WWW.IARIA.ORG

PANEL

Tendencies in Monitoring and Protecting Large (Cloud) Systems

Complexity

Three complementary views

- **Infrastructure/Networking**

  monitoring, measurements, correlations

- **Applications/Services**

  protection-oriented design

- **User ?**

  behavior

Lanes

- Traffic models
- Anomaly detection
- Measurements
- Policy/decision paradigms
- Protection
- System design
- Proofs, Validation
- User models

Small/medium/Big systems

Dynamic/Static systems

Panelists

Panelists

- **DongJin Lee, The University of Auckland, New Zealand**
- **Florian Kammüller, TU-Berlin, Germany**
- **Petre Dini, Concordia University / IARIA**

ICIMP 2010 (May 9 – 15 - 2010, Barcelona, Spain)
Tendencies in Monitoring and Protecting Large (Cloud) Systems

Panel Discussion

Difficulties in understanding Internet traffic behaviours

DongJin Lee, PhD
dongjin.lee@auckland.ac.nz

Department of Computer Science
The University of Auckland
New Zealand

Introduction

- Importance in Traffic Monitoring
  - [ISO] FCAPS: Fault, Configuration, Accounting, Performance, Security
  - *"..If we don't measure it, we don't know what's happening.."*
  - Interests by various groups of network operators
- Particular interests from commercial operators
  - Filling the bandwidth
  - *Always-On-P2P traffic* (and many others)
  - Choices of Increasing / restricting bandwidth
- Understanding traffic behaviours
  - Protocol trend analysis and Profiling
  - Prioritization and malicious detection, etc

Traffic behaviours

- Straightforward
  - Packet statistics (e.g., connections, size in bytes, duration)
  - Analyses (e.g., normal/abnormal traffic, bandwidth constraints)
- Method
  - Deep Packet Inspection (payload)
  - Behaviour Inspection (flows)
- Difficult
  - Complexity (e.g., protocol evolution, incorrect protocol)
  - Grey areas (e.g., In between yes/no, correct/Incorrect)

Discussion 1

- Deep Packet Inspection:
  - Accurate
  - Appliance
  - Privacy issues
  - Widely used (IDS, firewall)
- Behaviour Inspection:
  - Accurate and Increasing
  - Fast path
  - No privacy issue
  - Widely researched

Doubtful

- Accurate
- Processing demand
- Privacy issues (which layers?)
- Widely used (IDS, firewall)

Doubtful

- Accurate enough?
- Can be processing demand
- Privacy issues with accuracy
- Just Research?

Fast, Cheap and Reliable (Pick Two)

Attributes (1)

Table 2: Discriminators and Definitions

Number	Short	Long
1	Server Port	Port Number at server; we can establish server and client ports as we limit ourselves to flows for which we see the initial connection set-up.
2	Client Port	Port Number at client
3	min_IAT	Minimum packet inter-arrival time for all packets of the flow (considering both directions).
4	q1_IAT	First quartile inter-arrival time
5	med_IAT	Median inter-arrival time
6	mean_IAT	Mean inter-arrival time
7	q3_IAT	Third quartile packet inter-arrival time
8	max_IAT	Maximum packet inter-arrival time
9	var_IAT	Variance in packet inter-arrival time
10	min_data_wire	Minimum of bytes in (Ethernet) packet, using the size of the packet *on the wire*.

[1]

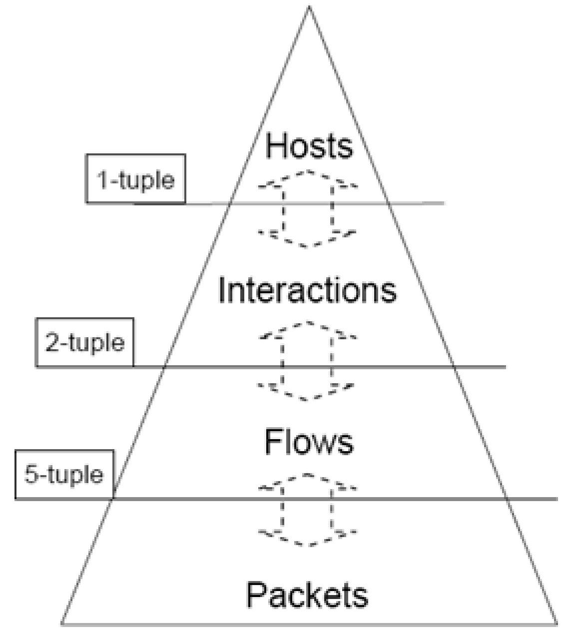
- 248 discriminators (attributes), using *only* the packet and flow attributes
- Much more attributes to be extracted at higher level

[1] Andrew W. Moore, Denis Zuev, Michael Crogan "**Discriminators for use in flow-based classification**", Technical Report, RR-05-13, Department of Computer Science, Queen Mary, University of London, August, 2005, http://www.cl.cam.ac.uk/~awm22/publications/RR-05-13.pdf

Attributes (2)

- A single IP host can produce multiple *interactions,*
  e.g., [**A** -> B], [**A** -> C], [**A** -> D]

- Each interaction actually consists of multiple *flows,*
  e.g., (A:80 -> B:2221)
  (A:80 -> B:2222)
  (A:80 -> B:2223)



> packets to 5-tuple **flow** [srcIP, dstIP, srcPort, dstPort, Protocol]
> 5-tuple flows to 2-tuple **interaction** [srcIP, dstIP]
> 2-tuple interactions to 1-tuple (IP) **host** [srcIP]

- A lot more *attributes* if considering 'interactions' and 'hosts'
  - Especially important for profiling emerging applications

Discussion 2

- Difficulties in Behaviour inspection
  - Too many attributes / metrics
  - Too many algorithms / methodologies to perform
  - Too many technical / statistical tweaks
- Many should ‘work’ okay
  - Mix and match (hopefully it *should* work reasonably)
  - How many traffic patterns?
  - Trace selections (we may need ‘Netflix’-like prize)
- Cats and Mice
  - Applications evolve, Detections evolve, Applications evolve…
  - Encryption (*everything*) the final solution? Then how to really profile?

Summary

- Difficulties in monitoring
  - Traffic classification (accuracy, how and what to use in attributes)
  - Meta traces for interest groups
- More difficulties (out of scope here)
  - Net Neutrality
    - ‘...don’t touch my traffic...’
    - ‘...prioritizing streaming traffic...’
  - Privacy and Advertisement
    - ‘...we want to know about you...’
    - Non-intrusive approach ever possible?
- When to stop monitoring?
  - When nobody cares

1. Panel: Tendencies in Monitoring and Protection

Florian Kammüller

Institut für Softwaretechnik und Theoretische Informatik

ICIMP 2010

Barcelona

11. May 2010

My Angle on Monitoring and Protecting Large Cloud Systems

My field: (Formal) Security Engineering

- Discipline to design and develop secure systems
- Methods: (Formal) Models

!!! Need to Integrate Monitoring and Protection as Requirement

My Interest (see tomorrow's talk)

- Develop ASP$_{fun}$ a calculus for
  - functional
  - active objects
  - distributed
  - plus typing
  - Formal language development in Isabelle/HOL

$\Longrightarrow$ Good security properties

$\Longrightarrow$ Privacy enforcement by restricting information flows

$\Longrightarrow$ Prototype for ASP$_{\text{fun}}$ in Erlang

ASP$_{fun}$– Asynchronous Sequential Processes – functional

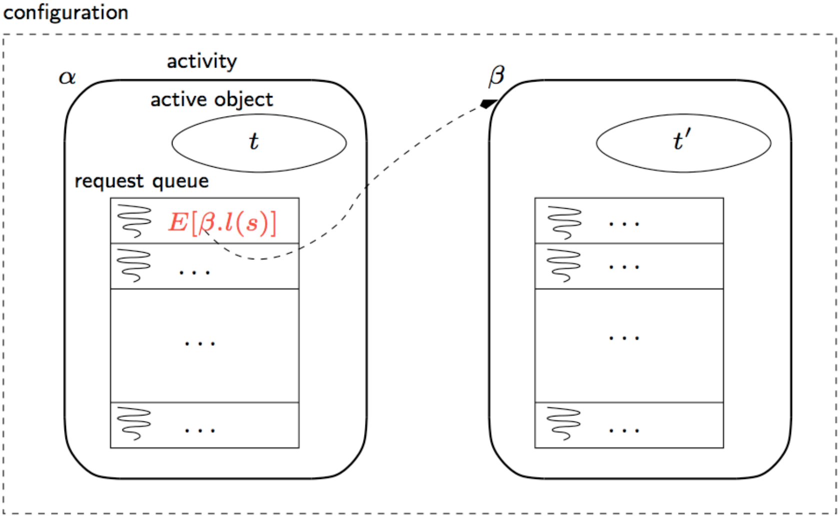
- ProActive (Inria/ActiveEON): Java API for active objects

- New calculus ASP$_{fun}$ for ProActive
- Asynchronous communication with *Futures*
  - Futures are *promises* to results of method calls
  - Futures enable asynchronous communication

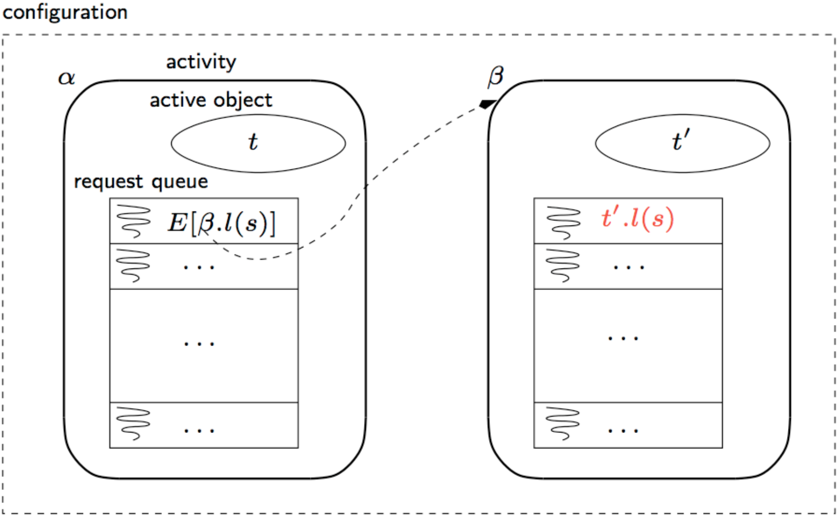
$\Rightarrow$ ASP$_{fun}$ avoids deadlocks when accessing futures

ASP$_{fun}$

ASP$_{fun}$: at a glance

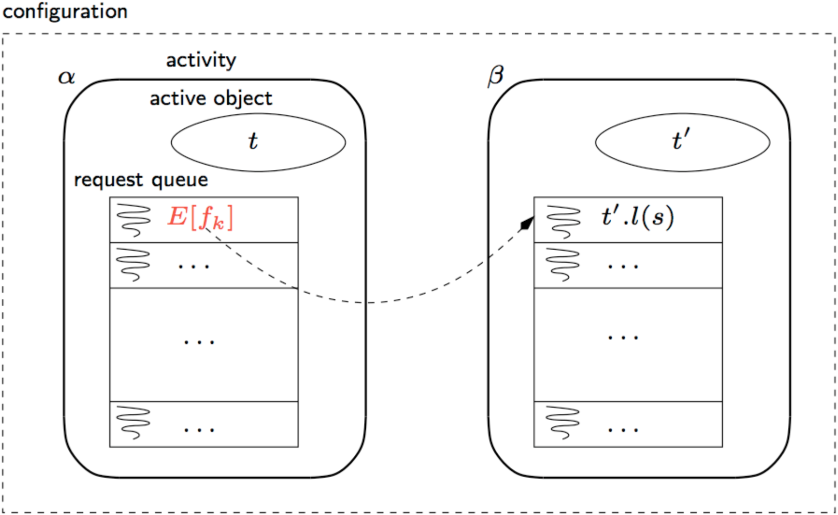


ASP$_{fun}$: at a glance



ASP$_{fun}$

ASP$_{fun}$: at a glance



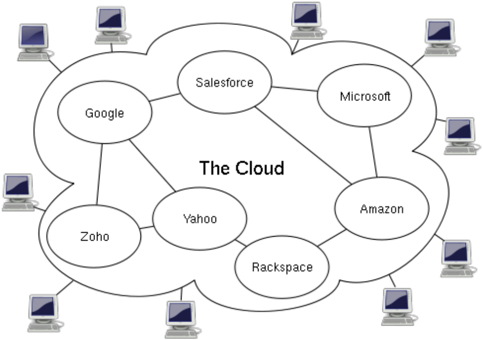
Tendencies in Distributed Systems

Cloud computing should not be confused with the following models

- Client-server: basically any distributed application distinguishing service-providers and service-requesters
- Grid computing: distributed, parallel computing; super-computer is a loosely coupled cluster
- Peer-to-peer: distributed architecture, no central coordination, participants can be suppliers and consumers (contrast client-server)

Cloud computing

- Idea: Rent usage of physical infrastructure from providers
- Architecture: cloud components communicate via API's, usually Web services.



$\Longrightarrow$ Grid $\subseteq$ Cloud $\subseteq$ P2P

Cloud Security

Security Issues associated with the cloud, three general areas (Wikipedia)

- Security and Privacy
  - Data protection
  - Identity management
  - Physical and personnel security
  - Availability
  - Application security (firewall, auditing)
- Compliance
- Legal and contractual issues

Positioning

Clouds give up on physical location of computing: How can we guarantee security goals?

- Security goals: their usual Enforcement $\Longrightarrow$ New problems
  - Integrity (authentication, data integrity): Security portocols, Cryptography, physical protection

  $\Longrightarrow$ Key distribution (Identity of a provider)
  - Confidentiality : Cryptography, physical protection

  $\Longrightarrow$ Where do we keep keys?
  - nonrepudiation (legal issues): Audit trails, logs

  $\Longrightarrow$ Where do we (physically) keep logs?

Cloud computing is between Grid and Peer-to-Peer: no physical protection, (much less) security