


```
39     bashCommand = form_command(a,b,c,n, optimised = False)
40     output = subprocess.check_output(['bash','-c', bashCommand]).decode
41     ('utf8')
42     ans_unopt.append(float(output.split("\n")[35].split()[3]))
43     total_fetches = output.split("\n")[31].split()[2]
44     total_misses = output.split("\n")[34].split()[2]
45     miss_rate = str(float(total_misses)/float(total_fetches))
46     miss_compolsory = output.split("\n")[36].split()[2]
47     miss_capacity = output.split("\n")[37].split()[2]
48     miss_conflict = output.split("\n")[38].split()[2]
49     f.write('{} , {} , {} , {} , {} , {} , {} , {} , {} , {} , {} \n'.format('No',a,b,c,n,
50     total_fetches,total_misses,miss_rate,miss_compolsory,miss_conflict ,
51     miss_capacity))
52     f.flush()
53     f.close()
54     ax[i//2,i%2].plot(sampler, ans_opt, linestyle='dashed', linewidth =
55     2, marker='o', markersize=4,label='optimised')
56     ax[i//2,i%2].plot(sampler, ans_unopt, linestyle='dashed', linewidth =
57     2, marker='o', markersize=4,label='unoptimised')
58     ax[i//2,i%2].set_xlabel('Associativity')
59     ax[i//2,i%2].set_ylabel('Demand Miss Rate')
60     ax[i//2,i%2].set_title('b = {}, c = {}, n = {}'.format(b,c,n))
61     ax[i//2,i%2].legend(loc = 'upper right')
62 plt.savefig('../plots/varya.png')
63 print('\n\n\n')
64
65 # variation in block size
66 print('
=====
')
67 print('Variation in Block Size')
68 print('
=====
')
69 sampler = [1, 2, 4, 8, 16, 32, 64, 128, 256]
70 fig, ax = plt.subplots(2,2,figsize = (15,15))
71 fig.suptitle('Variation in b')
72 for i, (a,c,n) in enumerate([(2,8,100),(4,8,100),(2,16,100),(4,16,100)
73 ]):
74     ans_opt = []
75     ans_unopt = []
76     f = open('../csvs/variationb/{}_{}_{}.csv'.format(a,c,n), 'w')
77     f.write('Opt,a,b,c,n,Total Fetches,Total Misses,Miss Rate,Compulsory
78     miss, Conflict miss, Capacity miss\n')
79     f.flush()
80     for b in sampler:
81         print("a = ",a," b = ",b," c = ",c," n = ",n)
82         bashCommand = form_command(a,b,c,n, optimised = True)
83         output = subprocess.check_output(['bash','-c', bashCommand]).decode
84         ('utf8')
85         ans_opt.append(float(output.split("\n")[35].split()[3]))
86         total_fetches = output.split("\n")[31].split()[2]
87         total_misses = output.split("\n")[34].split()[2]
88         miss_rate = str(float(total_misses)/float(total_fetches))
89         miss_compolsory = output.split("\n")[36].split()[2]
90         miss_capacity = output.split("\n")[37].split()[2]
```

```
83     miss_conflict = output.split("\n")[38].split()[2]
84     f.write('{} , {} , {} , {} , {} , {} , {} , {} , {} , {} , {} \n'.format('Yes',a,b,c,n,
    total_fetches,total_misses,miss_rate,miss_compolsory,miss_conflict,
    miss_capacity))
85
86     bashCommand = form_command(a,b,c,n, optimised = False)
87     output = subprocess.check_output(['bash','-c', bashCommand]).decode
    ('utf8')
88     ans_unopt.append(float(output.split("\n")[35].split()[3]))
89     total_fetches = output.split("\n")[31].split()[2]
90     total_misses = output.split("\n")[34].split()[2]
91     miss_rate = str(float(total_misses)/float(total_fetches))
92     miss_compolsory = output.split("\n")[36].split()[2]
93     miss_capacity = output.split("\n")[37].split()[2]
94     miss_conflict = output.split("\n")[38].split()[2]
95     f.write('{} , {} , {} , {} , {} , {} , {} , {} , {} , {} , {} \n'.format('No',a,b,c,n,
    total_fetches,total_misses,miss_rate,miss_compolsory,miss_conflict,
    miss_capacity))
96 f.flush()
97 f.close()
98 ax[i//2,i%2].plot(sampler, ans_opt, linestyle='dashed', linewidth =
    2, marker='o', markersize=4,label='optimised')
99 ax[i//2,i%2].plot(sampler, ans_unopt, linestyle='dashed', linewidth =
    2, marker='o', markersize=4,label='unoptimised')
100 ax[i//2,i%2].set_xlabel('Block Size')
101 ax[i//2,i%2].set_ylabel('Demand Miss Rate')
102 ax[i//2,i%2].set_title('a = {}, c = {}, n = {}'.format(a,c,n))
103 ax[i//2,i%2].legend(loc = 'upper right')
104 plt.savefig('../plots/varyb.png')
105 print('\n\n\n')
106
107 # variation in cache size
108 print('
    =====
    ')
109 print('Variation in Cache Size')
110 print('
    =====
    ')
111 sampler = [1, 2, 4, 8, 16, 32, 64, 128, 256]
112 fig, ax = plt.subplots(2,2,figsize = (15,15))
113 fig.suptitle('Variation in c')
114 for i, (a,b,n) in enumerate([(2,64,100),(4,64,100),(2,32,100)
    ,(4,32,100)]):
115     ans_opt = []
116     ans_unopt = []
117     f = open('../csvs/variationc/{}_{}_{}.csv'.format(a,b,n), 'w')
118     f.write('Opt,a,b,c,n,Total Fetches,Total Misses,Miss Rate,Compulsory
    miss, Conflict miss, Capacity miss\n')
119     f.flush()
120     for c in sampler:
121         print("a = ",a," b = ",b," c = ",c," n = ",n)
122         bashCommand = form_command(a,b,c,n, optimised = True)
123         output = subprocess.check_output(['bash','-c', bashCommand]).decode
    ('utf8')
124         ans_opt.append(float(output.split("\n")[35].split()[3]))
```

```
125     total_fetches = output.split("\n")[31].split()[2]
126     total_misses = output.split("\n")[34].split()[2]
127     miss_rate = str(float(total_misses)/float(total_fetches))
128     miss_compolsory = output.split("\n")[36].split()[2]
129     miss_capacity = output.split("\n")[37].split()[2]
130     miss_conflict = output.split("\n")[38].split()[2]
131     f.write('{}{}{}{}{}{}{}{}{}{}{}{}{}\n'.format('Yes',a,b,c,n,
total_fetches,total_misses,miss_rate,miss_compolsory,miss_conflict,
miss_capacity))

132
133     bashCommand = form_command(a,b,c,n, optimised = False)
134     output = subprocess.check_output(['bash','-c', bashCommand]).decode
('utf8')
135     ans_unopt.append(float(output.split("\n")[35].split()[3]))
136     total_fetches = output.split("\n")[31].split()[2]
137     total_misses = output.split("\n")[34].split()[2]
138     miss_rate = str(float(total_misses)/float(total_fetches))
139     miss_compolsory = output.split("\n")[36].split()[2]
140     miss_capacity = output.split("\n")[37].split()[2]
141     miss_conflict = output.split("\n")[38].split()[2]
142     f.write('{}{}{}{}{}{}{}{}{}{}{}{}{}\n'.format('No',a,b,c,n,
total_fetches,total_misses,miss_rate,miss_compolsory,miss_conflict,
miss_capacity))
143 f.flush()
144 f.close()
145 ax[i//2,i%2].plot(sampler, ans_opt, linestyle='dashed', linewidth =
2, marker='o', markersize=4,label='optimised')
146 ax[i//2,i%2].plot(sampler, ans_unopt, linestyle='dashed', linewidth =
2, marker='o', markersize=4,label='unoptimised')
147 ax[i//2,i%2].set_xlabel('Cache Size')
148 ax[i//2,i%2].set_ylabel('Demand Miss Rate')
149 ax[i//2,i%2].set_title('a = {}, b = {}, n = {}'.format(a,b,n))
150 ax[i//2,i%2].legend(loc = 'upper right')
151 plt.savefig('../plots/varyc.png')
152 print('\n\n\n')
153
154 # variation in matrix size
155 print('
=====
')
156 print('Variation in Matrix Size')
157 print('
=====
')
158 sampler = [4, 8, 16, 32, 64, 128]
159 fig, ax = plt.subplots(2,2,figsize = (15,15))
160 fig.suptitle('Variation in n')
161 for i, (a,b,c) in enumerate([(2,64,8),(4,64,16),(2,32,8),(4,32,64)]):
162     ans_opt = []
163     ans_unopt = []
164     f = open('../csvs/variationn/{_}_{_}.csv'.format(a,b,c), 'w')
165     f.write('Opt,a,b,c,n,Total Fetches,Total Misses,Miss Rate,Compulsory
miss, Conflict miss, Capacity miss\n')
166     f.flush()
167     for n in sampler:
168         print("a = ",a," b = ",b," c = ",c," n = ",n)
```

```
169     bashCommand = form_command(a,b,c,n, optimised = True)
170     output = subprocess.check_output(['bash', '-c', bashCommand]).decode('utf8')
171     ans_opt.append(float(output.split("\n")[35].split()[3]))
172     total_fetches = output.split("\n")[31].split()[2]
173     total_misses = output.split("\n")[34].split()[2]
174     miss_rate = str(float(total_misses)/float(total_fetches))
175     miss_compulsory = output.split("\n")[36].split()[2]
176     miss_capacity = output.split("\n")[37].split()[2]
177     miss_conflict = output.split("\n")[38].split()[2]
178     f.write('{} , {} , {} , {} , {} , {} , {} , {} , {} , {} , {} \n'.format('Yes', a, b, c, n,
total_fetches, total_misses, miss_rate, miss_compulsory, miss_conflict,
miss_capacity))
179
180     bashCommand = form_command(a,b,c,n, optimised = False)
181     output = subprocess.check_output(['bash', '-c', bashCommand]).decode('utf8')
182     ans_unopt.append(float(output.split("\n")[35].split()[3]))
183     total_fetches = output.split("\n")[31].split()[2]
184     total_misses = output.split("\n")[34].split()[2]
185     miss_rate = str(float(total_misses)/float(total_fetches))
186     miss_compulsory = output.split("\n")[36].split()[2]
187     miss_capacity = output.split("\n")[37].split()[2]
188     miss_conflict = output.split("\n")[38].split()[2]
189     f.write('{} , {} , {} , {} , {} , {} , {} , {} , {} , {} , {} \n'.format('No', a, b, c, n,
total_fetches, total_misses, miss_rate, miss_compulsory, miss_conflict,
miss_capacity))
190 f.flush()
191 f.close()
192 ax[i//2,i%2].plot(sampler, ans_opt, linestyle='dashed', linewidth =
2, marker='o', markersize=4, label='optimised')
193 ax[i//2,i%2].plot(sampler, ans_unopt, linestyle='dashed', linewidth =
2, marker='o', markersize=4, label='unoptimised')
194 ax[i//2,i%2].set_xlabel('Matrix Size')
195 ax[i//2,i%2].set_ylabel('Demand Miss Rate')
196 ax[i//2,i%2].set_title('a = {}, b = {}, c = {}'.format(a,b,c))
197 ax[i//2,i%2].legend(loc = 'upper right')
198 plt.savefig('../plots/varyn.png')
199 print('\n\n\n')
```