

# Memory

## Benchmark

memset and memcpy with hot memory(all memory is memseted before test)

server TStore04:

```
numactl --cpunodebind=0 --membind=0 ./memory
```

bench\GB/s	glibc	MP glibc	my	MP my
memset	8.2	20.2	18.1	27.4
memcpy	7.8	19.4	9.6	20.0

## Page fault

memset with page fault

### Page fault latency

latency(us)	Base page(1G/4096K)	THP(1G/2M)
PC	0.71	78
TStore04	1.47	460
TEAKER	2.67	606

### Page fault bandwidth

bandwidth(GB/s)	Base page	THP
PC	4.37	12.10
TStore04	2.35	3.23

### Page fault handler

```

def handle_pte_fault():
    if pagefault is valid:
        #page not present
        if not present in pte:
            if pte is none:
                do_no_page
            else:
                do_swap_page
        #page present
        else:
            if write_access while entry is COW:
                #copy-on-write
                do_wp_page

def do_no_page():
    if anonymous:
        if read access:
            use zero page
        else write_access:
            alloc page
    else:
        ## file-backed: read/COW/share
        do_fault()

```

**Clear page:** ptrace and process\_vm\_readv/writev syscall can be used to access other process's memory with some permission. So unnecessary to clear page sometimes.

## Huge page

---

Advantages:

1. less page fault
2. higher TLB hit ratio
3. less TLB miss cost, less page walk cost because of less memory access(5 layer to 4 layer?)

Issues:

1. deduplication ATC'17
2. fragmentation, fairness, promotion and demotion, page fault latency. OSDI'16