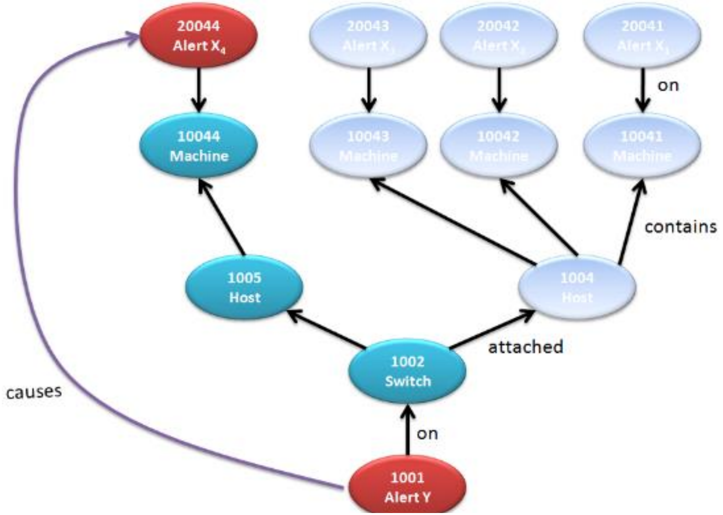
Causality inference from CB alert logs

Diagnosis



Topology based RCA templates [Nokia CloudBand / OpenStack Vitrage project]

- Expert rules **+ Inferred Rules** + dynamic instantiation of RCA templates => root causes



Dynamic instantiation of RCA templates

NOKIA

Inferring propagation scenarios from alarm logs (Radio Acess Network)

Complexity reduction and automated causal analysis

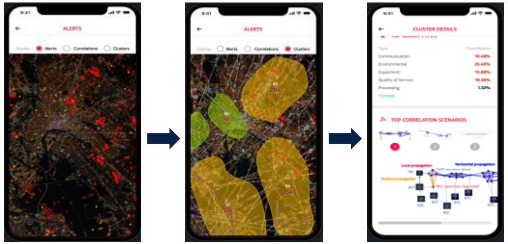
Goal: Reducing the time to dispatch KPI (time to resolve an incident)

Approach: Structural and Causal analysis of the alarm "chaos"

- Using statistical causal inference techniques enforced with exogeneous knowledge, e.g. physical/logical topology, or expertise

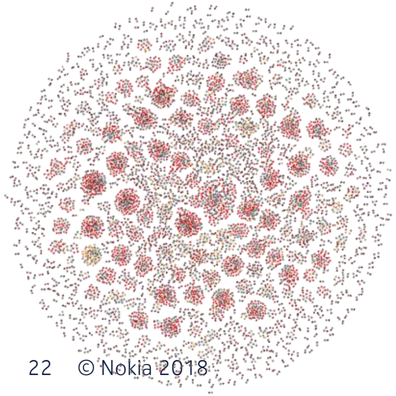
Impact: Number of possible explanations reduced by 10K

Presented at Global Analyst Forum 2018

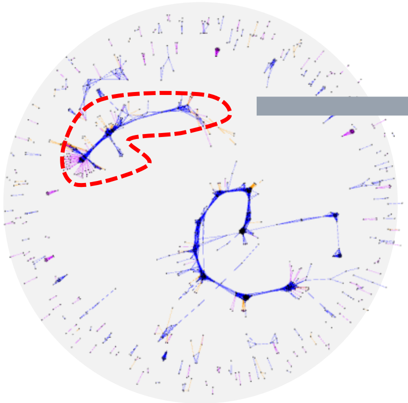


NSW DI demonstrator on a smartphone

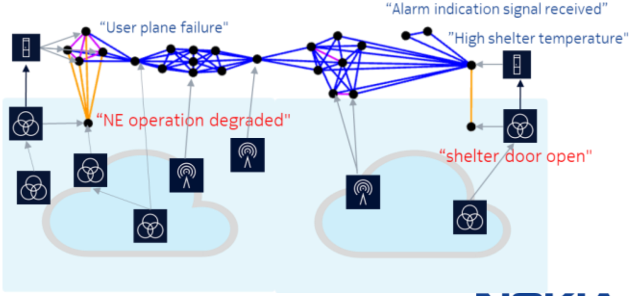
Alarm chaos



Reduced space



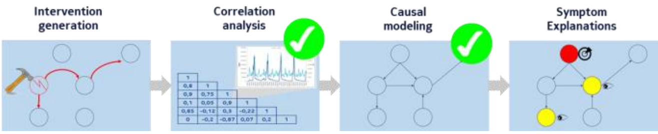
Zooming on an alert propagation scenario



NOKIA

Inferring propagation scenarios from alarm logs (2)

Complexity reduction and automated causal analysis



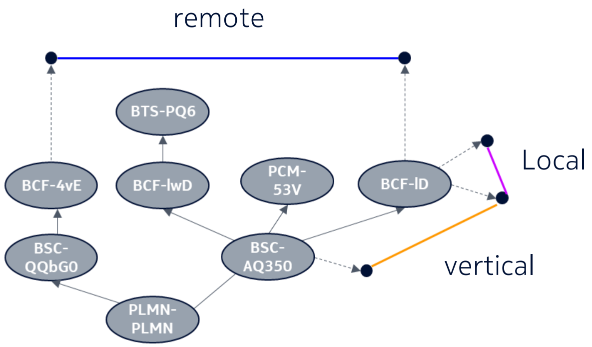
Dataset: large radio access network

- 200K network elements, 1 Million alarms every day (40 000 alarms/ hour), several hours per incident, up to 24 hours in some cases

Structural and Causal analysis of the alarm "chaos"

Using statistical causal inference techniques enforced with exogeneous knowledge injection (e.g. physical/logical topology, or expertise)

- Dynamic correlation graphs based on local/vertical/remote resource relations
- Graph summarization
- Number of possible explanations reduced by 10k



NOKIA

Root Cause Analysis of Future Networks

Summary of important challenges

Model-based approaches are challenges by the fact that in real systems models are often unknown!

- Fault injection and interventional approaches to causality discovery – *completeness, combinatorics*
- Observational analysis / statistical causal inference - *missing sufficient fault data volumes*
- Observational analysis assisted by some structural /topological knowledge - *partial topology knowledge*

Problem translation into the event space : adequate anomaly detection mechanisms

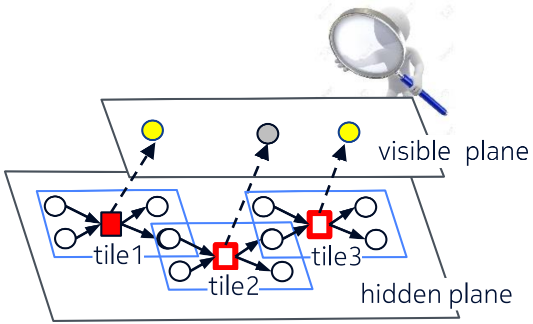
- Not specified in complex cloud-based environments
- Need for methods coupling anomaly detection and fault diagnosis

Challenges for Model Learning and Diagnosis

- High-dimensionality
- Dynamically reconfigurable systems, dynamically tracking the changes in the model
- Partially specified and partially observed models (controlled environments)
- Non-causal observation
- High-responsiveness : self-healing control loops

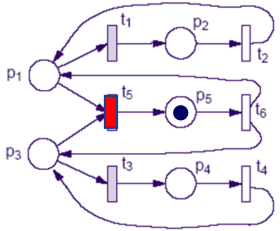
NOKIA

Fault model-based approach
Tile-based fault model



Tile-based model

elementary fault behaviors of various node types:
<pre-conditions, alarm/event, post-conditions>



Petri Net semantics

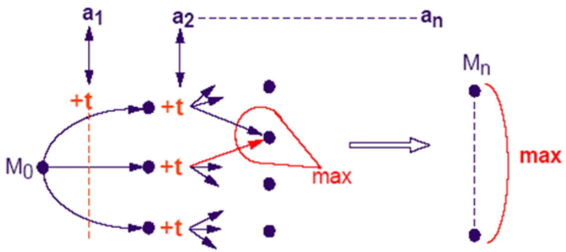
distributed state / local conditions
concurrency

Uncertainty representation

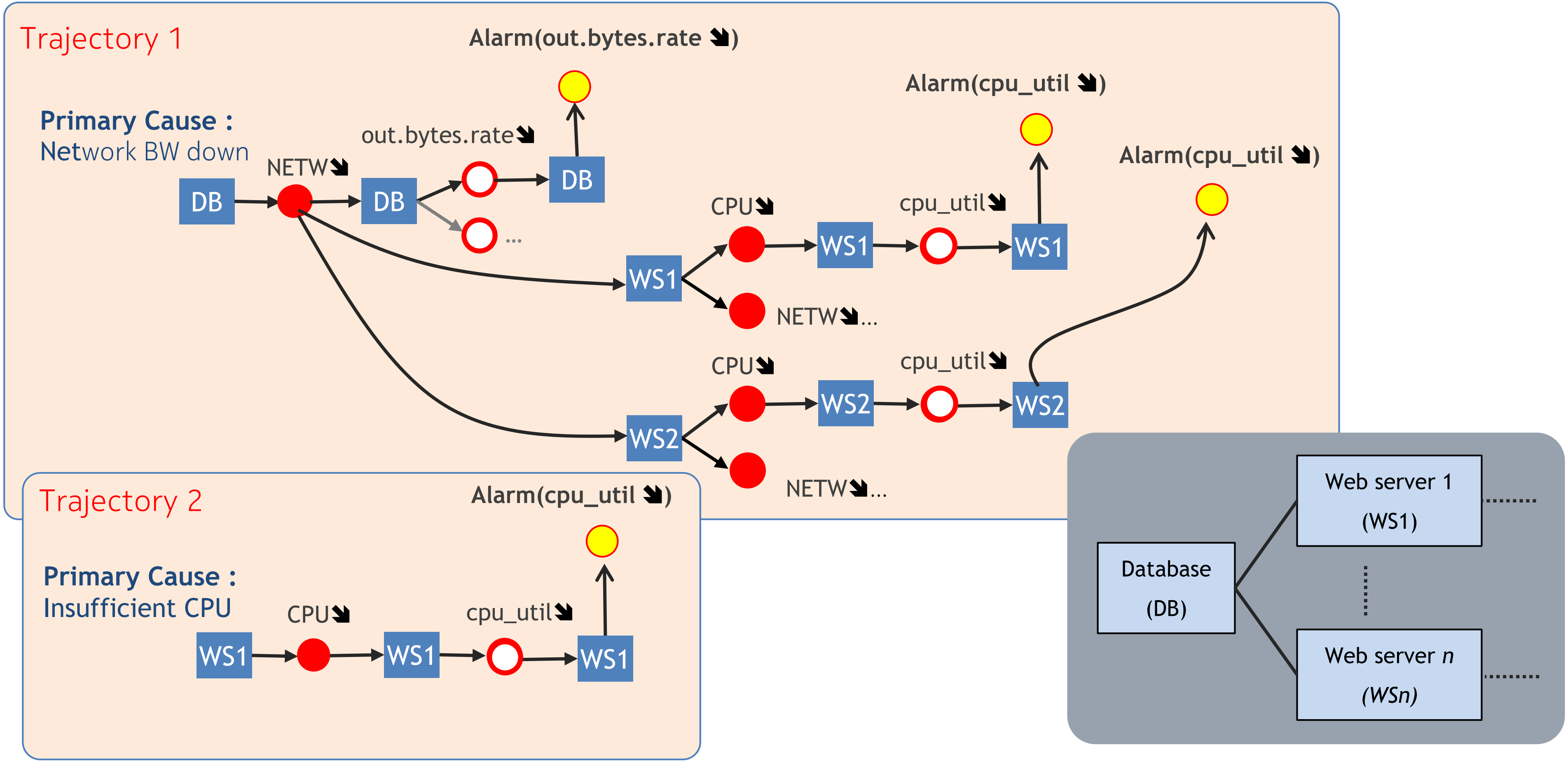
unobserved tiles
likelihood measure

Model execution : fault trajectory building & diagnosis

Max likelihood estimation / Viterbi algorithm variants
Stream-based puzzle assembling using tiles (Spark Streaming)



Trajectory building



Model discovery in controlled environments

Stimulus-based modeling for Cloud : Open questions

Verification & testing of derived tiles

- Model properties, coverage of real system faults and behaviors

Automatic learning of “complex” behaviors

- Forward propagation “Conflict” behaviors,
- “Synchronization” behaviors induced by the co-occurrence of multiple faults

Diagnosis algorithms under non-causal ordering of observed events

- On-line vs batch processing with sliding window, distribution vs stream analytics

Dealing with dynamically reconfigurable systems, e.g. auto-scaling resources

- Relevance of the tile model learned in the Sanbox => asymptotic analysis of correlations, parametric tile model ?
- (distributed) diagnosis algorithms for reconfigurable models (changing the number of resources)

Dealing with partially modeled systems, e.g. unknown types of resources (not covered by the model)

- On-line micro-stimulus approach, …?

