A lightweight MapReduce framework for secure processing with SGX

Rafael Pires*, Daniel Gavril†, Pascal Felber*, Emanuel Onica† and Marcelo Pasin*

* University of Neuchâtel, Switzerland
† Alexandru Ioan Cuza University of Iaşi, Romania

rafael.pires@unine.ch

International Workshop on Assured Cloud Computing and QoS aware Big Data
WACC – May 14th, 2017 – Madrid, Spain
Outline

● Introduction

● Background:
  – Intel SGX (privacy assurance)
  – MapReduce (processing)
  – SCBR (communication)

● Architecture and implementation

● Experiments and results

● Concluding remarks
Introduction

- MapReduce is widely used for data processing in public cloud infrastructures

- Such environments do not offer strong security guarantees
  - Multi-tenant
  - Large software stack
  - Storage and network encryption is not enough
  - Malicious personnel managing system software may inspect or tamper with clear data during processing

- Trusted execution environments in commodity hardware may help
Software guard extensions

- Intel SGX
  - Commodity hardware
  - 6th generation (Skylake, Q4 2015)
  - Enclaves

- Protects applications from system software and other applications

- Security perimeter is the CPU Package boundary

- Data and code is only unencrypted inside CPU

- Memory confidentiality, integrity and freshness are assured (performance cost)

- Provides sealing and attestation mechanisms
Software guard extensions

- It does not prevent
  - **Side channel attacks**: e.g., traffic, power or timing analysis
  - **Denial of service attacks**
    System software still allocates memory pages and deals with I/O
MapReduce

- Simple model
- Leverages parallel task processing
Secure content-based routing

- Clients register their interests (subscriptions)
- Publishers provide content (publications)
- Filter engine deployed in the cloud routes publications to matching Clients
- Data is encrypted outside enclaves
- Privacy is assured by performing the filtering inside enclaves
Lightweight MapReduce

- Lua interpreter inside enclaves (lightweight, easily embeddable)
- Encrypted code and data provisioned by Client

<table>
<thead>
<tr>
<th></th>
<th>size (bytes) (text+data+bss)</th>
</tr>
</thead>
<tbody>
<tr>
<td>client</td>
<td>407,016</td>
</tr>
<tr>
<td>worker</td>
<td>315,637</td>
</tr>
<tr>
<td>worker enclave</td>
<td>822,763</td>
</tr>
<tr>
<td>scbr</td>
<td>292,348</td>
</tr>
<tr>
<td>scbr enclave</td>
<td>367,031</td>
</tr>
</tbody>
</table>
local json = require "json"

function hash(key, rcount)
    return string.byte(key, 1) % rcount
end

function combine(key, value)
    local clist = json:decode(value)
    local sum = 0
    for k, v in pairs(clist) do
        sum = sum + v
    end
    push(key, sum)
end

function map(key, value)
    for word in value:gmatch("\%w+\") do
        push(word, 1)
    end
end

function reduce(key, value)
    local clist = json:decode(value)
    local sum = 0
    for k, v in pairs(clist) do
        sum = sum + v
    end
    push(key, sum)
end
Bootstrap
Provisioning of code and data

[Diagram: Flowchart of code and data provisioning between a Client and SCBR.]
**Experiment – K-means**

- **Data split - 2D data points**
  - map() 
  - map() 
  - reduce()

- Compute the distance to each data point from each cluster center and assign points to the cluster centers

- Compute the new cluster centers

- Compute the error and decide whether to continue iteration

---

International Workshop on Assured Cloud Computing and QoS aware Big Data  
WACC – May 14th, 2017 – Madrid, Spain
K-means profiling

![Graph 1: Avg centroid distances vs iterations](chart1.png)

![Graph 2: Time vs Number of clusters](chart2.png)

**Graph 1:**
- Blue line: Initial centroids closer to resulting ones
- Orange line: Initial centroids far from resulting ones

**Graph 2:**
- Different line styles represent different data sizes:
  - \( n = 1k \)
  - \( n = 10k \)
  - \( n = 100k \)
  - \( n = 1M \)
Results

<table>
<thead>
<tr>
<th>n</th>
<th>Split</th>
<th>Shuffle</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1k</td>
<td>58.7 kB</td>
<td>112.1 kB</td>
<td>4.3 kB</td>
</tr>
<tr>
<td>10k</td>
<td>257.2 kB</td>
<td>1.1 MB</td>
<td>4.5 kB</td>
</tr>
<tr>
<td>100k</td>
<td>2.2 MB</td>
<td>11 MB</td>
<td>4.6 kB</td>
</tr>
<tr>
<td>1M</td>
<td>19.1 MB</td>
<td>96.5 MB</td>
<td>4.6 kB</td>
</tr>
</tbody>
</table>
Conclusions

- Viable solution for limited amount of sensitive data
- Privacy assurance with no required knowledge from the user with respect to:
  - Attestation mechanisms
  - Cryptographic and integrity primitives
  - SGX
- Performance bottlenecks can be overcome (SGX2)
Thank you.
Questions?

Rafael Pires
rafael.pires@unine.ch

Daniel Gavril
Pascal Felber
Emanuel Onica
Marcelo Pasin

International Workshop on Assured Cloud Computing and QoS aware Big Data
WACC – May 14th, 2017 – Madrid, Spain