

Avaliação de disponibilidade de alternativas arquiteturais em ambientes de Mobile Cloud Computing

Danilo Oliveira

Universidade Federal de Pernambuco

dmo4@cin.ufpe.br

22 de outubro de 2013

- 1 Introdução
- 2 Metodologia
- 3 Validação dos modelos
- 4 Cenários alternativos de Mobile Cloud Computing
- 5 Conclusões

Mobile Cloud Computing - o que é?

- Definição 1 – Uma cloud tradicional na qual alguns dos dispositivos envolvidos são móveis
- Definição 2 – Utilização da tecnologia de cloud para aumentar o poder de execução de dispositivos móveis
- Definição 3 – Uma nuvem “ad hoc” onde o “data center” é formado por dispositivos móveis
- Outra tendência é a nova forma de computação em nuvem denominada “Mobile Backend as a Service” (MBaaS)

Mobile Cloud Computing - questões

- Problemas enfrentados na computação móvel – baixa capacidade de processamento, memória, armazenamento; bateria
 - Um objetivo da MCC é superar tais limitações da computação movel
- Problemas enfrentados na MCC – exigência de uma conexão wireless sempre disponível, latência, bateria ainda é um problema...

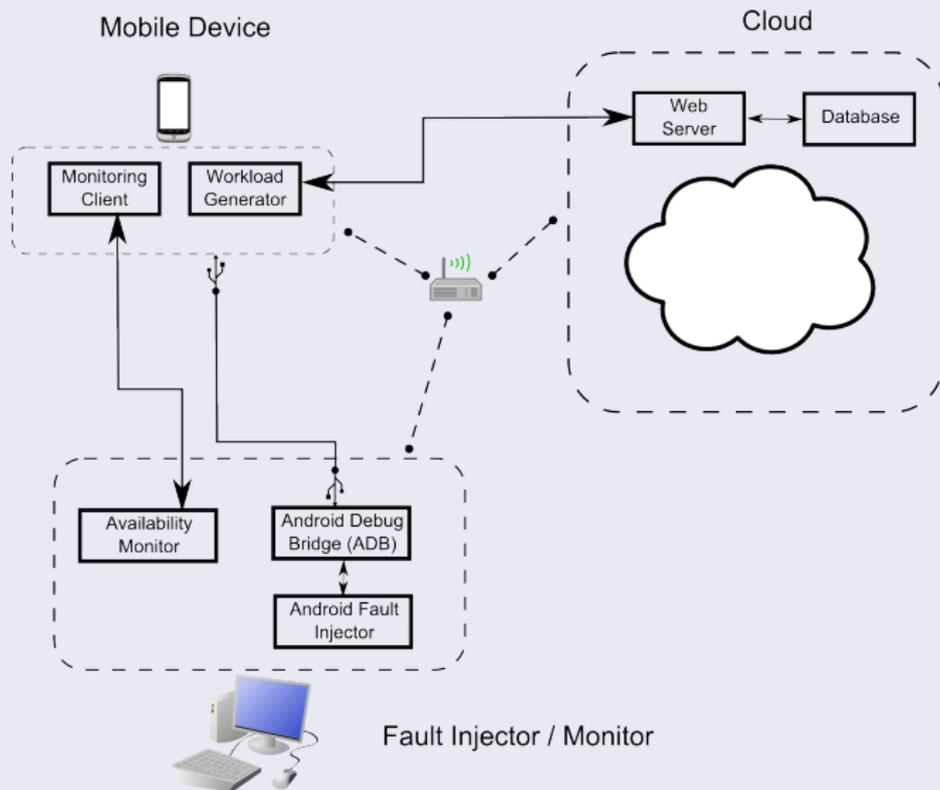
Objetivo do trabalho

- Desenvolver uma metodologia de avaliação de disponibilidade em ambientes de Mobile Cloud Computing
- Aplicar a metodologia em uma série de cenários propostos para aumentar a disponibilidade de um ambiente MCC

Metodologia - resumo

- 1 Definição de um cenário base e criação de um modelo analítico (RBD) para este cenário
- 2 Construção de scripts de simulação para avaliação da disponibilidade do cenário base
- 3 Experimentos de injeção de falhas para o cenário base
- 4 Comparação dos resultados obtidos pelas três técnicas – validação dos modelos
- 5 Definição de novos cenários e respectivos modelos, que apresentem melhorias na disponibilidade em relação ao cenário base

Arquitetura do testbed



Script de simulação para disponibilidade para o NS-2

```
set cloud [new SeriesComponent cloud $ns]
$cloud addComponent $node01
$cloud addComponent $node02

set mhw_battery [new ExponentialComponent mhw_battery $ns [expr 8.996
set mso_mapp [new ExponentialComponent mso_mapp $ns [expr 272.924747/
set mobile_device [new SeriesComponent mobile_device $ns]
$mobile_device addComponent $mhw_battery
$mobile_device addComponent $mso_mapp

set wifi [new ExponentialComponent wifi $ns [expr 5.99640216/$FACTOR]
set series01 [new SeriesComponent series01 $ns]

$series01 addComponent $cloud
$series01 addComponent $mobile_device
$series01 addComponent $wifi

$node01 startAll
$node02 startAll
$mobile_device startAll
$wifi start

$ns at 120 "finish"

$ns run
```

Parâmetros do modelo (em horas)

Component	MTTF	MTTR
Mobile Hardware (M_HW)	22461.5	1.667
Battery	9	0.083
Mobile OS (M_OS)	1440.9	0.033
Mobile App (M_APP)	336.7	0.0167
WiFi Access Point	10000	1.667
WiFi Signal	6	0.078
Hardware of Node (N_HW)	8760	1.667
OS of Node (N_OS)	2893	0.25
KVM	2990	1
VM OS (V_SO)	2893	0.25
VM App (V_APP)	788	1

Parâmetros do modelo (em horas) – componentes agrupados

Group	MTTF	MTTR
V_OS+V_APP	619.311	0.840
N_HW+N_OS+KVM	1259.025	0.770
M_HW+Battery	8.996	0.084
M_OS + M_APP	272.925	0.020
WIFI	5.996	0.079

Fórmulas de disponibilidade para o experimento

- Estimador #1:

$$\hat{A}_1 = \frac{\text{Number of "up" samples}}{\text{Total number of samples}} \quad (1)$$

- Estimador #2 para cálculo de intervalo de confiança

$$\hat{A}_2 = \frac{MTTF}{MTTF + MTTR}, \quad (2)$$

- Limites inferior e superior do intervalo de confiança:

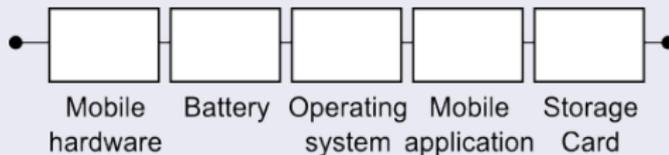
$$A_L = \frac{1}{1 + \left(\frac{1}{\hat{A}_2} - 1\right) F_{\alpha/2, 2n, 2n}}, \quad (3)$$

$$A_U = \frac{1}{1 + \left(\frac{1}{\hat{A}_2} - 1\right) F_{1-\alpha/2, 2n, 2n}}, \quad (4)$$

Validação do modelo – resultados

Technique	Availability
RBD model	0.1528886
Simulation model	90% C.I. (0.1524987, 0.1530292)
Fault injection experiment	$\hat{A}_1 = 0.156066$ 90% C.I. (0.017314, 0.418270)

1º cenário - Arquitetura *store and forward*



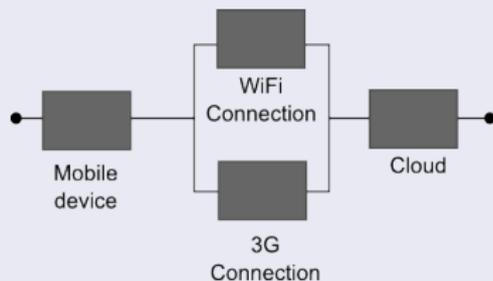
- Vantagens -

- Redução do consumo energético
- Máxima disponibilidade

- Desvantagens -

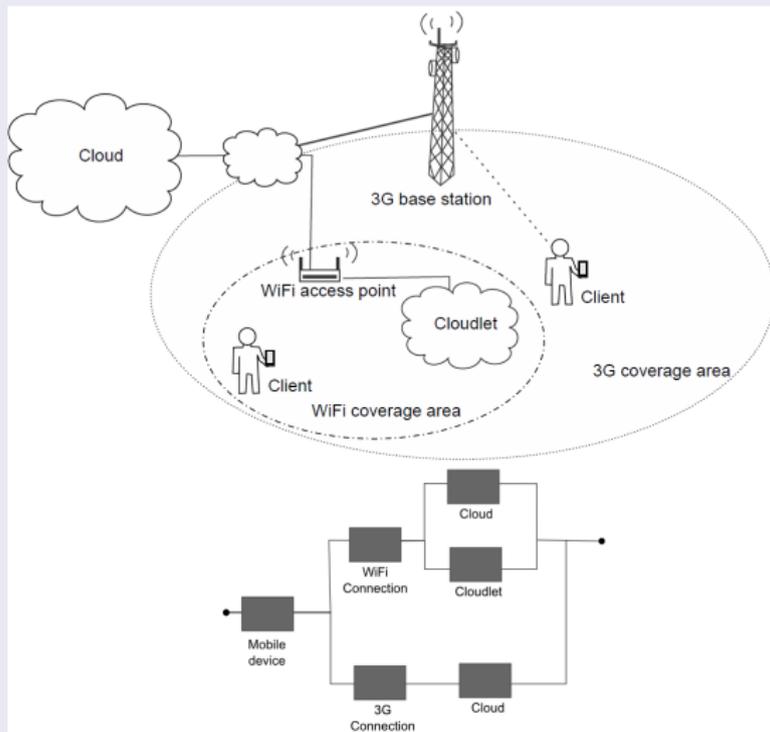
- Nem sempre essa arquitetura poderá ser utilizada

2º cenário - Múltiplas interfaces de rede



Cenários alternativos de Mobile Cloud Computing

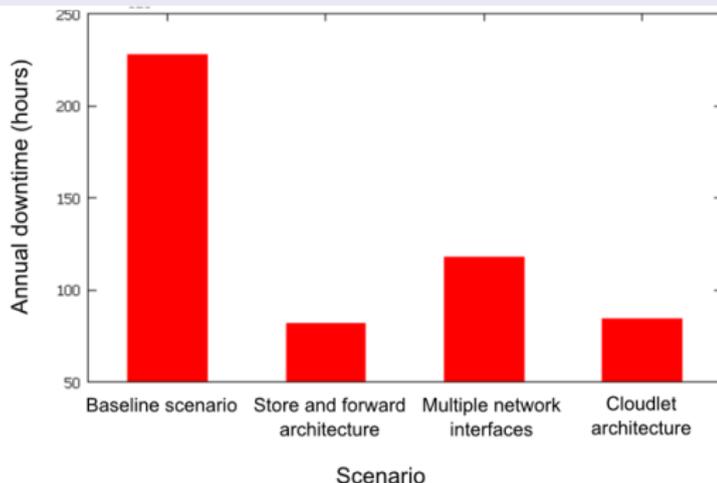
3º cenário - Cloudlet



Resultados

COMPARISON OF PROPOSED ARCHITECTURES

Scenario	Availability
Baseline scenario	0.9740028
Store and forward architecture	0.9906802
Multiple network interfaces	0.9865474
Cloudlet architecture	0.9903837



Conclusões

- Disponibilidade ainda é um fator crítico em aplicações em MCC, considerando a baixa disponibilidade do cenário base
- Arquitetura store and forward fornece o nível máximo de disponibilidade
- Arquitetura em cloudlet é capaz de se aproximar da arquitetura store and forward em termos de disponibilidade
- Em trabalhos futuros avaliaremos a disponibilidade de mecanismos de tolerância a falhas na nuvem utilizando a metodologia proposta