

1. Amdahl: Welche Form nimmt die Kurve für den Speedup bei steigender Core-Zahl an? Wie erklärt sich der geringe Speedup durch mehr CPU?

2. Wie wirkt sich eine NUMA Architektur auf die Performance aus? Was sollte man tun?

3. Fügen Sie die Aktion ein

| Request    | Source    | State of<br>addressed<br>cache block | Type of<br>cache action | Function and explanation |
|------------|-----------|--------------------------------------|-------------------------|--------------------------|
| Write hit  | Processor | Shared                               | Coherence               | ??                       |
| Invalidate | Bus       | Shared                               | Coherence               | ??                       |

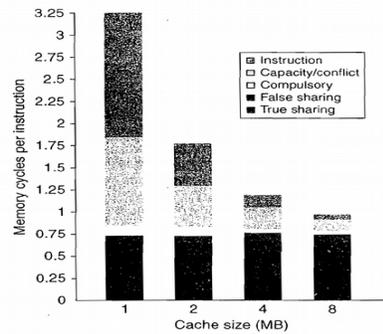
4. Erklären Sie den Effekt eines zusätzlichen „Exclusive“ bzw. „Owned“ States auf die Performance des Bus-Interconnects.

5. True or False Sharing? P1 und P2 haben X1 im Cache. Die cacheline hat X1 und X2 als Inhalt. (Tipp: compare to individual access on single word level)

(T/F)

| Time | P1       | P2       |
|------|----------|----------|
| 1    | Write x1 |          |
| 2    |          | Read x2  |
| 3    | Write x1 |          |
| 4    |          | Write x2 |
| 5    | Read x2  |          |

6. Wie wirkt sich die Verdoppelung des L3 caches auf die Parameter aus: instruction misses, conflict/capacity misses, compulsory misses, false sharing, true sharing?



7. Welchen Vorteil bringt eine LL/SS Synchronisation gegenüber einer simplen Test-and-Set Instruktion auf Multi-Cores?

```

try: LL R2,0(R1) ;load linked
      DADDUI R3,R2,#1 ;increment
      SC R3,0(R1) ;store conditional
      BEQZ R3,try ;branch store fails

```

8. Welche möglichen Abläufe ergeben sich unter Sequential Consistency?

```

P1:    A = 0;          P2:    B = 0;
       .....
       A = 1;          .....
L1:    if (B == 0)...  L2:    if (A == 0)...

```

9. Was passiert bei gleichzeitigem Aufruf beider Methoden? Wovon hängt es ab? Wie könnte man es verhindern?

```

Class foo {
public static int x;
public static int y;

static int sum_a(void)
{
int s = 0;
int i;
for (i = 0; i < 1000000; ++i)
s += foo.x;
return s;
}
static void inc_b(void)
{
int i;
for (i = 0; i < 1000000; ++i)
++foo.y;
}}

```

10. Strong vs. weak speedup?

11. BONUS Question: 2 threads perform 20000 increment operations on the same variable (no caching). What is the theoretical lower limit of the result? What result do you expect on average?