End-to-End QoS on Shared Clouds for Highly Dynamic, Large-Scale Sensing Data Streams

Rafael Tolosana-Calasanz, José A. Bañares, Congduc Pham and **Omer F. Rana**

{rafaelt, banares}@unizar.es, congduc.pham@univ-pau.fr, o.f.rana@cs.cardiff.ac.uk

Universidad de Zaragoza
Zaragoza, Spain

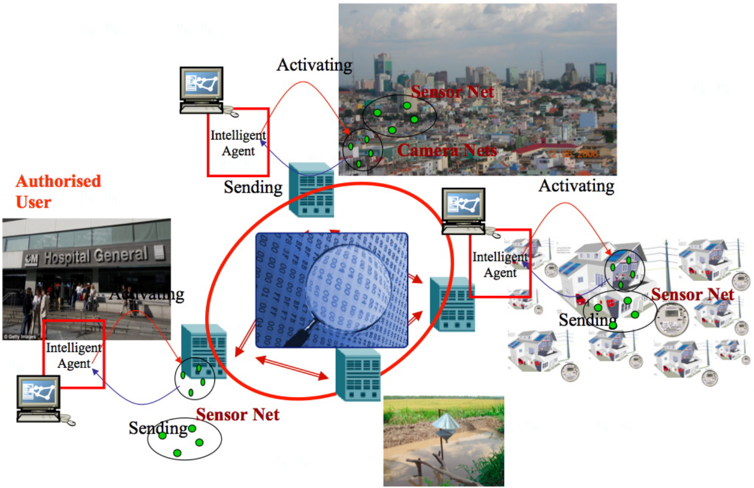
Université de Pau
Pau, France

Cardiff University
Cardiff, United Kingdom

May 13^{rd}, 2012

DPMSS 2012 – CCGrid 2012
Ottawa, Canada

Sensor Data Aggregation



Approach & Focus

Adaptive infrastructure for sensor data analysis

- **Multiple** concurrent data streams
- **Variable** properties: rate and data types; various processing models
- Support for **in-transit** analysis
- Support for **admission** control & flow **isolation** at each node

Approach & Focus

Adaptive infrastructure for sensor data analysis

- **Multiple** concurrent data streams
- **Variable** properties: rate and data types; various processing models
- Support for **in-transit** analysis
- Support for **admission** control & flow **isolation** at each node

Key focus

- Develop a **model** and a **system architecture** to support adaptive stream processing
- Model is **executable** directly and can be combined with monitoring
- Model behaviour can be modified through an external **controller**

Zaragoza

Outline

1. Background
   - Petri nets
   - Reference nets
2. In-transit Analysis
3. System Architecture & Model
   - Token Bucket
   - Autonomic Data Transfer Service
4. Evaluation
5. Conclusions and Future Work

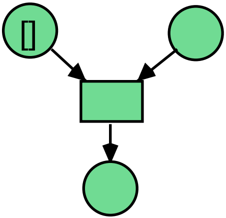
Outline

1. Background
   - Petri nets
   - Reference nets
2. In-transit Analysis
3. System Architecture & Model
   - Token Bucket
   - Autonomic Data Transfer Service
4. Evaluation
5. Conclusions and Future Work

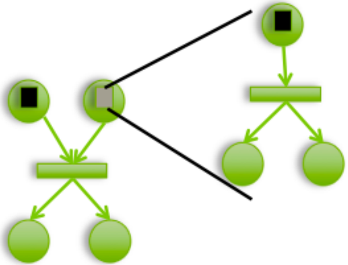
Petri nets

Characteristics

- directed bipartite **graph**
- **2** types of **nodes**: places and transitions
- **arcs**: place-transition, transition-place
- **tokens**: move on the graph
- **static** structural nature



Reference nets



Characteristics

- **tokens** can be nets – dynamic hierarchies of Petri nets
- **Java** inscriptions & **Renew** interpreter
- we can build executable rapid prototype models – concurrency

Universidad Zaragoza 1542 | Cardiff University Prifysgol Caerdydd | Université de Pau et des Pays de l'Adour

Outline

Universidad Zaragoza 1542

CARDIFF UNIVERSITY PRIFYSGOL CAERDYD

UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR

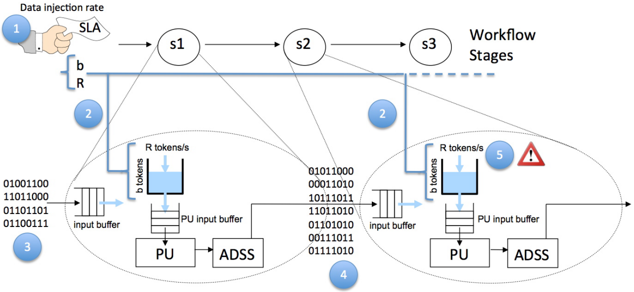
In-transit Analysis

Characteristics

- Perform **partial/full** processing of data from source to destination
- Benefit from availability of **slack** in the network – i.e. availability of excess capacity at processing nodes
- Useful to support: filtering, statistical analysis (min, max, avg) over a window size – i.e. common (often repeated) operations
- Same operation available at **multiple nodes** – location of analysis not important

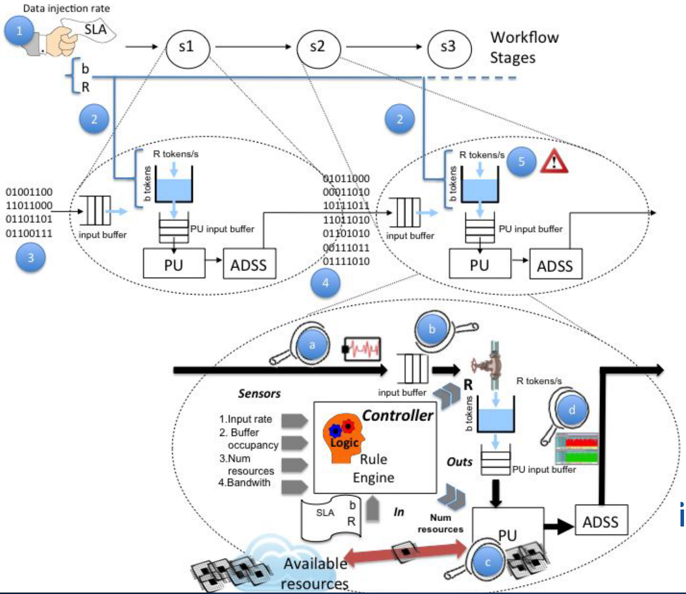
Outline

System Architecture



- **3** key components / node: Token Bucket, Processing Unit & output streaming
- Each component provides various **tunable parameters** – these can be externally modified

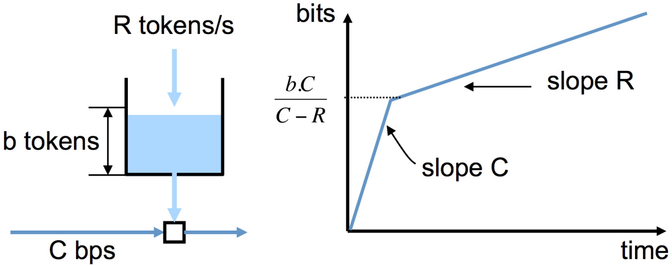
System Architecture



Some rules as an example

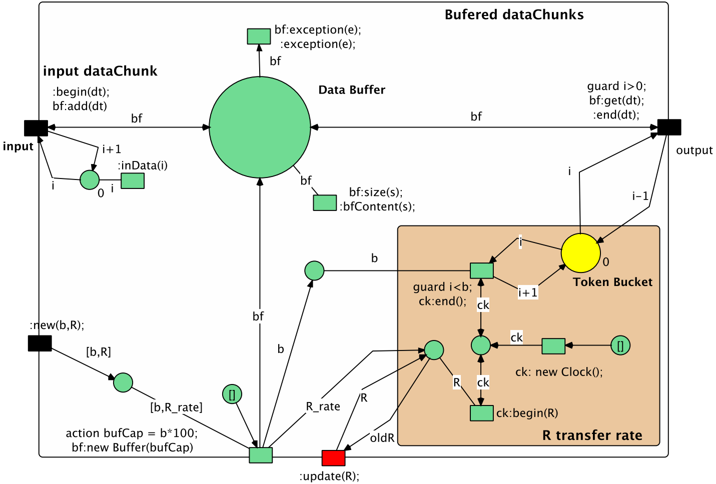
1	**E:** B_i over threshold **C:** SLA_i allows control the addition of N_i resources	$\Delta NumRes_{ij} = min(N_i, (\lambda_{ij} - R_{ij})/\hat{\delta_{ij}})$ $\Delta R_{ij} = \lambda_{ij} - R_{ij}$
2	**E:** B_i over threshold **C:** SLA_i allows control the use of free resources	$\Delta R_{ij} = \sum_{i=1}^{n} NumRes_{ij} * \hat{\delta_{ij}} - \sum_{i=1}^{n} R_{ij}$

Token Bucket Behaviour

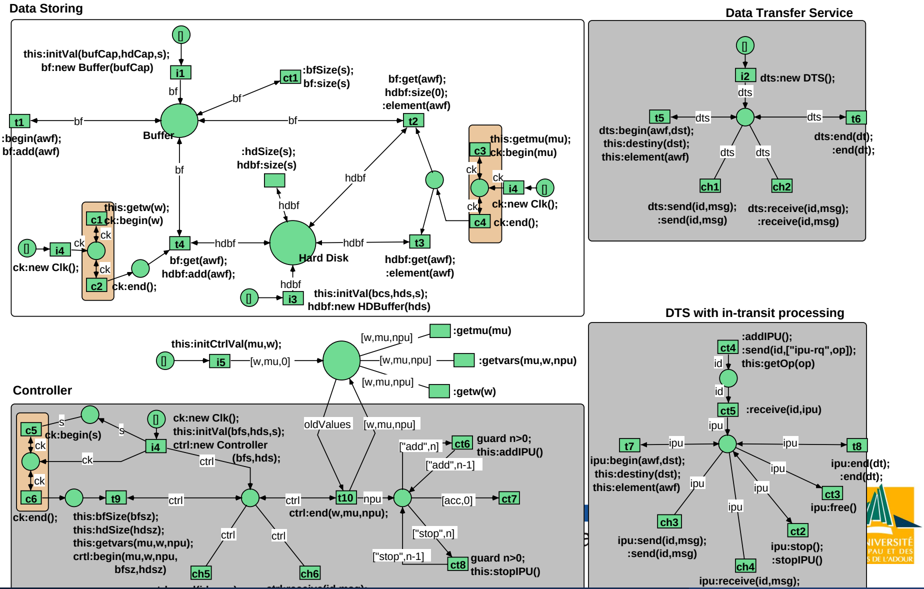


- Two key parameters of interest: R and b.
- Behaviour is dictated by changes in these two parameters.

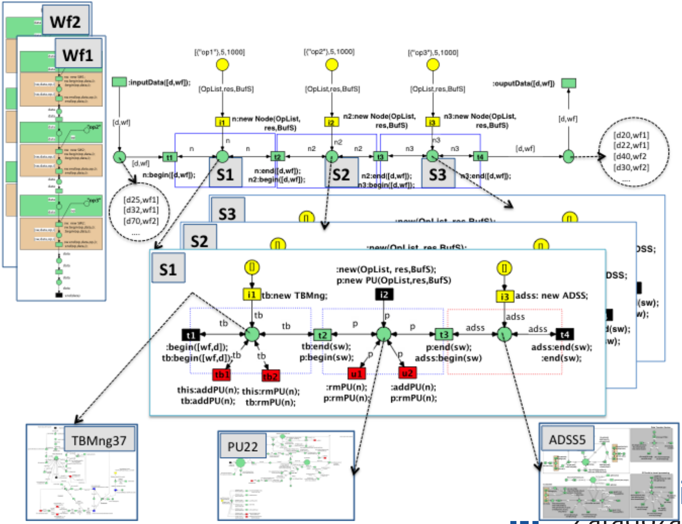
Token Bucket & Processing Units



Autonomic Data Transfer Service



Renew Model


Wf2
Wf1
[("op1"),5,1000]
[("op2"),5,1000]
[("op3"),5,1000]
:inputData([d,wf]);
:ouputData([d,wf])
[OpList,res,BufS]
n:new Node(OpList, res,BufS)
n2:new Node(OpList, res,BufS)
n3:new Node(OpList, res,BufS)
[d,wf]
n:begin([d,wf]);
n:end([d,wf]); n2:begin([d,wf]);
n2:end([d,wf]); n3:begin([d,wf]);
n3:end([d,wf]);
S1
S2
S3
[d20,wf1] [d22,wf1] [d40,wf2] [d30,wf2]
[d25,wf1] [d32,wf1] [d70,wf2]
:new(OpList, res,BufS); p:new PU(OpList,res,BufS)
tb:new TBMng;
adss: new ADSS;
:begin([wf,d]); tb:begin([wf,d]);
tb:end(sw); p:begin(sw);
p:end(sw); adss:begin(sw)
adss:end(sw); :end(sw);
this:addPU(n); tb:addPU(n);
this:rmPU(n); tb:rmPU(n);
:rmPU(n); p:rmPU(n);
:addPU(n); p:rmPU(n);
TBMng37
PU22
ADSS5

Outline

Experiments

Outline

Conclusions and Future Work

Conclusions

- in-transit processing of **multiple** data streams over a **shared** (elastic) infrastructure
- dynamic Token Bucket (admission control): support of **variable bursts**
- **elastic Processing Unit**: add / reduce computational resources
- Autonomic Data Transfer Service: **adaptive** transfers
- Controller: monitors & modifies behaviour
  - Rule based but can use other strategies (i.e. reinforcement learning)
  - QoS(node 1) + ... + QoS(node n) $\neq$ end-to-end QoS i.e. a we consider a hierarchical control strategy (but on-going!)

Conclusions and Future Work

Conclusions

- in-transit processing of **multiple** data streams over a **shared** (elastic) infrastructure
- dynamic Token Bucket (admission control): support of **variable bursts**
- **elastic Processing Unit**: add / reduce computational resources
- Autonomic Data Transfer Service: **adaptive** transfers
- Controller: monitors & modifies behaviour
  - Rule based but can use other strategies (i.e. reinforcement learning)
  - QoS(node 1) + ... + QoS(node n) $\neq$ end-to-end QoS i.e. a we consider a hierarchical control strategy (but on-going!)

Future Work

- Currently validating in an Electrical Vehicles scenario
- Focus on pipeline MoC – applicable to other workflow engines where data transfer can be modified

End-to-End QoS on Shared Clouds for Highly Dynamic, Large-Scale Sensing Data Streams

Rafael Tolosana-Calasanz, José A. Bañares, Congduc Pham and **Omer F. Rana**

{rafaelt, banares}@unizar.es, congduc.pham@univ-pau.fr, o.f.rana@cs.cardiff.ac.uk

Universidad de Zaragoza
Zaragoza, Spain

Université de Pau
Pau, France

Cardiff University
Cardiff, United Kingdom

May 13^{rd}, 2012

DPMSS 2012 – CCGrid 2012

Ottawa, Canada