Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Πανεπιστήμιο Θεσσαλίας

Παρουσίαση της Ερευνητικής Θεματολογίας και της Προόδου των Διδακτορικών Διατριβών των Υποψήφιων Διδακτόρων

Πέμπτη 6 Μαρτίου 2014

Υποψήφιος Διδάκτορας
Κατσαλής Κώστας

Αρχιτεκτονικές Δικτύων προσανατολισμένες στην Παροχή Υπηρεσιών

Επιβλέποντες Καθηγητές:
• Αικατερίνη Χούστη
• Λέανδρος Τασιούλας
• Γιάννης Βινιώτης

Email: kkatsalis@inf.uth.gr
Stochastic Control in Service Oriented Architectures

- Service Oriented Architectures
- Software Defined Networks
- Stochastic Control
Service-oriented architecture is a design paradigm to build computer software in the form of services.

- A methodology to build applications
- Adopted by all large software providers (IBM, Microsoft, etc).
- We don’t care how a service is implemented. We only care how to use it.
SOA is a form of technology architecture that adheres to the principles of service-orientation.

• **Loose coupling:** minimum dependencies between services.
• **Service contract:** Formal definition of service, Communications agreement.
• **Autonomy:** Control over the logic they encapsulate.
• **Abstraction:** Beyond what is described in the contract, services hide implementation logic.
• **Reusability:** Logic is divided into services with the intention of promoting reuse.
• **Composability:** Collections of services.
• **Statelessness:** Minimum retaining information specific to an activity.
• **Discoverability:** Designed so that they can be found via available discovery mechanisms.
Service Oriented Architectures

Applications must be a Set of Services

- Web Applications can no longer be allowed to contain large masses of compiled executable code.
- Applications must be a set of services
Service Oriented Architectures

SOA Implementation Paradigms

- SOAP, RPC
- REST
- DCOM
- CORBA
- Web Services
- WCF
- ESB

How should services be designed?

How should messages be designed?

How should the relationship between services be defined?

How should service descriptions be designed?
The W3C defines a "Web service" as "a software system designed to support interoperable machine-to-machine interaction over a network."
The core infrastructure of an SOAIF can be typically provided by an **ESB**.

ESB is a software architecture construct which provides fundamental services for complex architectures via an event-driven and standards-based messaging engine (the bus).

Developers typically implement an ESB using technologies found in a category of middleware infrastructure products, usually based on recognized standards.
Service Oriented Architectures

- Software ESB implementations exist
- XML Appliances is an ESB hardware implementation.

**Capable of offloading overtaxed servers by processing XML, Web services, and other message formats at wire speed.**

**XML accelerators**
Accelerate XPath processing. Performance boost between 10 and 100 times in the number of messages per second that can be processed.

**Integration appliance.** Scalable, wire-speed, any-to-any message transformation, such as arbitrary binary, flat text and XML messages, which include COBOL copybook, CORBA, CICS, ISO 8583, ASN.1, EDI, and others

**XML security gateways**
also known as XML firewalls
PhD Research

Research on Algorithms that satisfy SLAs on various network metrics.

Stochastic control of processing power of servers hosting ESBs in order to provide Service Differentiation.

Traffic Engineering in Software Defined Networks

Network as a Service

Wireless Network Virtualization
What we investigate:

In overload situations, allocate a predefined percentage of CPU resources to every service domain/customer.
Service Differentiation

SLA

<table>
<thead>
<tr>
<th>CPU</th>
<th>Domain 1</th>
<th>..</th>
<th>Domain m</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overload</td>
<td>40%</td>
<td>..%</td>
<td>15%</td>
</tr>
</tbody>
</table>

- Unknown Arrival process Statistics
- Unknown Service Process Statistics
- No correlation of packet size with processing time
Dynamic Scheduling

Service Differentiation: Dynamic Scheduling

Limitations are jointly imposed by:

- The way that the applications are build and deployed.
- The coordination with the actual infrastructure needed to support these applications.
Approach: Stochastic Control

Dynamic Scheduling

Stochastic Buffering

Linear Approximations

SLA

<table>
<thead>
<tr>
<th>CPU</th>
<th>Domain 1</th>
<th>..</th>
<th>Domain m</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overload</td>
<td>40%</td>
<td>..%</td>
<td>15%</td>
</tr>
</tbody>
</table>

Sliding Window Dynamic Scheduling

CPU Overload

Linear Approximations
Dynamic Weighted Round Robin policies.

We define and prove a Class of dynamic policies achieving the SLA

1. All bound – round, non idling, negative drift policies converge

2. All bound – round, non idling, negative drift policies converge to the goal
All bound – round, non idling, negative drift policies converge

All bound – round, non idling, negative drift policies converge to the goal

**Bound round policy:** $P(L_n < \infty) = 1$

**Negative Drift policy:** $w_i = 0$ if $U_i(t) > p_i$
Dynamic Scheduling

Service Differentiation

SLA:

<table>
<thead>
<tr>
<th>Domain 1</th>
<th>Domain 2</th>
<th>Domain 3</th>
<th>Domain 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>20%</td>
<td>30%</td>
<td>40%</td>
</tr>
</tbody>
</table>

BGP

Effect of Service Rate

CPU Utilization

CPU Utilization Deviation

Time
To avoid communication overhead we keep the calculated weights constant for a fixed period of time $T$.

Weights remain the same during $T$.

After $T$ new weights are calculated for the new round according to BGP policy.
Bucket Queue System (BQS)
When an incoming request from a domain must be enqueued because the CPU is busy:

- If the domain is underutilized and the minimum queue size does not exceed a limit $q_c$, the policy forwards the request to the queue with the minimum queue size.

- In all other cases the request is forwarded to the bucket queue.
Dynamic Stochastic Buffering

Service Differentiation

<table>
<thead>
<tr>
<th>Domain 1</th>
<th>Domain 2</th>
<th>Domain 3</th>
<th>Domain 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>20%</td>
<td>30%</td>
<td>40%</td>
</tr>
</tbody>
</table>

![CPU Utilization Graph]

3/4/2014 22
Linear Approximations

Service Differentiation

\[ U^u_{PR,j} = \frac{T_e}{t_p} \times P^u_j - \frac{T_e - t_p}{t_p} \times (U^u_j + U^u_{q,j}) \]

Queue State factor

\[ U^u_{q,j} = a \times \left[ \min \left( b, \sum_{i=0}^{b} \sum_{j=0}^{b} q_{ij} \right) \right] \times U^u_j \]

Intuitive term and is used to model the fact that the requests that are in the apps queues at the time of decision will use CPU so we consider this usage as already taken.
Collaborations with professor’s Yiannis Viniotis team

Electrical and Computer Engineering
NC State University, Raleigh, USA

IBM, Raleigh, USA


• K.Katsalis, L.Tassiulas, Y.Viniotis “Throughput guarantees in cluster of XML Appliances”. 2nd Summer School on Service and Software Architectures, Infrastructures and Engineering (SSAIE 2010)
Further Research

Further Research: Software Defined Networks

- SOA drastically changed the way developers and the business looked at technology.
- Network, server, and storage to be tied together to enable the principles of SOA to be executed at the network layer.
- When SOA principles become applied to the networking layer, the network becomes more accessible, programmable, and flexible.
Further Research: SDNs

Further Research: Software Defined Networks

- Service & Management Layer
- Control Plane
- Data Plane
1. A cross-domain virtualization solution.

2. Seamless integration of access and metro network domains.

3. Creation and operation of infrastructure slices.

4. Support for dynamic end-to-end services.
CONTENT Project

Project Objectives

• Extend the operating range of the MVNO.

• “own” and operate virtual resources on the optical metro.

• Bridge the gap between the wireless access and the computational infrastructure.
CONTENT Project

Project Objectives

- Virtual operators will be able to access a tuple of virtual wireless, optical, IT resources.
- QoS and resilience guarantees for end-to-end connectivity.
- Framework for Cloud and mobile Cloud services.
• **GMPLS control plane**: A Path Computation Element (PCE) with a Sub Lambda Assignment Engine (SLAE) handles path-finding functions of the network.

• **GMPLS signalling protocols**: populate network information among the TSON nodes, and reserve the resource for setting up the TSON light-paths.

• **Layer 2 functions**: Direct control of the data plane of TSON nodes with FPGA.

• **Layer 1 functions**: Traffic is transported and switched over the allocated time slots.
CONTENT Project

Wireless Network

NITOS
CONTENT Project

End to end Network virtualization
CONTENT Project

Research Issues

- CONTENT Architecture
- Virtual Wireless Networks
- QoS and Service Differentiation Mechanisms
CONTENT Project

Research Issues

- Physical Infrastructure
- Wireless TSON Interface
- TSON-DC Interface
- Infrastructure Management
- Wireless Access
- Optical Metro
- Data Centers
- Protocol Manager
- WiFi/LTE Driver
- TSON Driver
- Wireless TSON Interface
- Optical Metro
- Cloud Manager System
- End-to-end Cloud+Net Service Orchestration
- Service Orchestration Layer
- Control
- Virtualization
- Resource Management
- Resource Abstraction
- Enhanced Network Functions (routing, mobility, TE, etc)
- Virtual Wireless CP
- Virtual Optical CP
- Virtual Resource 1
- Virtual Resource 2
- Virtual Resource n-1
- Virtual Resource n
- Cloud+Net Service Orchestration
- Layer
- End-to-end Cloud
- Net Service Orchestration
- Virtual Resource n-1
- Virtual Optical CP
- Virtual Wireless CP

3/4/2014
Virtual Wireless Network Control

Throughput Dependencies:
- Distance
- Same frequency another AP
- How many users share

Multiple interfaces available

802.11a
- 5 GHz
- OFDM
- 54Mbps
- Less range
  (20Mbps average
  throughput)

802.11b
- 2.4 GHz
- DSSS
- 11Mbps

802.11n
- 2.4/5 GHz
- MIMO
- OFDM

802.11g
- 2.4/5 GHz
- OFDM
- 54Mbps
  (22Mbps average
  throughput)

Wireless Node

LTE/ WiMAX

OpenFlow Tier

Feedback from optical

Feedback

Action

Network Tier

Data Plane

OMF Control Framework

Slices allocation:
Nitos Scheduler

Nitos Scheduler

OM Aggregator

Databases & other servers

Control Plane

UH Gateway

Geant
CONTENT Project

Research


Σας ευχαριστώ για την προσοχή σας!

Κατσαλής Κώστας
Email: kkatsalis@inf.uth.gr