

# Seminar Report: Opty

Maria Gabriela Valdes and Victoria Beleuta

December 21, 2014

## 1 Introduction

During this lab, we implemented a transaction server using optimistic concurrency control in Erlang. Opty is a concurrency control method applied to transactional systems such as relational database management systems. It assumes that multiple transactions can complete without interfering with each other, while using data resources without acquiring locks. Before committing, each transaction checks if any other transaction has changed the entry. If it hasn't been changed the transaction can commit, if there are conflicts, the transaction rollsback and restarts.

## 2 Work done

We have completed the code files provided to correctly implement the algorithm. Our source code can be found in the src folder. To start the algorithm locally you have to call the function *start* of module *opty* with six arguments in the following order: number of clients, number of entries, number of updates, number of reads, duration of experiments and size of the subset of entries to be used by each client. It's important to emphasize that each entry subset of each client is created with different random values of the original set of entries. For example: *opty:start(3,4,1,1,5,2)*. With this command the Opty algorithm will start with three clients, and a store with four entries, one write operation, one read operation, with a duration of 5 seconds and a subset of 2 entries out of the total provided.

To start the algorithm with the server and clients in different Erlang nodes you first have to start two different Erlang environments. For example:

- In one terminal type *erl -sname optyserver*. This will represent an Erlang node name *optyserver@server*.
- In another different terminal type *erl -sname clients*. This will represent a Erlang node name *clients@server*.

First we start the server in the node *optyserver@server* with the command *opty:start\_server(10)*, where 10 is the number of entries. To start the clients you have to call, in the clients node, the function *start\_clients* of module *opty* with seven arguments: number of clients, number of entries, number of updates, number of reads, duration of experiments, the name of the server's node and the size of the subset of entries. For example: *opty:start\_clients(5, 10, 1, 1, 5, optyserver@server, 3)*.

### 3 Experiments

In the same machine:

- i) different number of concurrent clients in the system;

We ran the following commands to see how the algorithm runs:

- *opty:start(1,10,1,1,5,3)*

```
7> opty:start(1, 10, 1, 1, 5, 3).
Starting: 1 CLIENTS, 10 ENTRIES, 1 UPDATES PER TRANSACTION, 1 READS PER TRANSACTION, 3
SIZE OF SUBSET OF ENTRIES,
DURATION 5 s
Stopping...
1: Transactions TOTAL:73423, OK:73423, -> 100.0 %
stop
```

- *opty:start(5,10,1,1,5,3)*

```
8> opty:start(5, 10, 1, 1, 5, 3).
Starting: 5 CLIENTS, 10 ENTRIES, 1 UPDATES PER TRANSACTION, 1 READS PER TRANSACTION, 3
SIZE OF SUBSET OF ENTRIES,
DURATION 5 s
Stopping...
1: Transactions TOTAL:21073, OK:18649, -> 88.49712902766574 %
2: Transactions TOTAL:21031, OK:18718, -> 89.00194950311445 %
3: Transactions TOTAL:21010, OK:18687, -> 88.94336030461685 %
4: Transactions TOTAL:21030, OK:18592, -> 88.40703756538278 %
5: Transactions TOTAL:20899, OK:18463, -> 88.34393990143069 %
stop
```

- `opty:start(10,10,1,1,5,3)`

```
9> opty:start(10, 10, 1, 1, 5, 3).
Starting: 10 CLIENTS, 10 ENTRIES, 1 UPDATES PER TRANSACTION, 1 READS PER TRANSACTION,
3 SIZE OF SUBSET OF ENTRIES,
DURATION 5 s
Stopping...
1: Transactions TOTAL:11278, OK:8596, -> 76.21918779925518 %
2: Transactions TOTAL:11274, OK:8647, -> 76.69859854532552 %
3: Transactions TOTAL:11242, OK:8523, -> 75.81391211528198 %
4: Transactions TOTAL:11279, OK:8692, -> 77.06356946537814 %
5: Transactions TOTAL:11260, OK:8592, -> 76.30550621669627 %
6: Transactions TOTAL:11268, OK:8608, -> 76.39332623358183 %
7: Transactions TOTAL:11284, OK:8702, -> 77.1180432470755 %
8: Transactions TOTAL:11296, OK:8578, -> 75.93838526912181 %
9: Transactions TOTAL:11310, OK:8649, -> 76.47214854111405 %
10: Transactions TOTAL:11284, OK:8578, -> 76.01914214817441 %
stop_
```

- `opty:start(20,10,1,1,5,3)`

```
10> opty:start(20, 10, 1, 1, 5, 3).
Starting: 20 CLIENTS, 10 ENTRIES, 1 UPDATES PER TRANSACTION, 1 READS PER TRANSACTION,
3 SIZE OF SUBSET OF ENTRIES,
DURATION 5 s
Stopping...
1: Transactions TOTAL:5747, OK:3524, -> 61.31894901687837 %
2: Transactions TOTAL:5764, OK:3676, -> 63.77515614156835 %
3: Transactions TOTAL:5762, OK:3620, -> 62.825407844498436 %
4: Transactions TOTAL:5763, OK:3481, -> 60.40256810688877 %
5: Transactions TOTAL:5767, OK:3563, -> 61.78255592162303 %
6: Transactions TOTAL:5752, OK:3578, -> 62.20445062586926 %
7: Transactions TOTAL:5753, OK:3519, -> 61.168086215887364 %
8: Transactions TOTAL:5755, OK:3559, -> 61.841876629018245 %
9: Transactions TOTAL:5746, OK:3545, -> 61.695092238078665 %
10: Transactions TOTAL:5775, OK:3583, -> 62.04329004329004 %
11: Transactions TOTAL:5760, OK:3513, -> 60.989583333333336 %
12: Transactions TOTAL:5769, OK:3506, -> 60.77309759057029 %
13: Transactions TOTAL:5772, OK:3572, -> 61.88496188496188 %
14: Transactions TOTAL:5773, OK:3599, -> 62.34193660142041 %
15: Transactions TOTAL:5759, OK:3534, -> 61.36482028129884 %
16: Transactions TOTAL:5750, OK:3509, -> 61.02608695652174 %
17: Transactions TOTAL:5792, OK:3587, -> 61.93024861878453 %
18: Transactions TOTAL:5771, OK:3513, -> 60.87333217813204 %
19: Transactions TOTAL:5776, OK:3611, -> 62.517313019390585 %
20: Transactions TOTAL:5766, OK:3565, -> 61.82795698924731 %
stop_
```

- `opty:start(40,10,1,1,5,3)`

```
11> opty:start(40, 10, 1, 1, 5, 3).
Starting: 40 CLIENTS, 10 ENTRIES, 1 UPDATES PER TRANSACTION, 1 READS PER TRANSACTION,
3 SIZE OF SUBSET OF ENTRIES,
DURATION 5 s
Stopping...
1: Transactions TOTAL:3035, OK:1534, -> 50.543657331136735 %
2: Transactions TOTAL:3037, OK:1567, -> 51.59697069476457 %
3: Transactions TOTAL:3040, OK:1601, -> 52.66447368421053 %
4: Transactions TOTAL:3040, OK:1547, -> 50.88815789473684 %
5: Transactions TOTAL:3041, OK:1562, -> 51.36468267017428 %
6: Transactions TOTAL:3035, OK:1541, -> 50.774299835255356 %
7: Transactions TOTAL:3039, OK:1526, -> 50.2138861467588 %
8: Transactions TOTAL:3036, OK:1503, -> 49.50592885375494 %
9: Transactions TOTAL:3032, OK:1553, -> 51.22031662269129 %
10: Transactions TOTAL:3037, OK:1567, -> 51.59697069476457 %
11: Transactions TOTAL:3044, OK:1566, -> 51.44546649145861 %
12: Transactions TOTAL:3041, OK:1568, -> 51.5619861887537 %
13: Transactions TOTAL:3041, OK:1538, -> 50.57546859585663 %
14: Transactions TOTAL:3035, OK:1556, -> 51.26853377265239 %
15: Transactions TOTAL:3041, OK:1560, -> 51.29891483064781 %
16: Transactions TOTAL:3044, OK:1583, -> 52.00394218134034 %
17: Transactions TOTAL:3045, OK:1575, -> 51.724137931034484 %
18: Transactions TOTAL:3041, OK:1518, -> 49.91779020059191 %
19: Transactions TOTAL:3043, OK:1549, -> 50.90371344068354 %
20: Transactions TOTAL:3045, OK:1556, -> 51.10016420361248 %
21: Transactions TOTAL:3045, OK:1558, -> 51.16584564860427 %
22: Transactions TOTAL:3045, OK:1595, -> 52.38095238095238 %
23: Transactions TOTAL:3045, OK:1526, -> 50.11494252873563 %
24: Transactions TOTAL:3050, OK:1551, -> 50.85245901639344 %
25: Transactions TOTAL:3045, OK:1529, -> 50.21346469622332 %
26: Transactions TOTAL:3049, OK:1543, -> 50.606756313545425 %
27: Transactions TOTAL:3052, OK:1610, -> 52.75229357798165 %
28: Transactions TOTAL:3051, OK:1586, -> 51.982956407735166 %
29: Transactions TOTAL:3052, OK:1572, -> 51.507208387942335 %
30: Transactions TOTAL:3057, OK:1611, -> 52.69872423945044 %
31: Transactions TOTAL:3051, OK:1594, -> 52.2451655195018 %
32: Transactions TOTAL:3059, OK:1577, -> 51.5527950310559 %
33: Transactions TOTAL:3061, OK:1600, -> 52.270499836654686 %
34: Transactions TOTAL:3061, OK:1546, -> 50.50637046716759 %
35: Transactions TOTAL:3064, OK:1587, -> 51.79503916449086 %
36: Transactions TOTAL:3060, OK:1641, -> 53.627450980392155 %
37: Transactions TOTAL:3056, OK:1586, -> 51.897905759162306 %
38: Transactions TOTAL:3061, OK:1574, -> 51.42110421430905 %
39: Transactions TOTAL:3062, OK:1579, -> 51.567602873938604 %
40: Transactions TOTAL:3065, OK:1586, -> 51.74551386623165 %
stop_
```

- optry:start(80,10,1,1,5,3)

```

12> optry:start(80, 10, 1, 1, 5, 3).
Starting: 80 CLIENTS, 10 ENTRIES, 1 UPDATES PER TRANSACTION, 1 READS PER TRANSACTION,
3 SIZE OF SUBSET OF ENTRIES,
DURATION 5 s
Stopping...
1: Transactions TOTAL:1557, OK:753, -> 48.36223506743738 %
2: Transactions TOTAL:1557, OK:731, -> 46.94926140012845 %
3: Transactions TOTAL:1558, OK:717, -> 46.02053915275995 %
4: Transactions TOTAL:1556, OK:729, -> 46.850899742938594 %
5: Transactions TOTAL:1558, OK:756, -> 48.52374839537869 %
6: Transactions TOTAL:1556, OK:710, -> 45.62982005141388 %
7: Transactions TOTAL:1559, OK:716, -> 45.92687620269403 %
8: Transactions TOTAL:1558, OK:733, -> 47.04749679075738 %
9: Transactions TOTAL:1558, OK:731, -> 46.919127086007705 %
10: Transactions TOTAL:1560, OK:724, -> 46.41025641025641 %
11: Transactions TOTAL:1560, OK:734, -> 47.05128205128205 %
12: Transactions TOTAL:1563, OK:737, -> 47.152911068458096 %
13: Transactions TOTAL:1564, OK:714, -> 45.65217391304348 %
14: Transactions TOTAL:1562, OK:746, -> 47.75928297055058 %
15: Transactions TOTAL:1563, OK:735, -> 47.02495201535508 %
16: Transactions TOTAL:1564, OK:748, -> 47.82608695652174 %
17: Transactions TOTAL:1563, OK:708, -> 45.297504798464495 %
18: Transactions TOTAL:1566, OK:737, -> 47.06257982120051 %
19: Transactions TOTAL:1566, OK:705, -> 45.019157088122604 %
20: Transactions TOTAL:1567, OK:718, -> 45.82003828972559 %
21: Transactions TOTAL:1569, OK:766, -> 48.820905035054174 %
22: Transactions TOTAL:1570, OK:742, -> 47.261146496815286 %
23: Transactions TOTAL:1571, OK:773, -> 49.204328453214515 %
24: Transactions TOTAL:1571, OK:742, -> 47.23106301718651 %
25: Transactions TOTAL:1568, OK:728, -> 46.42857142857143 %
26: Transactions TOTAL:1571, OK:765, -> 48.6950986632718 %
27: Transactions TOTAL:1573, OK:740, -> 47.043865225683405 %
28: Transactions TOTAL:1571, OK:724, -> 46.08529598981541 %
29: Transactions TOTAL:1572, OK:733, -> 46.62849872773537 %
30: Transactions TOTAL:1571, OK:717, -> 45.63971992361553 %
31: Transactions TOTAL:1573, OK:686, -> 43.61093452002543 %
32: Transactions TOTAL:1574, OK:733, -> 46.56925031766201 %
33: Transactions TOTAL:1576, OK:751, -> 47.65228426395939 %
34: Transactions TOTAL:1576, OK:760, -> 48.223350253807105 %
35: Transactions TOTAL:1575, OK:734, -> 46.6031746031746 %
36: Transactions TOTAL:1576, OK:725, -> 46.00253807106599 %
37: Transactions TOTAL:1580, OK:784, -> 49.628253164556964 %
38: Transactions TOTAL:1570, OK:774, -> 49.049429657794676 %
39: Transactions TOTAL:1579, OK:769, -> 48.7017099430019 %
40: Transactions TOTAL:1577, OK:724, -> 45.90995561192137 %
41: Transactions TOTAL:1581, OK:764, -> 48.323845667299175 %
42: Transactions TOTAL:1583, OK:766, -> 48.389134554643086 %
43: Transactions TOTAL:1579, OK:698, -> 44.20519316022799 %
44: Transactions TOTAL:1577, OK:694, -> 44.00760938490805 %
45: Transactions TOTAL:1582, OK:728, -> 46.017699115944245 %
46: Transactions TOTAL:1583, OK:759, -> 47.946936197094125 %
47: Transactions TOTAL:1582, OK:740, -> 46.77623261694058 %
48: Transactions TOTAL:1583, OK:758, -> 47.88376500315856 %
49: Transactions TOTAL:1585, OK:764, -> 48.201892744479494 %
50: Transactions TOTAL:1583, OK:754, -> 47.6310802274163 %
51: Transactions TOTAL:1583, OK:735, -> 46.43082754264056 %
52: Transactions TOTAL:1586, OK:774, -> 48.80201765447667 %
53: Transactions TOTAL:1587, OK:727, -> 45.80970384373031 %
54: Transactions TOTAL:1588, OK:760, -> 47.850942065491185 %
55: Transactions TOTAL:1588, OK:721, -> 45.40302267002519 %
56: Transactions TOTAL:1588, OK:746, -> 46.977329974811084 %
57: Transactions TOTAL:1588, OK:759, -> 47.795969773299745 %
58: Transactions TOTAL:1587, OK:759, -> 47.82608695652174 %
59: Transactions TOTAL:1591, OK:775, -> 48.711502199874296 %
60: Transactions TOTAL:1590, OK:763, -> 47.9074213836478 %
61: Transactions TOTAL:1591, OK:710, -> 44.62602137020742 %
62: Transactions TOTAL:1592, OK:735, -> 46.16834170854271 %
63: Transactions TOTAL:1592, OK:757, -> 47.550251256281406 %
64: Transactions TOTAL:1592, OK:736, -> 46.231155778894475 %
65: Transactions TOTAL:1591, OK:742, -> 46.637335009428035 %
66: Transactions TOTAL:1592, OK:735, -> 46.16834170854271 %
67: Transactions TOTAL:1594, OK:738, -> 46.29861982434128 %
68: Transactions TOTAL:1597, OK:766, -> 47.96493425172198 %
69: Transactions TOTAL:1594, OK:738, -> 46.29861982434128 %
70: Transactions TOTAL:1595, OK:738, -> 46.26959247648903 %
71: Transactions TOTAL:1593, OK:726, -> 45.5743879472693 %
72: Transactions TOTAL:1597, OK:756, -> 47.338760175328744 %
73: Transactions TOTAL:1596, OK:724, -> 45.363408521303256 %
74: Transactions TOTAL:1600, OK:737, -> 46.0625 %
75: Transactions TOTAL:1601, OK:745, -> 46.53341661461587 %
76: Transactions TOTAL:1599, OK:736, -> 46.02876797998749 %
77: Transactions TOTAL:1604, OK:755, -> 47.069825436408976 %
78: Transactions TOTAL:1602, OK:725, -> 45.25593008739076 %
79: Transactions TOTAL:1603, OK:717, -> 44.728633811603245 %
80: Transactions TOTAL:1605, OK:762, -> 47.47663551401869 %
stop
_

```

- ii) different number of entries in the store;

We ran the following commands to see how the algorithm runs:

- `opty:start(5,1,1,1,1)`

```
2> opty:start(5,1,1,1,1).
Starting: 5 CLIENTS, 1 ENTRIES, 1 UPDATES PER TRANSACTION, 1 READS PER TRANSACTION, 1 SIZE OF
SUBSET OF ENTRIES,
DURATION 1 s
Stopping...
1: Transactions TOTAL:3959, OK:2245, -> 56.7062389492296 %
2: Transactions TOTAL:3933, OK:2256, -> 57.36079328756674 %
3: Transactions TOTAL:3951, OK:2318, -> 58.66869147051379 %
4: Transactions TOTAL:4038, OK:2396, -> 59.336305101535416 %
5: Transactions TOTAL:3978, OK:2288, -> 57.51633986928105 %
stop
```

- `opty:start(5,3,1,1,3)`

```
3> opty:start(5,3,1,1,3).
Starting: 5 CLIENTS, 3 ENTRIES, 1 UPDATES PER TRANSACTION, 1 READS PER TRANSACTION, 3 SIZE OF
SUBSET OF ENTRIES,
DURATION 1 s
Stopping...
1: Transactions TOTAL:3243, OK:2409, -> 74.28307123034227 %
2: Transactions TOTAL:3234, OK:2353, -> 72.75819418676562 %
3: Transactions TOTAL:3208, OK:2354, -> 73.3790523690773 %
4: Transactions TOTAL:3211, OK:2347, -> 73.09249454998442 %
5: Transactions TOTAL:3216, OK:2336, -> 72.636815920398 %
stop
```

- `opty:start(5,5,1,1,5)`

```
4> opty:start(5,5,1,1,5).
Starting: 5 CLIENTS, 5 ENTRIES, 1 UPDATES PER TRANSACTION, 1 READS PER TRANSACTION, 5 SIZE OF
SUBSET OF ENTRIES,
DURATION 1 s
Stopping...
1: Transactions TOTAL:2933, OK:2411, -> 82.20252301397886 %
2: Transactions TOTAL:2949, OK:2432, -> 82.46863343506273 %
3: Transactions TOTAL:2928, OK:2335, -> 79.74726775956285 %
4: Transactions TOTAL:2912, OK:2392, -> 82.14285714285714 %
5: Transactions TOTAL:2926, OK:2369, -> 80.96377306903622 %
stop
```

- `opty:start(5,10,1,1,1,10)`

```
5> opty:start(5,10,1,1,1,10).
Starting: 5 CLIENTS, 10 ENTRIES, 1 UPDATES PER TRANSACTION, 1 READS PER TRANSACTION, 10 SIZE OF SUBSET OF ENTRIES,
DURATION 1 s
Stopping...
1: Transactions TOTAL:1357, OK:1205, -> 88.7988209285188 %
2: Transactions TOTAL:1372, OK:1234, -> 89.94169096209913 %
3: Transactions TOTAL:1394, OK:1239, -> 88.88091822094691 %
4: Transactions TOTAL:1393, OK:1249, -> 89.66259870782484 %
5: Transactions TOTAL:1362, OK:1216, -> 89.28046989720998 %
stop
```

- `opty:start(5,100,1,1,1,100)`

```
6> opty:start(5,100,1,1,1,100).
Starting: 5 CLIENTS, 100 ENTRIES, 1 UPDATES PER TRANSACTION, 1 READS PER TRANSACTION, 100 SIZE OF SUBSET OF ENTRIES,
DURATION 1 s
Stopping...
1: Transactions TOTAL:50, OK:49, -> 98.0 %
2: Transactions TOTAL:53, OK:53, -> 100.0 %
3: Transactions TOTAL:55, OK:55, -> 100.0 %
4: Transactions TOTAL:55, OK:55, -> 100.0 %
5: Transactions TOTAL:57, OK:56, -> 98.24561403508773 %
stop
```

- `opty:start(5,1000,1,1,1,1000)`

```
7> opty:start(5,1000,1,1,1,1000).
Starting: 5 CLIENTS, 1000 ENTRIES, 1 UPDATES PER TRANSACTION, 1 READS PER TRANSACTION, 1000 SIZE OF SUBSET OF ENTRIES,
DURATION 1 s
Stopping...
1: Transactions TOTAL:5, OK:5, -> 100.0 %
2: Transactions TOTAL:7, OK:7, -> 100.0 %
3: Transactions TOTAL:6, OK:6, -> 100.0 %
4: Transactions TOTAL:7, OK:7, -> 100.0 %
5: Transactions TOTAL:7, OK:7, -> 100.0 %
stop
```

- iii) different number of write operations per transaction;

We ran the following commands to see how the algorithm runs:

- `opty:start(5,10,1,1,5,3)`

```
13> opty:start(5, 10, 1, 1, 5, 3).
Starting: 5 CLIENTS, 10 ENTRIES, 1 UPDATES PER TRANSACTION, 1 READS PER TRANSACTION, 3 SIZE OF SUBSET OF ENTRIES,
DURATION 5 s
Stopping...
1: Transactions TOTAL:20375, OK:18027, -> 88.4760736196319 %
2: Transactions TOTAL:20541, OK:18263, -> 88.90998490823232 %
3: Transactions TOTAL:20523, OK:18253, -> 88.93923890269454 %
4: Transactions TOTAL:20500, OK:18165, -> 88.60975609756098 %
5: Transactions TOTAL:20501, OK:18127, -> 88.42007706941125 %
stop
```

- `opty:start(5,10,5,1,5,3)`

```
14> opty:start(5, 10, 5, 1, 5, 3).
Starting: 5 CLIENTS, 10 ENTRIES, 5 UPDATES PER TRANSACTION, 1 READS PER TRANSACTION, 3
SIZE OF SUBSET OF ENTRIES,
DURATION 5 s
Stopping...
1: Transactions TOTAL:11151, OK:8029, -> 72.00251098556183 %
2: Transactions TOTAL:11107, OK:7988, -> 71.91860988565769 %
3: Transactions TOTAL:11070, OK:8055, -> 72.76422764227642 %
4: Transactions TOTAL:11144, OK:8087, -> 72.56819813352476 %
5: Transactions TOTAL:11122, OK:7990, -> 71.83959719474915 %
stop_
```

- `opty:start(5,10,10,1,5,3)`

```
15> opty:start(5, 10, 10, 1, 5, 3).
Starting: 5 CLIENTS, 10 ENTRIES, 10 UPDATES PER TRANSACTION, 1 READS PER TRANSACTION,
3 SIZE OF SUBSET OF ENTRIES,
DURATION 5 s
Stopping...
1: Transactions TOTAL:8559, OK:5720, -> 66.83023717724033 %
2: Transactions TOTAL:8493, OK:5617, -> 66.13681855645827 %
3: Transactions TOTAL:8557, OK:5717, -> 66.81079817693117 %
4: Transactions TOTAL:8527, OK:5748, -> 67.40940541808374 %
5: Transactions TOTAL:8534, OK:5716, -> 66.97914225451137 %
stop_
```

- `opty:start(5,10,20,1,5,3)`

```
16> opty:start(5, 10, 20, 1, 5, 3).
Starting: 5 CLIENTS, 10 ENTRIES, 20 UPDATES PER TRANSACTION, 1 READS PER TRANSACTION,
3 SIZE OF SUBSET OF ENTRIES,
DURATION 5 s
Stopping...
1: Transactions TOTAL:5409, OK:3449, -> 63.76409687557774 %
2: Transactions TOTAL:5448, OK:3454, -> 63.39941262848752 %
3: Transactions TOTAL:5441, OK:3459, -> 63.57287263370704 %
4: Transactions TOTAL:5510, OK:3459, -> 62.776769509981854 %
5: Transactions TOTAL:5481, OK:3490, -> 63.67451195037402 %
stop_
```

- `opty:start(5,10,40,1,5,3)`

```
17> opty:start(5, 10, 40, 1, 5, 3).
Starting: 5 CLIENTS, 10 ENTRIES, 40 UPDATES PER TRANSACTION, 1 READS PER TRANSACTION,
3 SIZE OF SUBSET OF ENTRIES,
DURATION 5 s
Stopping...
1: Transactions TOTAL:3225, OK:1804, -> 55.93798449612403 %
2: Transactions TOTAL:3213, OK:1764, -> 54.90196078431372 %
3: Transactions TOTAL:3230, OK:1825, -> 56.5015479876161 %
4: Transactions TOTAL:3244, OK:1873, -> 57.737361282367445 %
5: Transactions TOTAL:3228, OK:1860, -> 57.62081784386617 %
stop_
```



- `opty:start(5,10,80,1,5,3)`

```
18> opty:start(5, 10, 80, 1, 5, 3).
Starting: 5 CLIENTS, 10 ENTRIES, 80 UPDATES PER TRANSACTION, 1 READS PER TRANSACTION,
3 SIZE OF SUBSET OF ENTRIES,
DURATION 5 s
Stopping...
1: Transactions TOTAL:1848, OK:1040, -> 56.277056277056275 %
2: Transactions TOTAL:1855, OK:1024, -> 55.20215633423181 %
3: Transactions TOTAL:1846, OK:1041, -> 56.39219934994583 %
4: Transactions TOTAL:1846, OK:983, -> 53.25027085590466 %
5: Transactions TOTAL:1843, OK:970, -> 52.63157894736842 %
stop_
```

- iv) different ratio of read and write operations per transaction;

We ran the following commands to see how the algorithm runs:

- `opty:start(5,10,1,100,1,5)`

```
14> opty:start(5,10,1,100,1,5).
Starting: 5 CLIENTS, 10 ENTRIES, 1 UPDATES PER TRANSACTION, 100 READS PER TRANSACTION, 5 SIZE
OF SUBSET OF ENTRIES,
DURATION 1 s
Stopping...
1: Transactions TOTAL:89, OK:40, -> 44.943820224719104 %
2: Transactions TOTAL:86, OK:44, -> 51.16279069767442 %
3: Transactions TOTAL:88, OK:49, -> 55.68181818181818 %
4: Transactions TOTAL:92, OK:40, -> 43.47826086956522 %
5: Transactions TOTAL:88, OK:36, -> 40.90909090909091 %
stop_
```

- `opty:start(5,10,1,10,1,5)`

```
12> opty:start(5,10,1,10,1,5).
Starting: 5 CLIENTS, 10 ENTRIES, 1 UPDATES PER TRANSACTION, 10 READS PER TRANSACTION, 5 SIZE O
F SUBSET OF ENTRIES,
DURATION 1 s
Stopping...
1: Transactions TOTAL:683, OK:453, -> 66.32503660322108 %
2: Transactions TOTAL:664, OK:446, -> 67.16867469879519 %
3: Transactions TOTAL:685, OK:487, -> 71.0948905109489 %
4: Transactions TOTAL:687, OK:468, -> 68.12227074235808 %
5: Transactions TOTAL:671, OK:454, -> 67.6602086438152 %
stop_
```

- `opty:start(5,10,1,3,1,5)`

```
3> opty:start(5,10,1,3,1,5).
Starting: 5 CLIENTS, 10 ENTRIES, 1 UPDATES PER TRANSACTION, 3 READS PER TRANSACTION,
5 SIZE OF SUBSET OF ENTRIES,
DURATION 1 s
Stopping...
1: Transactions TOTAL:1381, OK:1072, -> 77.6249094858798 %
2: Transactions TOTAL:1347, OK:1031, -> 76.54046028210838 %
3: Transactions TOTAL:1355, OK:1039, -> 76.6789667896679 %
4: Transactions TOTAL:1377, OK:1064, -> 77.26942628903413 %
5: Transactions TOTAL:1338, OK:1011, -> 75.56053811659193 %
stop_
```

- `opty:start(5,10,1,2,1,5)`

```
2> opty:start(5,10,1,2,1,5).
Starting: 5 CLIENTS, 10 ENTRIES, 1 UPDATES PER TRANSACTION, 2 READS PER TRANSACTION,
5 SIZE OF SUBSET OF ENTRIES,
DURATION 1 s
Stopping...
1: Transactions TOTAL:1832, OK:1503, -> 82.0414847161572 %
2: Transactions TOTAL:1814, OK:1466, -> 80.81587651598677 %
3: Transactions TOTAL:1808, OK:1478, -> 81.74778761061947 %
4: Transactions TOTAL:1887, OK:1544, -> 81.82299947005829 %
5: Transactions TOTAL:1888, OK:1525, -> 80.77330508474576 %
stop
```

- `opty:start(5,10,2,1,1,5)`

```
5> opty:start(5,10,2,1,1,5).
Starting: 5 CLIENTS, 10 ENTRIES, 2 UPDATES PER TRANSACTION, 1 READS PER TRANSACTION,
5 SIZE OF SUBSET OF ENTRIES,
DURATION 1 s
Stopping...
1: Transactions TOTAL:1751, OK:1417, -> 80.92518560822387 %
2: Transactions TOTAL:1801, OK:1481, -> 82.23209328151027 %
3: Transactions TOTAL:1761, OK:1423, -> 80.80636002271437 %
4: Transactions TOTAL:1761, OK:1453, -> 82.5099375354912 %
5: Transactions TOTAL:1807, OK:1505, -> 83.28721638074155 %
stop
```

- `opty:start(5,10,3,1,1,5)`

```
4> opty:start(5,10,3,1,1,5).
Starting: 5 CLIENTS, 10 ENTRIES, 3 UPDATES PER TRANSACTION, 1 READS PER TRANSACTION,
5 SIZE OF SUBSET OF ENTRIES,
DURATION 1 s
Stopping...
1: Transactions TOTAL:1408, OK:1073, -> 76.20738636363636 %
2: Transactions TOTAL:1360, OK:1017, -> 74.77941176470588 %
3: Transactions TOTAL:1439, OK:1085, -> 75.3995830437804 %
4: Transactions TOTAL:1372, OK:1047, -> 76.31195335276968 %
5: Transactions TOTAL:1472, OK:1153, -> 78.3288043478261 %
stop
```

- `opty:start(5,10,10,1,1,5)`

```
13> opty:start(5,10,10,1,1,5).
Starting: 5 CLIENTS, 10 ENTRIES, 10 UPDATES PER TRANSACTION, 1 READS PER TRANSACTION, 5 SIZE O
F SUBSET OF ENTRIES,
DURATION 1 s
Stopping...
1: Transactions TOTAL:1019, OK:650, -> 63.78802747791953 %
2: Transactions TOTAL:1049, OK:682, -> 65.0142993326978 %
3: Transactions TOTAL:1035, OK:693, -> 66.95652173913044 %
4: Transactions TOTAL:1007, OK:643, -> 63.853028798411124 %
5: Transactions TOTAL:1012, OK:678, -> 66.99604743083005 %
stop
```

- `opty:start(5,10,100,1,1,5)`

```
15> opty:start(5,10,100,1,1,5).
Starting: 5 CLIENTS, 10 ENTRIES, 100 UPDATES PER TRANSACTION, 1 READS PER TRANSACTION, 5 SIZE
OF SUBSET OF ENTRIES,
DURATION 1 s
Stopping...
1: Transactions TOTAL:118, OK:50, -> 42.3728813559322 %
2: Transactions TOTAL:121, OK:60, -> 49.586776859504134 %
3: Transactions TOTAL:125, OK:60, -> 48.0 %
4: Transactions TOTAL:124, OK:58, -> 46.774193548387096 %
5: Transactions TOTAL:123, OK:52, -> 42.27642276422764 %
stop
```

v) different percentage of accessed entries with respect to the total number of entries;

We ran the following commands to see how the algorithm runs:

- `opty:start(5,10,2,2,1,2)`

```
16> opty:start(5,10,2,2,1,2).
Starting: 5 CLIENTS, 10 ENTRIES, 2 UPDATES PER TRANSACTION, 2 READS PER TRANSACTION, 2 SIZE OF
SUBSET OF ENTRIES,
DURATION 1 s
Stopping...
1: Transactions TOTAL:1747, OK:1264, -> 72.3526044647968 %
2: Transactions TOTAL:1824, OK:1366, -> 74.89035087719299 %
3: Transactions TOTAL:1788, OK:1342, -> 75.0559284116331 %
4: Transactions TOTAL:1791, OK:1337, -> 74.65103294249023 %
5: Transactions TOTAL:1852, OK:1387, -> 74.89200863930886 %
stop
```

- `opty:start(5,10,2,2,1,3)`

```
17> opty:start(5,10,2,2,1,3).
Starting: 5 CLIENTS, 10 ENTRIES, 2 UPDATES PER TRANSACTION, 2 READS PER TRANSACTION, 3 SIZE OF
SUBSET OF ENTRIES,
DURATION 1 s
Stopping...
1: Transactions TOTAL:1739, OK:1241, -> 71.36285221391604 %
2: Transactions TOTAL:1758, OK:1253, -> 71.27417519908987 %
3: Transactions TOTAL:1738, OK:1223, -> 70.36823935558112 %
4: Transactions TOTAL:1743, OK:1228, -> 70.45324153757889 %
5: Transactions TOTAL:1812, OK:1285, -> 70.91611479028697 %
stop
```

- `opty:start(5,10,2,2,1,4)`

```
18> opty:start(5,10,2,2,1,4).
Starting: 5 CLIENTS, 10 ENTRIES, 2 UPDATES PER TRANSACTION, 2 READS PER TRANSACTION, 4 SIZE OF
SUBSET OF ENTRIES,
DURATION 1 s
Stopping...
1: Transactions TOTAL:1666, OK:1159, -> 69.56782713085234 %
2: Transactions TOTAL:1703, OK:1193, -> 70.05284791544334 %
3: Transactions TOTAL:1699, OK:1207, -> 71.04178928781636 %
4: Transactions TOTAL:1695, OK:1154, -> 68.08259587020649 %
5: Transactions TOTAL:1710, OK:1231, -> 71.98830409356725 %
stop
```

- `opty:start(5,10,2,2,1,5)`

```
19> opty:start(5,10,2,2,1,5).
Starting: 5 CLIENTS, 10 ENTRIES, 2 UPDATES PER TRANSACTION, 2 READS PER TRANSACTION, 5 SIZE OF
SUBSET OF ENTRIES,
DURATION 1 s
Stopping...
1: Transactions TOTAL:1671, OK:1160, -> 69.4195092758827 %
2: Transactions TOTAL:1667, OK:1158, -> 69.46610677864427 %
3: Transactions TOTAL:1686, OK:1178, -> 69.86951364175563 %
4: Transactions TOTAL:1647, OK:1123, -> 68.1845780206436 %
5: Transactions TOTAL:1691, OK:1162, -> 68.71673565937316 %
stop
```

- `opty:start(5,10,2,2,1,8)`

```
20> opty:start(5,10,2,2,1,8).
Starting: 5 CLIENTS, 10 ENTRIES, 2 UPDATES PER TRANSACTION, 2 READS PER TRANSACTION, 8 SIZE OF
SUBSET OF ENTRIES,
DURATION 1 s
Stopping...
1: Transactions TOTAL:1484, OK:999, -> 67.31805929919138 %
2: Transactions TOTAL:1494, OK:1011, -> 67.6706827309237 %
3: Transactions TOTAL:1507, OK:1079, -> 71.59920371599203 %
4: Transactions TOTAL:1483, OK:1015, -> 68.44234659474039 %
5: Transactions TOTAL:1515, OK:1059, -> 69.9009900990099 %
stop
```

## In different machines:

We created the server node: `optyserver@localhost` and ran the command `opty:start_server(10)`. Then we created the clients node: `clients@localhost` and ran the command `opty:start_clients(5, 10, 1, 1, 5, optyserver@server, 3)`.

```
Victorias-MacBook-Pro:cpds_opty vbeleut@ erl -sname optyserver@localhost
Erlang/OTP 17 [erts-6.2] [source-aa0efb3] [64-bit] [smp:4:4] [async-threads:10] [hipe] [kernel-poll:false]

Eshell V6.2 (abort with ^G)
(optyserver@localhost)> cover:compile_directory().
[[ok,client],
 {ok,entry},
 {ok,handler},
 {ok,opty},
 {ok,opty_sep},
 {ok,server},
 {ok,store},
 {ok,validator}]
(optyserver@localhost)> opty:start_server(10).
true
(optyserver@localhost)> ]

Victorias-MacBook-Pro:cpds_opty vbeleut@ erl -sname clients@localhost
Erlang/OTP 17 [erts-6.2] [source-aa0efb3] [64-bit] [smp:4:4] [async-threads:10] [hipe] [kernel-poll:false]

Eshell V6.2 (abort with ^G)
(clients@localhost)> cover:compile_directory().
[[ok,client],
 {ok,entry},
 {ok,handler},
 {ok,opty},
 {ok,opty_sep},
 {ok,server},
 {ok,store},
 {ok,validator}]
(clients@localhost)> opty:start_clients(5,10,1,1,5,optyserver@localhost,3).
Starting: 5 CLIENTS, 10 ENTRIES, 1 UPDATES PER TRANSACTION, 1 READS PER TRANSACTION,
3 SIZE OF SUBSET OF ENTRIES,
DURATION 5 s
Stopping...
1: Transactions TOTAL:4283, OK:3837, -> 89.58673826756946 %
2: Transactions TOTAL:4289, OK:3816, -> 88.97178829564001 %
3: Transactions TOTAL:4295, OK:3819, -> 88.91734575087311 %
4: Transactions TOTAL:4284, OK:3892, -> 90.84967320261438 %
5: Transactions TOTAL:4297, OK:3878, -> 90.24901093786363 %
stop
(clients@localhost)> ]
```

## 4 Open questions

1) What is the impact of each of these parameters on the success rate? Is the success rate the same for the different clients?

The success rate is influenced by all of these parameters:

- For the experiment *i* we see that as we increase the number of clients, the success rate decreases. This happens because as the number of clients increases, the likelihood that this clients have very similar or even equal subset of entries increases as well, and this of course, increases the number of possible conflicts to arise, increasing as well the number of aborted transactions, and as a consequence the number of non conflicting transactions decreases.
- For the experiment *ii* however, we see that as we increase the number of entries, the success rate also increases. This happens because with a small number of entries to work with, the likelihood of having conflicts increases because the clients share a very similar subset of entries, and as a consequence this arises more conflicts between their transactions and more aborted transactions. With a bigger entry set

to share between several clients, it is more likely that each client will have different subsets of entries to work on and less number of conflicting or aborting transactions.

- For experiment *iii* as expected the more write operations we try to perform the more conflicts arise forcing transactions to restart and their success rate decreases.
- For experiment *iv* we notice that if as we increase the nr of reads to the nr of writes, the success rate decreases (for 1 write we perform 2 reads with a success rate of 81%, while if we perform for each 1 write, 100 reads we have a success of 45% ) and if we increase the nr of writes to the nr of reads the success rate decreases as well (for 1 read we perform 2 writes with a success rate of 81%, while if we perform for each 1 read, 100 writes we have a success rate of 45%). We observe that the bigger the difference between the nr of reads and writes the more the success rate will decrease.
- For experiment *v* we can see that as the size of the client's entry subset increases, the number of conflicting transactions increases as well. This happens because with a bigger subset of entries, the likelihood of this subsets to have the same entries as other client's subset increases as well. As the size of the subset increases, each client will "share" and perform read and write operations in the "same" entry set increasing the probability of arising conflicts between their transactions.

2) If we run this in a distributed Erlang network, where is the handler running?

In a distributed network the handler runs in the clients node, because each node is responsible for creating it's own handler. Because of this, when the clients are destroyed so is the handler.

## 5 Personal opinion

During this seminar we got an in depth view of how the Optimistic Concurrency Control algorithm works with backward validation, both locally and in a distributed environment. We are aware that this same algorithm could work with forward validation but it is interesting that thanks to the way in which the code was structured, such modification could be done only by modifying the *validation* module. In our opinion, this lab had the right level of difficulty for this course and the amount of time available to us.