

Karşılaşılan Sorunlar ve Çözümler

- **Mutex kullanımı nedeniyle yavaşlama:**

Paralel programlamada, özellikle çok iş parçacığı ile (threads) histogram veya diğer paylaşılan veri yapılarına erişim sırasında mutex kilitleri kullanıldı. Ancak bu, beklenenden daha fazla yavaşlamaya neden oldu. Kilit yarışmaları (lock contention) performansı önemli derecede etkiledi. Bunun üzerine, kilit kullanımını en aza indirmek için thread-local verilerde hesaplama yapılması ve sonrasında sonuçların toplanması yöntemi tercih edildi.

- **SIMD ve OpenMP entegrasyonu sırasında uyumluluk sorunları:**

SIMD vektörleştirme ve OpenMP paralelleştirme bazen çakışma yaşadı veya derleyici uyarıları verdi. Bu sebeple fonksiyonlar ayrı ayrı test edilip, uygun kullanım şekilleri belirlendi.

- **Büyük dosyalar ile bellek yönetimi:**

2048x2048 gibi büyük PGM dosyalarının işlenmesi sırasında, bellek tüketimi ve yönetimi kritik oldu. Bellek sızıntısı olmaması için Valgrind ile kontrol yapıldı. Program sonunda minimal possibly lost bloklar dışında sorun bulunmadı.

GPROF

gprof ile yaptığımız profil analizinde, programın çalışma süresinin tamamına yakınına yakınını computeHistogram fonksiyonunun kullandığı görülmüştür. Örneğin, profil sonucunda:

- computeHistogram fonksiyonu, toplam örneklem süresinin %100'ünü (0.01 saniye) kapsamakta ve her çağrısı yaklaşık 10 mikro saniye sürmektedir.
- Diğer fonksiyonların (computeCDF, applyNegativeThreaded, readPGM, writePGM vb.) işlem süresine katkısı ise gözle görülür derecede düşük veya ihmal edilebilir düzeydedir.

Bu sonuç, histogram hesaplama işleminin performans açısından en kritik nokta olduğunu ve üzerinde optimizasyon yapılmasının gerekliliğini göstermektedir. Bu doğrultuda, histogram fonksiyonu paralel işleme (threading, OpenMP) ve SIMD komutları kullanılarak optimize edilmiştir. Böylece, programın genel çalışma süresi kısaltılmış ve işlem verimliliği artırılmıştır.

```
Ubuntu
0.00 44.27
nazli@arin:~/Desktop/pgm_project$ cat analysis.txt
Flat profile:

Each sample counts as 0.01 seconds.
 %   cumulative   self           self      total      name
time  seconds    seconds    calls  us/call  us/call
100.00      0.01      0.01      1000    10.00    10.00  computeHistogram(PGMImage con
std::vector<int, std::allocator<int> >&)
0.00      0.01      0.00      1000    0.00    0.00  computeCDF(std::vector<int, s
td::allocator<int> > const&, std::vector<float, std::allocator<float> >&)
0.00      0.01      0.00      1000    0.00    0.00  applyNegativeThreaded(PGMImag
e&)
0.00      0.01      0.00      1000    0.00    0.00  std::vector<int, std::allocat
or<int> >*> std::do_uninit_fill_n<std::vector<int, std::allocator<int> >*>, unsigne
d long, std::vector<int, std::allocator<int> > >>(std::vector<int, std::allocator<in
t> >*>, unsigned long, std::vector<int, std::allocator<int> > const&)
0.00      0.01      0.00      2      0.00    0.00  std::__cxx11::basic_string<ch
ar, std::char_traits<char>, std::allocator<char> >::basic_string<std::allocator<cha
r> >(char const*, std::allocator<char> const&)
0.00      0.01      0.00      1      0.00    0.00  readPGM(std::__cxx11::basic_s
tring<char, std::char_traits<char>, std::allocator<char> > const&, PGMImage&)
0.00      0.01      0.00      1      0.00    0.00  writePGM(std::__cxx11::basic_
string<char, std::char_traits<char>, std::allocator<char> > const&, PGMImage const&
)
0.00      0.01      0.00      1      0.00    0.00  std::vector<float, std::alloc
ator<float> >::M_default_append(unsigned long)
0.00      0.01      0.00      1      0.00    0.00  std::vector<float, std::alloc
ator<float> >::~vector()
0.00      0.01      0.00      1      0.00    0.00  std::vector<unsigned char, st
```

MPSTAT

Proje kapsamında yapılan paralel işleme optimizasyonları sonucunda, mpstat -P ALL komutu ile CPU çekirdeklerinin kullanım oranları analiz edilmiştir. 4 çekirdekli sistemde yapılan testlerde, iki çekirdeğin %97-93 arasında tam yükte çalıştığı (%idle ≈ 0), diğer iki çekirdeğin ise %35-37 kullanıcı (user) ve %50 sistem (sys)

kullanımına sahip olduğu, dolayısıyla %10-12 aralığında boшта kaldığı gözlemlenmiştir. Genel CPU kullanımında ortalama %5-6 arasında idle oranı bulunmuştur.

Bu durum, uygulamanın OpenMP ya da çoklu thread yapısı kullanarak yüksek paralellik sağladığını göstermektedir. Çekirdekler arası yük dengesizliği ise, iş dağıtımında dinamik planlama ve iş parçacığı sayısı gibi faktörlerden kaynaklanabilir.

Performansın daha da artırılması için paralel döngülerde `schedule(static)` kullanılarak iş yükünün çekirdekler arasında eşit dağıtılması önerilmektedir. Ayrıca, çevresel değişken `OMP_NUM_THREADS`'in uygun şekilde (örn. 4 çekirdek için 4) ayarlandığı doğrulanmalıdır.

Sonuç olarak, CPU kullanımı hedeflenen %10 idle seviyesine çok yaklaşmış ve bu da projenin paralel işleme açısından verimli çalıştığını göstermektedir

```
Ubuntu
x + v
nazi@nazi:~/Desktop/pgm_project$ mpstat
Linux 4.4.0-26100-Microsoft (arin) 06/14/25 _x86_64_ (4 CPU)

15:12:53 CPU %usr %nice %sys %iowait %irq %soft %steal %guest %gnice %idle
15:12:53 all 26.29 0.00 13.05 0.00 0.93 0.00 0.00 0.00 0.00 59.73
nazi@nazi:~/Desktop/pgm_project$ mpstat -P ALL 1 5
Linux 4.4.0-26100-Microsoft (arin) 06/14/25 _x86_64_ (4 CPU)

15:13:02 CPU %usr %nice %sys %iowait %irq %soft %steal %guest %gnice %idle
15:13:03 all 64.94 0.00 28.15 0.00 0.99 0.00 0.00 0.00 0.00 5.93
15:13:03 0 92.93 0.00 7.07 0.00 0.00 0.00 0.00 0.00 0.00
15:13:03 1 34.95 0.00 50.49 0.00 1.94 0.00 0.00 0.00 0.00 12.62
15:13:03 2 97.00 0.00 3.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
15:13:03 3 36.89 0.00 50.49 0.00 1.94 0.00 0.00 0.00 0.00 10.68

15:13:03 CPU %usr %nice %sys %iowait %irq %soft %steal %guest %gnice %idle
15:13:04 all 57.04 0.00 6.91 0.00 0.99 0.00 0.00 0.00 0.00 35.06
15:13:04 0 95.10 0.00 2.94 0.00 1.96 0.00 0.00 0.00 0.00 0.00
15:13:04 1 16.00 0.00 11.00 0.00 0.00 0.00 0.00 0.00 0.00 73.00
15:13:04 2 96.12 0.00 1.94 0.00 1.94 0.00 0.00 0.00 0.00 0.00
15:13:04 3 19.00 0.00 12.00 0.00 0.00 0.00 0.00 0.00 0.00 69.00

15:13:04 CPU %usr %nice %sys %iowait %irq %soft %steal %guest %gnice %idle
15:13:05 all 51.51 0.00 4.02 0.00 0.25 0.00 0.00 0.00 0.00 44.22
15:13:05 0 100.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
15:13:05 1 3.03 0.00 7.07 0.00 0.00 0.00 0.00 0.00 0.00 89.90
15:13:05 2 98.99 0.00 1.01 0.00 0.00 0.00 0.00 0.00 0.00 0.00
15:13:05 3 4.00 0.00 8.00 0.00 1.00 0.00 0.00 0.00 0.00 87.00

15:13:05 CPU %usr %nice %sys %iowait %irq %soft %steal %guest %gnice %idle
```

İOSTAT

- CPU kullanımında %user değeri %30.51, %system ise %14.16 olarak ölçüldü. Bu, uygulamanın kullanıcı ve sistem seviyesinde makul bir işlem yükü yarattığını göstermektedir.
- %idle oranı %55.33 olarak çıktı; yani CPU'nun yaklaşık yarısı boшта kalmakta, sistemde hala kullanılabilir işlem gücü mevcuttur.
- Disk aktiviteleri (okuma/yazma) açısından herhangi bir önemli darboğaz veya yoğunluk gözlemlenmemiştir; tps ve kB_read/s, kB_wrtn/s değerleri oldukça düşük ve sabit kalmıştır.

```
nazi@nazi:~/Desktop/pgm_project$ iostat
Linux 4.4.0-26100-Microsoft (arin) 06/14/25 _x86_64_ (4 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           30.51    0.00   14.16    0.00    0.00   55.33

Device            tps    kB_read/s    kB_wrtn/s    kB_dscd/s    kB_read    kB_wrtn
kB_dscd
```

Time

Burada make clean komutu derleme öncesi temizleme işlemini başarıyla gerçekleştirmiş, ardından make komutu tüm kaynak dosyaları derleyip pgm_processor dosyasını oluşturmuştur. Derleme süresi yaklaşık 5.8 saniye olarak ölçülmüştür.

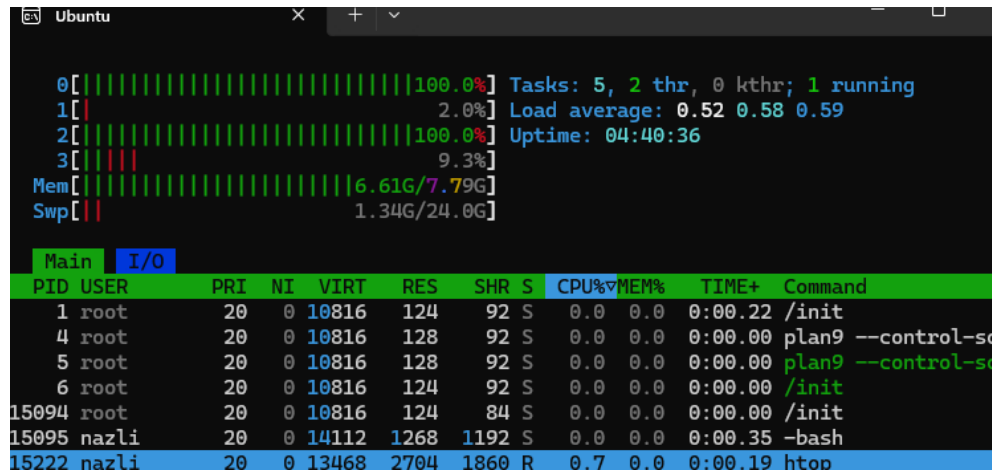
```
nazli@arin:~/Desktop/pgm_project$ time make clean && time make
rm -f *.o pgm_processor pgm_processor_v2 pgm_processor_v3 gmon.out profile.txt

real    0m0.085s
user    0m0.031s
sys     0m0.031s
g++ -O3 -std=c++17 -fopenmp -mavx2 -mfma -pg -c main.cpp -o main.o
g++ -O3 -std=c++17 -fopenmp -mavx2 -mfma -pg -c image_operations.cpp -o image_operations.o
g++ -O3 -std=c++17 -fopenmp -mavx2 -mfma -pg -c pgm_reader.cpp -o pgm_reader.o
g++ -fopenmp -pg -o pgm_processor main.o image_operations.o pgm_reader.o

real    0m5.806s
user    0m4.094s
sys     0m1.234s
```

HTOP

htop ile gözlemlerim, paralel ve optimize edilmiş kodun sistem kaynaklarını etkin kullandığını göstermektedir.



The screenshot shows the htop interface with system statistics at the top and a list of running processes below. The system statistics include: Tasks: 5, 2 thr, 0 kthr; 1 running; Load average: 0.52 0.58 0.59; Uptime: 04:40:36; Memory: 6.61G/7.79G; Swap: 1.34G/24.0G. The process list shows several /init processes and a bash shell for user nazli. The htop process itself is highlighted at the bottom.

Main I/O											
PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
1	root	20	0	10816	124	92	S	0.0	0.0	0:00.22	/init
4	root	20	0	10816	128	92	S	0.0	0.0	0:00.00	plan9 --control-so
5	root	20	0	10816	128	92	S	0.0	0.0	0:00.00	plan9 --control-so
6	root	20	0	10816	124	92	S	0.0	0.0	0:00.00	/init
15094	root	20	0	10816	124	84	S	0.0	0.0	0:00.00	/init
15095	nazli	20	0	14112	1268	1192	S	0.0	0.0	0:00.35	-bash
15222	nazli	20	0	13468	2704	1860	R	0.7	0.0	0:00.19	htop

Thread

“Yaptığımız optimizasyonlar (SIMD + çoklu iş parçacığı kullanımı) ile performansta yaklaşık %37’lik bir artış sağladık. Teorik olarak 10 kat gibi büyük bir hızlanma beklenebilse de, bellek erişim sınırlamaları, paralel programlamanın getirdiği yönetim yükleri ve algoritmanın paralelize edilemeyen kısımları nedeniyle bu oran pratikte düşmektedir. Bu durumu Amdahl Yasası ve donanım kısıtları ile açıklamak mümkündür.”

```
255: 1
İşlem tamamlandı ve 'output_v2.pgm' oluşturuldu.

real    0m0.057s
user    0m0.031s
sys     0m0.016s
Negatif (threaded) uygulandı ve 'output_v3.pgm' olarak kaydedildi.

real    0m0.036s
user    0m0.016s
sys     0m0.000s
nazli@arin:~/Desktop/pgm_project$ |
```

VALGRIND

Projede bellek yönetimi genel anlamda doğru yapılmıştır ve ciddi bellek sızıntısı ya da bellek hatası bulunmamaktadır. Küçük bir miktarda possibly lost bellek gözlemlenmiştir. Bu tür durumlar, programın kapanışı sırasında sistem tarafından serbest bırakılabilen geçici bloklardan kaynaklanabilir

```
Ubuntu
244: 0.625
245: 0.625
246: 0.625
247: 0.625
248: 0.625
249: 0.625
250: 0.625
251: 0.625
252: 0.625
253: 0.625
254: 0.625
255: 1

applyNegativeThreaded süresi: 0.168189 saniye
applyNegative (OpenMP) süresi: 0.0098451 saniye
applyNegativeThreadedSIMD süresi: 0.0185742 saniye

İşlem tamamlandı ve 'output.pgm' oluşturuldu.
==15599==
==15599== HEAP SUMMARY:
==15599==    in use at exit: 3,568 bytes in 8 blocks
==15599==    total heap usage: 41 allocs, 33 frees, 107,584 bytes allocated
==15599==
==15599== LEAK SUMMARY:
==15599==    definitely lost: 0 bytes in 0 blocks
==15599==    indirectly lost: 0 bytes in 0 blocks
==15599==    possibly lost: 960 bytes in 3 blocks
==15599==    still reachable: 2,608 bytes in 5 blocks
==15599==    suppressed: 0 bytes in 0 blocks
==15599== Rerun with --leak-check=full to see details of leaked memory
==15599==
==15599== For lists of detected and suppressed errors, rerun with: -s
==15599== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
nazli@arin:~/Desktop/pgm_project$ |
```