

Fase 2 - Implementação de Eleição

Douglas Alves
Marcelo Matsumoto
Tiago Silva

Universidade Estadual de Campinas

14 de fevereiro de 2011

Como funciona?

- Um cliente que quer ser o líder deve criar um znode efêmero e sequencial em `"/ELECTION"`.
- O cliente cujo znode possuir o menor ID é nomeado o líder, enquanto os outros clientes esperam enquanto esse não caia.
- Caso o líder caia uma nova eleição é realizada.
- Para que a queda seja conhecida, é colocado um watcher sobre o znode do líder que é disparado quando ele cai.

Líder caiu?

Election.java

```
1 synchronized public void process(WatchedEvent event){
2     synchronized (mutex) {
3         if (event.getType() != Event.EventType.None) {
4             boolean isNodeDeleted = event.getType().equals(EventType
                .NodeDeleted);           // Verifica se o evento eh
                de queda.
5             boolean LiderAtual = event.getPath().equals(lider); //
                Verifica se evento ocorreu no líder.
6
7             if (isNodeDeleted && LiderAtual) {
8                 System.out.println("O no lider caiu");
9                 mutex.notify();
10                notifyAll(); //Acorda clientes
11            }
12        }
13    }
14 }
```

Líder atual é atualizado

Election.java

```
1      /* Insere na classe o lider atual calculado  
      pela classe queue. */  
2      public void SetLider(String s) {  
3          this.lider = s;  
4      }
```

Esperando líder cair

Election.java

```
1      public void run() {  
2          try {  
3              synchronized (this) {  
4                  wait(); // Espera líder cair  
5              }  
6          } catch (InterruptedException e) {  
7          }  
8      }
```

Inserindo znode efêmero na disputa pela liderança

Election.java

```
1      String produce(int i) throws KeeperException,
        InterruptedException{
2          ByteBuffer b = ByteBuffer.allocate(4);
3          byte[] value;
4
5          // Add child with value i
6          b.putInt(i);
7          value = b.array();
8          return zk.create(root + "/" + n_ + value, Ids.
                OPEN_ACL_UNSAFE,
9              CreateMode.EPHEMERAL_SEQUENTIAL);
10     }
```

Quantos clientes estão querendo ser líder

Election.java

```
1      int tamanho() throws KeeperException,
           InterruptedException{
2          Stat stat = new Stat();
3          zk.getData("/ELECTION", false, stat);
4
5          return stat.getNumChildren();
6      }
```

Qual é o menor ID?

Election.java

```
1      String menor() throws KeeperException, InterruptedException {
2          int retvalue = -1;
3          Stat stat = null;
4          String aux = new String();
5
6          while (true) {
7              List<String> list = zk.getChildren(root, true);
8              if (list.size() != 0) {
9                  Integer min = new Integer(list.get(0).substring(7));
10                 aux = list.get(0).substring(2);
11                 for (String s : list) {
12                     Integer tempValue = new Integer(s.substring(7));
13                     if (tempValue < min) {
14                         min = tempValue;
15                         aux = s.substring(2);
16                     }
17                 }
18                 return aux;
19             }
20             return INFINITO;
21         }
22     }
```


Monitoração do znode

Election.java

```
1      void monitora(String s){  
2          this.dm = new DataMonitor(zk, "/ELECTION/n_  
           " + s, null, this);  
3      }
```

Criação de znode e espera pela queda do líder

Election.java

```
1      public static void election(String args[]){
2          Queue q = new Queue(args[0], "/ELECTION");
3          try{
4              int selfId = Integer.parseInt(q.produce(0).substring(13));
5                  //Criação do znode
6              System.out.println("Meu ID: " + selfId);
7
8              List<String> list = q.zk.getChildren("/ELECTION", true);
9              int aux = -1;
10             while (q.tamanho() != 0) {
11                 Integer menor = new Integer(q.menor());
12                 if (selfId < menor){
13                     System.out.println("Eu morri!!");
14                     return;
15                 }
16                 if (menor != aux){
17                     System.out.println("Menor filho:" + menor);
18                     aux = menor;
19                     q.SetLider("/ELECTION/n_" + q.menor());
20                     if (selfId == menor){ //Verifica se o menor id eh o meu
21                         System.out.println("Eu sou o lider\n");
22                     } else{
23                         naoLider(q);
24                     }
25                 }
26             }
27         }
28     }
```

Função não líder

Election.java

```
1
2     static public void naoLider(Queue q){
3         try{
4             System.out.println("O lider nao sou eu");
5             System.out.println("Entao eu vou
6                 monitorar o lider...\n");
7             q.monitora(q.menor());
8             q.run();
9             System.out.println("Fique esperando algo
10                 acontecer...\n");
11         } catch (KeeperException e){
12         } catch (InterruptedException e){
13         }
14     }
15     return;
16 }
```