

Paxos implementatie



Gemaakt door: Brandon betz, Charlie Choffat, Storm Joannes

Code:

Ons Paxos algoritme hebben we gemaakt door middel van OOP. We hadden de volgende classes hiervoor gemaakt:

Proposer:

De proposer class heeft de optie om een propose te sturen naar de acceptors. Het heeft een functie delivermessage die op basis van de geven input een message stuurt naar alle acceptors. Als de input bijvoorbeeld "propose" is stuurt de proposer een prepare message naar alle acceptors.

Acceptor:

De acceptor class krijgt een bericht van de proposers. Dit wordt gedaan met de functie receivemessage. Op basis van wat de proposer stuurt stuurt de acceptor een bericht terug. Dit kan een promise of een accepted/rejected zijn.

Learner

Als een propose wordt geaccepteerd door een merendeel van de acceptors wordt de propose value gestuurd naar de learner. De learner class slaat de propose value op in een matrix. Voor deze opdracht bestaat de matrix in de learner alleen uit alle mogelijke lettercombinaties (aa, ab, ac, etc).

Message

De message class is het bericht wat de computers verzenden. De message wordt aangemaakt en krijgt vier attributen mee: de computer die het bericht stuurt, de computer die het bericht moet ontvangen, het type bericht (propose, accept, etc.) en het propose nummer. Deze message wordt uiteindelijk aan de queue toegevoegd en uitgevoerd.

Network

De network class slaat alle messages op die worden gestuurd tussen de andere classes. Als er een punt komt dat er een computer stuk gaat wordt zijn laatste actie bewaard en pas ingezet wanneer de computer weer gefixt is. Dit wordt gedaan met de behulp van de functie extractmessage.

How to run

Om de code te runnen kun je main.py starten. Deze kan gerund worden via de terminal (of via de pycharm). We hebben de input gemaakt zoals het stond in de voorbeelden op canvas.

Je begint met het invullen van de hoeveelheid proposer, acceptors, learners en ticks. Als je een van deze waardes niet nodig hebt vul je hier een 0 in plaats van niks. Vervolgens wordt er gevraagd om een event te maken. Dit kan zoals op de voorbeelden van canvas. Er blijft gevraagd worden om een event te maken totdat je de input "0 END" geeft. Dan begint het algoritme.

Hieronder zijn de 3 voorbeelden die op canvas staan met onze code.

Voorbeeld 1:

```
Proposers, Acceptors, Learners, Ticks: 1 3 0 15
Event: 0 propose 1 42
Event: 0 end
0:    -> P1 propose v=42
1: P1 -> A2 prepare n=1
2: P1 -> A3 prepare n=1
3: P1 -> A1 prepare n=1
4: A2 -> P1 promise n=1 (Prior: None, None)
5: A3 -> P1 promise n=1 (Prior: None, None)
6: A1 -> P1 promise n=1 (Prior: None, None)
7: P1 -> A2 accept n=1 v=42
8: P1 -> A3 accept n=1 v=42
9: P1 -> A1 accept n=1 v=42
10: A2 -> P1 accepted n=1 v=42
11: A3 -> P1 accepted n=1 v=42
12: A1 -> P1 accepted n=1 v=42

P1 heeft wel consensus (voorgesteld: 42, geaccepteerd: 42)
```

Voorbeeld 2:

```
Proposers, Acceptors, Learners, Ticks: 2 3 0 50
Event: 0 propose 1 42
Event: 8 fail proposer 1
Event: 11 propose 2 37
Event: 26 recover proposer 1
Event: 0 end
0:    -> P1 propose v=42
1: P1 -> A2 prepare n=1
2: P1 -> A1 prepare n=1
3: P1 -> A3 prepare n=1
4: A2 -> P1 promise n=1 (Prior: None, None)
5: A1 -> P1 promise n=1 (Prior: None, None)
6: A3 -> P1 promise n=1 (Prior: None, None)
7: P1 -> A2 accept n=1 v=42
8: ** P1 kapot **
9:
10:
11:    -> P2 propose v=37
12: P2 -> A2 prepare n=2
13: P2 -> A1 prepare n=2
14: P2 -> A3 prepare n=2
15: A2 -> P2 promise n=2 (Prior: 1, 42)
16: A1 -> P2 promise n=2 (Prior: None, None)
17: A3 -> P2 promise n=2 (Prior: None, None)
18: P2 -> A2 accept n=2 v=42
19: P2 -> A1 accept n=2 v=42
20: P2 -> A3 accept n=2 v=42
21: A2 -> P2 accepted n=2 v=42
22: A1 -> P2 accepted n=2 v=42
23: A3 -> P2 accepted n=2 v=42
24:
25:
26: ** P1 gerepareerd **
27: P1 -> A1 accept n=1 v=42
28: P1 -> A3 accept n=1 v=42
29: A2 -> P1 accepted n=1 v=42
30: A1 -> P1 rejected n=1 v=42
31: A3 -> P1 rejected n=1 v=42
32: P1 -> A2 prepare n=3
33: P1 -> A1 prepare n=3
34: P1 -> A3 prepare n=3
35: A2 -> P1 promise n=3 (Prior: 2, 42)
36: A1 -> P1 promise n=3 (Prior: 2, 42)
37: A3 -> P1 promise n=3 (Prior: 2, 42)
38: P1 -> A2 accept n=3 v=42
39: P1 -> A1 accept n=3 v=42
40: P1 -> A3 accept n=3 v=42
41: A2 -> P1 accepted n=3 v=42
42: A1 -> P1 accepted n=3 v=42
43: A3 -> P1 accepted n=3 v=42

P1 heeft wel consensus (voorgesteld: 42, geaccepteerd: 42)
P2 heeft wel consensus (voorgesteld: 37, geaccepteerd: 42)
```

Voorbeeld 3:

Let op. Bij dit voorbeeld moet er ipv “f” alleen “t” staan. De code kan niet goed omgaan met spaties NA een letter.

```
Proposers, Acceptors, Learners, Ticks: 1 3 1 10000
Event: 0 PROPOSE 1 nl: g
Event: 100 PROPOSE 1 nl:ga
Event: 200 PROPOSE 1 nl:af
Event: 300 propose 1 nl:f
Event: 400 PROPOSE 1 en: g
Event: 400 PROPOSE 1 en: g
Event: 600 PROPOSE 1 en:re
Event: 700 PROPOSE 1 en:ea
Event: 800 PROPOSE 1 en:at
Event: 900 PROPOSE 1 en:t
Event: 0 end
0: -> P1 propose v=nl: g
1: P1 -> A1 prepare n=1
2: P1 -> A3 prepare n=1
3: P1 -> A2 prepare n=1
4: A1 -> P1 promise n=1 (Prior: None, None)
5: A3 -> P1 promise n=1 (Prior: None, None)
6: A2 -> P1 promise n=1 (Prior: None, None)
7: P1 -> A1 accept n=1 v=nl: g
8: P1 -> A3 accept n=1 v=nl: g
9: P1 -> A2 accept n=1 v=nl: g
10: A1 -> P1 accepted n=1 v=nl: g
11: A3 -> P1 accepted n=1 v=nl: g
12: A2 -> P1 accepted n=1 v=nl: g
13: P1 -> L1 succes n=1 v=nl: g
100: -> P1 propose v=nl:ga
101: P1 -> A1 prepare n=2
102: P1 -> A3 prepare n=2
103: P1 -> A2 prepare n=2
104: A1 -> P1 promise n=2 (Prior: None, None)
105: A3 -> P1 promise n=2 (Prior: None, None)
106: A2 -> P1 promise n=2 (Prior: None, None)
107: P1 -> A1 accept n=2 v=nl:ga
108: P1 -> A3 accept n=2 v=nl:ga
109: P1 -> A2 accept n=2 v=nl:ga
110: A1 -> P1 accepted n=2 v=nl:ga
111: A3 -> P1 accepted n=2 v=nl:ga
112: A2 -> P1 accepted n=2 v=nl:ga
113: P1 -> L1 succes n=2 v=nl:ga
200: -> P1 propose v=nl:af
201: P1 -> A1 prepare n=3
202: P1 -> A3 prepare n=3
203: P1 -> A2 prepare n=3
204: A1 -> P1 promise n=3 (Prior: None, None)
205: A3 -> P1 promise n=3 (Prior: None, None)
206: A2 -> P1 promise n=3 (Prior: None, None)
207: P1 -> A1 accept n=3 v=nl:af
208: P1 -> A3 accept n=3 v=nl:af
209: P1 -> A2 accept n=3 v=nl:af
210: A1 -> P1 accepted n=3 v=nl:af
211: A3 -> P1 accepted n=3 v=nl:af
212: A2 -> P1 accepted n=3 v=nl:af
213: P1 -> L1 succes n=3 v=nl:af
300: -> P1 propose v=nl:f
301: P1 -> A1 prepare n=4
302: P1 -> A3 prepare n=4
303: P1 -> A2 prepare n=4
304: A1 -> P1 promise n=4 (Prior: None, None)
305: A3 -> P1 promise n=4 (Prior: None, None)
306: A2 -> P1 promise n=4 (Prior: None, None)
307: P1 -> A1 accept n=4 v=nl:f
308: P1 -> A3 accept n=4 v=nl:f
309: P1 -> A2 accept n=4 v=nl:f
310: A1 -> P1 accepted n=4 v=nl:f
311: A3 -> P1 accepted n=4 v=nl:f
312: A2 -> P1 accepted n=4 v=nl:f
313: P1 -> L1 succes n=4 v=nl:f
```

```
400: -> P1 propose v=en: g
401: P1 -> A1 prepare n=5
402: P1 -> A3 prepare n=5
403: P1 -> A2 prepare n=5
404: A1 -> P1 promise n=5 (Prior: None, None)
405: A3 -> P1 promise n=5 (Prior: None, None)
406: A2 -> P1 promise n=5 (Prior: None, None)
407: P1 -> A1 accept n=5 v=en: g
408: P1 -> A3 accept n=5 v=en: g
409: P1 -> A2 accept n=5 v=en: g
410: A1 -> P1 accepted n=5 v=en: g
411: A3 -> P1 accepted n=5 v=en: g
412: A2 -> P1 accepted n=5 v=en: g
413: P1 -> L1 succes n=5 v=en: g
600: -> P1 propose v=en:re
601: P1 -> A1 prepare n=6
602: P1 -> A3 prepare n=6
603: P1 -> A2 prepare n=6
604: A1 -> P1 promise n=6 (Prior: None, None)
605: A3 -> P1 promise n=6 (Prior: None, None)
606: A2 -> P1 promise n=6 (Prior: None, None)
607: P1 -> A1 accept n=6 v=en:re
608: P1 -> A3 accept n=6 v=en:re
609: P1 -> A2 accept n=6 v=en:re
610: A1 -> P1 accepted n=6 v=en:re
611: A3 -> P1 accepted n=6 v=en:re
612: A2 -> P1 accepted n=6 v=en:re
613: P1 -> L1 succes n=6 v=en:re
700: -> P1 propose v=en:ea
701: P1 -> A1 prepare n=7
702: P1 -> A3 prepare n=7
703: P1 -> A2 prepare n=7
704: A1 -> P1 promise n=7 (Prior: None, None)
705: A3 -> P1 promise n=7 (Prior: None, None)
706: A2 -> P1 promise n=7 (Prior: None, None)
707: P1 -> A1 accept n=7 v=en:ea
708: P1 -> A3 accept n=7 v=en:ea
709: P1 -> A2 accept n=7 v=en:ea
710: A1 -> P1 accepted n=7 v=en:ea
711: A3 -> P1 accepted n=7 v=en:ea
712: A2 -> P1 accepted n=7 v=en:ea
713: P1 -> L1 succes n=7 v=en:ea
800: -> P1 propose v=en:at
801: P1 -> A1 prepare n=8
802: P1 -> A3 prepare n=8
803: P1 -> A2 prepare n=8
804: A1 -> P1 promise n=8 (Prior: None, None)
805: A3 -> P1 promise n=8 (Prior: None, None)
806: A2 -> P1 promise n=8 (Prior: None, None)
807: P1 -> A1 accept n=8 v=en:at
808: P1 -> A3 accept n=8 v=en:at
809: P1 -> A2 accept n=8 v=en:at
810: A1 -> P1 accepted n=8 v=en:at
811: A3 -> P1 accepted n=8 v=en:at
812: A2 -> P1 accepted n=8 v=en:at
813: P1 -> L1 succes n=8 v=en:at
900: -> P1 propose v=en:t
901: P1 -> A1 prepare n=9
902: P1 -> A3 prepare n=9
903: P1 -> A2 prepare n=9
904: A1 -> P1 promise n=9 (Prior: None, None)
905: A3 -> P1 promise n=9 (Prior: None, None)
906: A2 -> P1 promise n=9 (Prior: None, None)
907: P1 -> A1 accept n=9 v=en:t
908: P1 -> A3 accept n=9 v=en:t
909: P1 -> A2 accept n=9 v=en:t
910: A1 -> P1 accepted n=9 v=en:t
911: A3 -> P1 accepted n=9 v=en:t
912: A2 -> P1 accepted n=9 v=en:t
913: P1 -> L1 succes n=9 v=en:t
```