iForesight

Intelligent CloudOps

iForesight *is an* ***intelligent new tool*** *aimed at SRE cloud maintenance teams. It enables them to* ***quickly detect anomalies*** *thanks to the use of* ***artificial intelligence*** *when* ***cloud service are slow or unresponsive****.*

HUAWEI ENTERPRISE A BETTER WAY

Mastering AIOps

using Deep Learning, Time-Series Analysis, and Distributed Tracing

Dr. Jorge Cardoso
Huawei Munich Research Center &
University of Coimbra

--jorge.cardoso@huawei.com--

2018.08.28

enterprise.huawei.com

HUAWEI TECHNOLOGIES CO., LTD.

· U C ·

DUSSELDORF, GERMANY
29–31 August
2018

SRE CON
EUROPE
MIDDLE EAST
AFRICA

HUAWEI

Abstract

Artificial Intelligence for IT Operations

In planet-scale deployments, the Operation and Maintenance (O&M) of cloud platforms cannot be done any longer manually or simply with off-the-shelf solutions. It requires self-developed automated systems, ideally exploiting the use of AI to provide tools for autonomous cloud operations. This talk will explain how **deep learning, distributed traces,** and **time-series analysis** (sequence analysis) can be used to effectively **detect anomalous cloud infrastructure behaviors** during operations to reduce the workload of human operators. The **iForesight** system is being used to evaluate this new O&M approach. iForesight 2.0 is the result of 2 years of research with the goal to provide an intelligent new tool aimed at SRE cloud maintenance teams. It enables them to quickly detect and predict anomalies thanks to the use of artificial intelligence when cloud services are slow or unresponsive.

Jorge Cardoso

GitHub | Slideshare.net | GoogleScholar

Dr. Jorge Cardoso is Chief Architect for Intelligent CloudOps at Huawei's German Research Center in Munich. Previously he worked for several major companies such as SAP Research (Germany) on the Internet of Services and the Boeing Company in Seattle (USA) on Enterprise Application Integration. He previously gave lectures at the Karlsruhe Institute of Technology (Germany), University of Georgia (USA), University of Coimbra and University of Madeira (Portugal). He recently published his latest book titled "Fundamentals of Service Systems" with Springer. His current research involves the development of the next generation of Cloud Operations and Analytics using AI, Cloud Reliability and Resilience, and High Performance Business Process Management systems. He has a Ph.D. in Computer Science from the University of Georgia (USA).

Interests Service Reliability Engineering, AIOps, Cloud Computing, Distributed Systems, Business Process Management

Positions in Industry BOEING SAP HUAWEI

AIOps

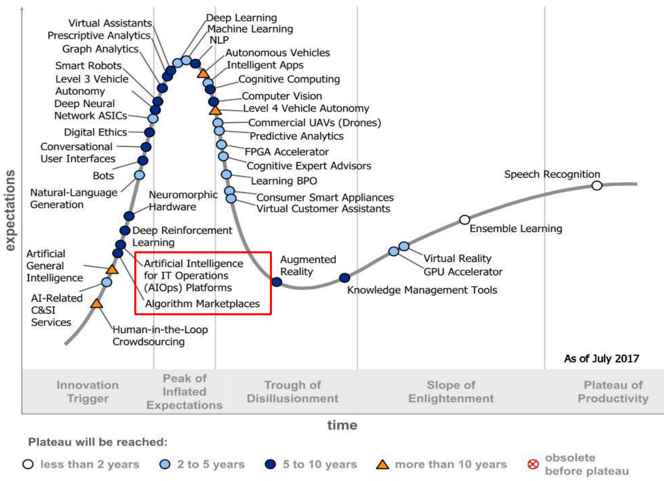
Artificial Intelligence for IT Operations

Every year the management of IT Operations is more complex

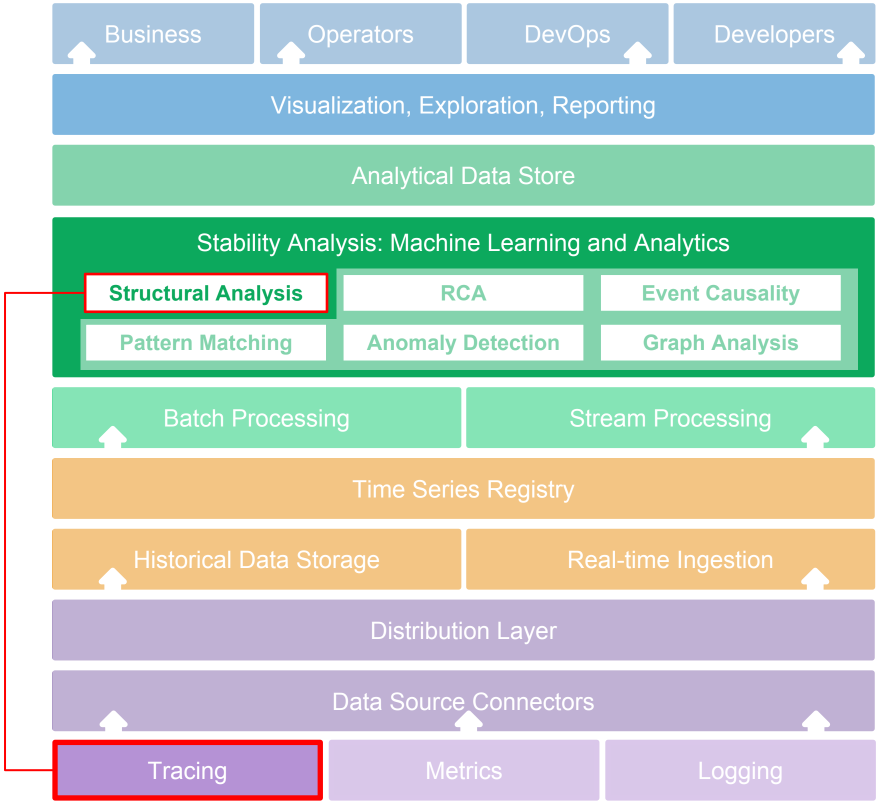
- **Increase in IT size, and event and alert volumes**
- **Digitalization with cloud, mobile, microservices**
- **Edge, IoT, serverless, …**

To deal with this complexity, businesses are turning to AI to automate incident management across production stacks, including application, infrastructure, and monitoring tools.

Figure 1. Hype Cycle for Artificial Intelligence, 2017



Source: Gartner



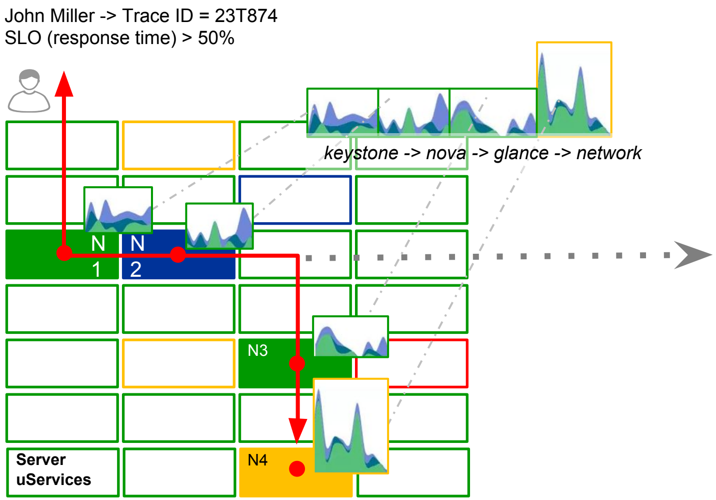
Reference architecture for AIOps platforms

HUAWEI

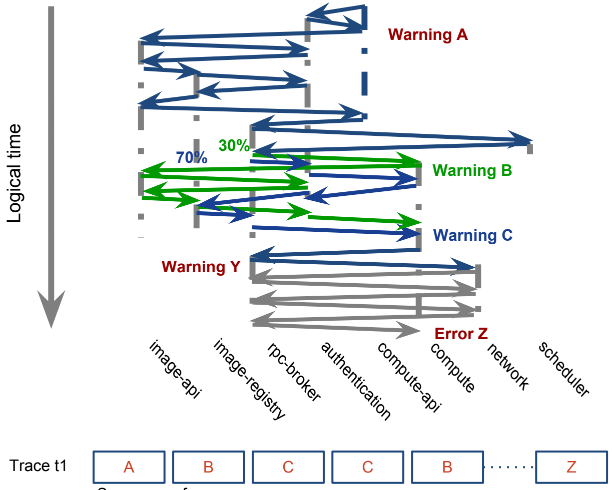
Objective

Detect Structural Changes of Distributed Traces

- ► **Scenario** Service requests' response time increased 50% when compared to 12h ago.
- ► **Observation** Distributed traces' structure has changed for the past 30 minutes
- ► **Root cause** Traces show a 3*retry to service A, before calling service B



Distributed Traces [1]



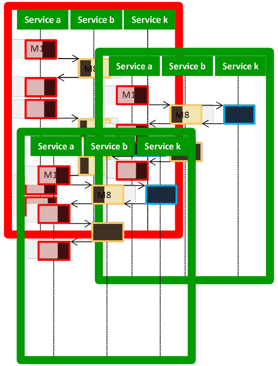
[1] B. H. Sigelman, L. A. Barroso, M. Burrows, P. Stephenson, M. Plakal, D. Beaver, S. Jaspan, C. Shanbhag, Dapper, a Large-Scale Distributed Systems Tracing Infrastructure, 2010, Google, Inc.

Objective

Detect Structural Changes of Distributed Traces

User Commands ▶ **Distributed Traces** ▶ **Encoding** ▶ **Detect Normal/Anomalous Traces**

host list
host show
hypervisor list
hypervisor show
hypervisor stats show
image add project
image create
image delete
image list
image remove project
image save
image set
image show
ip fixed add
flavor create
flavor delete
flavor list
flavor set
flavor show
flavor unset
…



Encoding

alb5g1jksao98aj8ik5e w21
ab05g1jksao98ajkk5ew 21
ab05g1jksalm8aj8ik5e w21
ab05g1jksao98aj8i--- ---

Detect Normal/Anomalous Traces

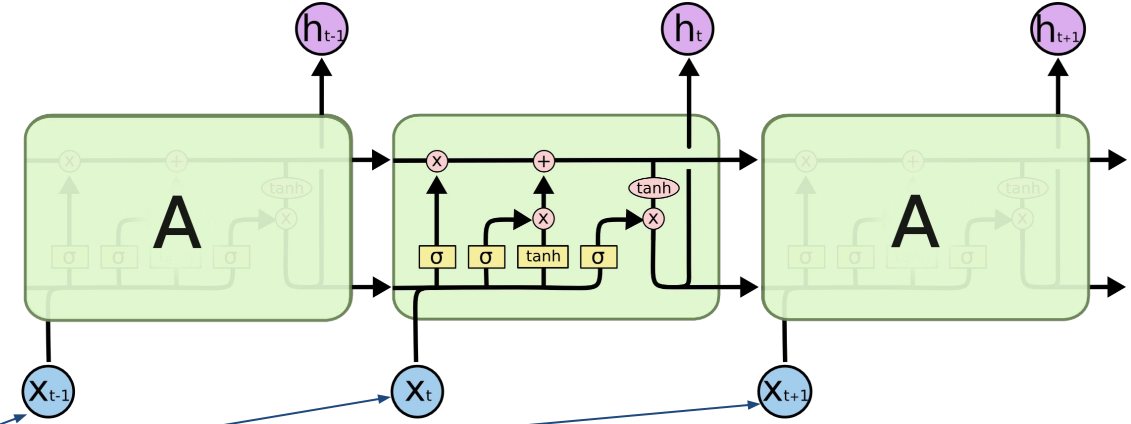
alb5g1jksao98aj8ik5e w21
ab05g1jksao98ajkk5ew 21
ab05g1jksalm8aj8ik5e w21
ab05g1jksao98aj8i--- ---

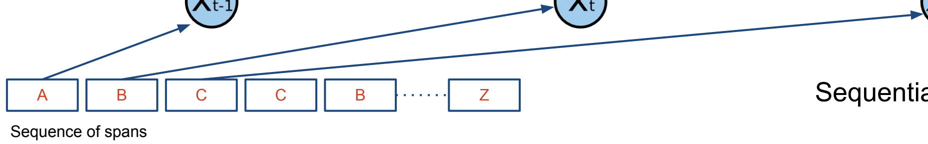
Long Short Term Memory (LSTM)

LSTMs [1] are models which capture sequential data by using an internal memory. They are very good for analyzing sequences of values and predicting the next point of a given time series. This makes LSTMs adequate for Machine Learning problems that involve sequential data (see [3]) such **speech recognition**, **machine translation**, **visual recognition and description**, and **distributed traces analysis**.

$$f_t = \sigma\,(X_t * U_f + H_{t-1} * W_f)$$
$$\bar{C}_t = \tanh\,(X_t * U_c + H_{t-1} * W_c)$$
$$I_t = \sigma\,(X_t * U_i + H_{t-1} * W_i)$$
$$O_t = \sigma\,(X_t * U_o + H_{t-1} * W_o)$$

$$C_t = f_t * C_{t-1} + I_t * \bar{C}_t$$
$$H_t = O_t * \tanh\,(C_t)$$





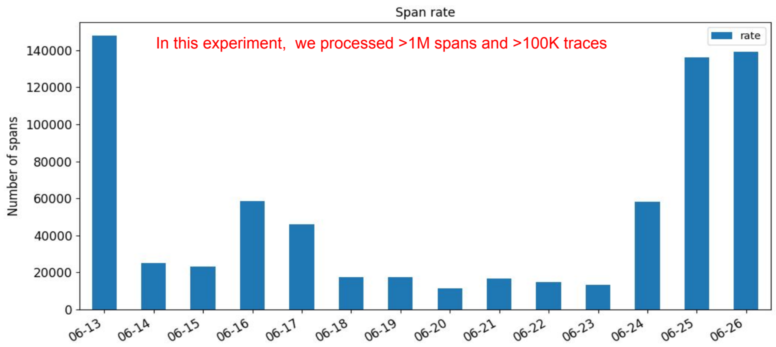
Sequential processing using LSTM [2]

[1] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.
[2] Sequential processing in LSTM (from: http://colah.github.io/posts/2015-08-Understanding-LSTMs/
[3] LSTM model description (from Andrej Karpathy. http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Structure Formalization

Trace and Span Representation

- **Training**
  - Collect traces
  - Parse and abstract spans
  - Feed span lists to deep neural network
  - Train the network with traces
- **Operation**
  - Collect real-time traces
  - Parse and abstract spans
  - Feed span lists to deep neural network
  - Retrieve probability matrix
  - Analyze matrix to decide on the severity of the anomaly



Partial view of the spans collected

Training

Sequence of spans

Trace t1	91	12	43	76	....	24
Trace t2	04	93	63	25	....	35
....						
Trace tn	38	18	44	24	....	61

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}.$$

Operation

Trace sampled

91	12	43	76	....	24

Predicted reconstruction

91	12	43	76	....	24

We analyze historical traces and collect the spans generated to create a structure containing span types (symbols) using label encoding. The network is training with the encoded traces. During operation, an LSTM network takes as input a sequence of symbols representing a new real-time trace, e.g., 1: “72”, 2: “83”, 3: “54”, …n: “23”. The output is a matrix with the probability of each symbol to belong to a learned structure. The analysis of the probability matrix enables to reason on if a trace should be mark are normal or anomalous.

HUAWEI

Overall Approach

From Encoding to Detection

Raw Span Events


```
{"TID": "3db29190256033ac83XXXXXX", "name": "get", "ts":
1528848XXXXXX, "id": "efa7fe0eb80dXXXX", "response_time":
14XXXXX, "ba": [{"key": "httpxxxxxxx_xxxx", "value": "200"},
{"key": "httpxxxx", "value": "https://
XXXXXX15638ec8de7423ea47199e2134/v2/
XXXXXX8691a747f98ed65f0c939731bf/xxxxxxx/yyyyyy?limit=200"},
{"key": "protocol", "value": "XXXX"}], "a": [{"timestamp":
1528848839823123, "value": "XX", "endpoint": {"servicexxxx":
"XXX", "ipv4": "xxx.75.xxx.253"}}, {"timestamp":
1528848839XXXXX, "value": "xx",     }]}
```


1

Encoded Traces

traceId	
00804a4929d70ee534dd974d716d47e9	[10, 10, 17, 6, 6, 8, 8, 7, 7]
008206bdf49f583940add552c232e050	[10, 10, 17, 6, 6, 8, 8, 7, 7]
00882090689180e7a7ed3e88185178b7	[10, 10, 17, 6, 6, 8, 8, 7, 7]
0089be97bd18e3ad9e482fb7bb6be86b	[10, 10, 17, 6, 6, 8, 8, 7, 7]
008bfa421e25e03e4344debda914f339	[10, 10, 17, 6, 6, 8, 8, 7, 7, 16]
008c0568c2554f096796bd10ba7c81a3	[11, 3, 3, 3, 10, 2, 10, 2, 10, 17, 5, 5, 16, ...
008fe0fa711a6d0b7c26b20fd731fa54	[10, 10, 17, 6, 6, 8, 7, 8, 7]
009cc4b398bfd04016eef2a95d125fc3	[23, 10, 10, 10, 19, 10, 10, 10, 10, 17]
009f5651ed705aa00029cf89cc0c84c3	[10, 10, 17, 6, 6, 8, 8, 7, 7]
00a12fd979a801ac8dd4252152bf9b37	[10, 10, 17, 6, 6, 8, 8, 7, 7]
00a186e1adce5436be8d83ace978ec67	[23, 10, 10, 10, 19, 10, 10, 10, 10, 10]

2

Encoding scheme

- The encoding scheme pads each sequence of inputs with zeros, up to a pre-defined maximum length.
- This allows to pre-allocate a chain of LSTM units of a specific length
- We also pass information about the sequence lengths. This is important for not treating the zero padding as actual inputs, and from injecting the error signal at the right unit in the sequence during back propagation.

Hot encoding

- A one hot encoding enables to represent categorical variables as binary vectors.
- Categorical values are mapped to integer values.
- Each integer value is represented as a binary vector that is all zero values except the index of the integer, which is marked with a 1.

LSTM Network

```python
# Create a simple LSTM model (sequence to sequence mapping)

def build_model(input_dim):
    model = Sequential()
    # units=100: Positive integer, dimensionality of the output space.
    model.add(LSTM(100, dropout=0.2, recurrent_dropout=0.2, return_sequences=True, input_shape=(20, input_dim)))
    model.add(LSTM(100, dropout=0.2, recurrent_dropout=0.2, return_sequences=True))
    model.add(LSTM(100, dropout=0.2, recurrent_dropout=0.2, return_sequences=True))

    # A Dense layer is used as the output for the network.
    model.add(TimeDistributed(Dense(input_dim, activation='softmax')))
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model
```

3

Network Training

```
Epoch 00087: val_loss improved from 0.01071 to 0.01064, saving
model to model.h5
Epoch 88/120
92101/92101 [==============================] - 70s 763us/step
- loss: 0.0114 - acc: 0.9960 - val_loss: 0.0106 - val_acc: 0.9
965

Epoch 00088: val_loss improved from 0.01064 to 0.01057, saving
model to model.h5
Epoch 89/120
92101/92101 [==============================] - 70s 760us/step
- loss: 0.0114 - acc: 0.9960 - val_loss: 0.0108 - val_acc: 0.9
963

Epoch 00089: val_loss did not improve from 0.01057
Epoch 90/120
92101/92101 [==============================] - 70s 764us/step
- loss: 0.0114 - acc: 0.9961 - val_loss: 0.0105 - val_acc: 0.9
965
```

4

Normal (green) and Anomalous (red) Traces Detected

```
Diff idx=16, y=33, yhat=77, alternatives top-4: [ 40  31 100  77] [NOK]
Problematic element: ZZZZZZZZZZZZZZ
traceId: 50cffdb15105d0bcXXXXXXXXXXXXXXXXXX
x:    [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 48 77 33 33 37]
y:    [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 77 33 33 37  0]
yhat: [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 79 77 31 31  0]
------------------------------------------------------------------
Diff idx= 7, y=80, yhat=30, alternatives top-4: [106  65  70  30] [NOK]
Problematic element: YYYYYYYYYYYYYY
traceId: b51b25866e351ee9XXXXXXXXXXXXXXXXXX
x:    [ 0  0  0  0  0  0  0 57 80 77 77 80 77 57 79 32 79 20 20 79]
y:    [ 0  0  0  0  0  0  0 80 77 77 80 77 57 79 32 79 20 20 79  0]
yhat: [ 0  0  0  0  0  0  0 30 77 77 77 77 77 79 32 79 20 20 79  0]
```

```
------------------------------------------------------------------
x:    [ 0  0  0  0  0  0  0  0  0  0  0 29 29 29 87 87 90 88 90 88]
y:    [ 0  0  0  0  0  0  0  0  0  0  0 29 29 87 87 90 88 90 88  0]
yhat: [ 0  0  0  0  0  0  0  0  0  0  0 29 29 87 87 90 90 90 88  0]
------------------------------------------------------------------
x:    [ 0  0  0  0  0  0  0 45 70 65 77 45  1  1 70 45 74  3 75 76]
y:    [ 0  0  0  0  0  0  0 70 65 77 45  1  1 70 45 74  3 75 76  0]
yhat: [ 0  0  0  0  0  0  0 70 65 45 45  1  1 70 45 74 10 75 76  0]
------------------------------------------------------------------
```

Keras and **Tensorflow**

- **LSTM** network
- **dropout layer of 0.2** (high, but necessary to avoid quick divergence)
- **Softmax** activation to generate probabilities over the different categories
- Because it is a classification problem, loss calculation via **categorical cross-entropy** compares the output probabilities against one-one encoding
- **ADAM** optimizer (instead of the classical stochastic gradient descent) to update weights

Tensorflow tutorial: https://github.com/campdav/text-rnn-tensorflow
Keras tutorial: https://github.com/campdav/text-rnn-keras

HUAWEI

Distributed Trace Anomaly Detection

Results

Goal: Classify distributed traces as normal or abnormal

- **Benefits**
  - Very high accuracy in detection: **99.73%**
  - Extremely fast detection: **O(n)**
  - The technique outperforms existing approaches
    e.g., sequence matching, petri nets, clustering

- **Limitations**
  - Requires a special (not trivial) handling when using recurrent neural networks, like LSTMs
  - Long traces may require very long training times
    e.g., back propagation across long sequences may result in vanishing gradients...

- **Improvements**
  - Truncate traces (remove steps from the beginning or the end)?
    e.g., Lower the accuracy of predictions
  - Summarize traces (remove well-known subtraces)?
    e.g., focus on most relevant parts of a trace while reducing their length
  - Evaluation the prediction of remaining time to completion (see [Tax, 2017])
  - Operationalize LSTM with other sequential data (see [Du, 2017])

[Tax, 2017] Predictive business process monitoring with LSTM neural networks, Tax, Niek and Verenich, Ilya and La Rosa, Marcello and Dumas, Marlon, Proceedings of the 29th International Conference on Advanced Information Systems Engineering, 2017, 477--492, Springer.
[Du, 2017] M. Du, F. Li, G. Zheng, and V. Srikumar. 2017. DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17). ACM, 1285-1298.

HUAWEI

HUAWEI ENTERPRISE ICT SOLUTIONS A BETTER WAY