KARADENİZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

PARALEL BİLGİSAYARLAR

BLOK ZİNCİRİ (BLOCKCHAİN) PROJESİ

365377
Burak Tiryaki

1. KULLANILACAK ORTAMIN HAZIRLANMASI

Projede OpenCL ve C++ kullanılmıştır. SHA256 hashlerini bulmak için ise c++ içinde ve OpenCL içerisinde olmak üzere iki farklı hazır kod kullanılmıştır.

1.1. OpenCL'in Derlenmesi

```
git clone --recursive https://github.com/KhronosGroup/OpenCL-SDK.git
cd OpenCL-SDK
md build
cd build
cmake ..
cmake -DCMAKE_INSTALL_PREFIX=C:\dev\opencl .. (install dizini ayarlanabilir.)
```

Yukarıdaki komutlar çalıştırıldıktan sonra build klasörüne girip VS projesini yönetici olarak çalıştırıyoruz (install için gerekli). VS'dan Release seçtikten sonra ALL_BUILD projesine sağ tıklayıp build ediyoruz. Daha sonra INSTALL projesi içinde sağ tık ve build yapıyoruz.

Bu işlemlerden sonra belirttiğimiz klasöre lib klasörü içinde OpenCL.lib dosyası oluşmuş olacak. İnstall için belirtilen klasörde include isminde faklı bir klasör daha oluşturup OpenCL-SDK\external\OpenCL-Headers içerisindeki CL klasörünü buraya kopyalıyoruz daha sonra OpenCL-CLHPP içerisindeki CL klasörünün içindeki .h dosyalarınıda include kalsörümüzdeki CL klasörüne kopyalıyoruz.

1.2. CMakeLists.txt ve Build

Projede daha hafif olduğu için VS yerine VS Code ile çalıştım ve projeyi build etmek için CMake kullandım.

```
cmake_minimum_required (VERSION 3.3)
project(gpuden)
find_package(OpenCL REQUIRED)
add_executable(gpuden main.cpp)
target_link_libraries(gpuden OpenCL::OpenCL)
```

Projeyi build etmek ve çalıştırmak için proje klasöründe build adında bir klasör oluşturup içinde aşağıdaki komutları çalıştırıyoruz.

```
cmake ..
cmake -DCMAKE_PREFIX_PATH=C:\dev\opencl .. (install dizinimiz. Farklı olabilir.)
cmake --build PROJECT_DIR/build --config Debug --target ALL_BUILD -j 6
cd Debug
gpuden.exe
```

2. ÇÖZÜM

Projede hashler denenirken 2 yöntemle nonce değerleri threadlere dağıtıldı.

2.1. 1. Yöntem (Büyük Parçalar)

```
__kernel void den(__global char *str, __global ulong *out, __global int *cntinue,
uint strLen, uint blockNum)
{
	ulong limit = ULONG_MAX;
	ulong step = limit / get_global_size(0);

	ulong startPoint = get_global_id(0) * step;

	//*cntinue = 1;//true
	for(ulong i = 1; i <= step && *cntinue; i++)
	{
		ulong nonce = combineWithNonceAndCalcHash(str, startPoint+i, strLen,
blockNum);
		barrier(CLK_GLOBAL_MEM_FENCE);
		if(nonce != 0)
		{
			atomic_xchg(cntinue, 0);
			atom_xchg(out, nonce);
			return;
		}
	}
	return;
}
```

get_global_size(0) Fonksiyonu toplam kaç thread olduğunu gösterir. Limit değerini buna bölerek step değerini buluyoruz. Step değeri her threadın kaç tane nonce değerini deneyeceğini belirler. Örnek olarak limit değeri 100 olsun ve 4 thread ile çalışalım. Bu durumda step 25 çıkar.
İlk thread 0-25 arasındaki değerleri denerken diğer threadler sırasıyla 25-50, 50-75, 75-100 arasındaki değerleri dener.

```
ulong combineWithNonceAndCalcHash(char *str, ulong nonce, uint strLen, uint
numOfZeros);
```

Yukardaki fonksiyon verilen string ile nonce ile belirtilen değeri birleştirir ve sha256 hash hesaplar. Oluşan hash numOfZeros değişkenindeki sayı kadar sıfır ile başlarsa geriye bulunan nonce değerini döndürür. Diğer durumda sıfır döndürür.

```
atomic_xchg(cntinue, 0);
atom_xchg(out, nonce);
```

Fonksiyonlarıyla uygun nonce değeri bulunduğunda cntinue değişkenini 0(false) eşitler ve diğer threadlerin çalışmaasını durdurur. Daha sonra ise out değişkenine nonce değeri atanır.

2.2. 2. Yöntem (Küçük Parçalar)

```
__kernel void den(__global char *str, __global ulong *out, __global int *cntinue,
uint strLen, uint blockNum)
{
	ulong step = get_global_size(0);

	//*cntinue = 1;//true
	for(ulong i = get_global_id(0) + 1; i < ULONG_MAX && *cntinue; i+=step)
	{
		ulong nonce = combineWithNonceAndCalcHash(str, i, strLen, blockNum);
		barrier(CLK_GLOBAL_MEM_FENCE);
		if(nonce != 0)
		{
			atomic_xchg(cntinue, 0);
			atom_xchg(out, nonce);
			return;
		}
	}
	return;
}
```

Burda ise step değişkeni thread sayısına eşitlenmiş. Burda threadlar ilk yöntemdeki gibi ULONG_MAX değerini büyük parçalara ayırarak işlemiyor. Her thread kendi ID'sine step değişkenini ekleyerek ilerliyor. Yine örnek olarak 100 değer deneyecek olursak ve 4 thread çalıştırırsak threadler 1-2-3-4 değerleriyle başlayacak 5-6-7-8 değerleriyle devam edecek en son 97-98-99-100 olduğunda bitecek.

Diğer yöntemden farklı olarak bulunan nonce değerleri genelde daha düşük değerler çıkıyor. Çünkü diğer yöntemde ULONG_MAX değişkeni büyük parçalara bölünüp threadlere dağıtılıyor 2. Threadden sonra bulunması halinde (çok büyük ihtimalle) bulunan nonce değeri otomatik olarak çok büyük çıkıyor. İlk threadın bulmadığı durumlarda en az $(2\text{^}64 - 1)/(thread\ sayısı)$ kadar.

3. SONUÇ

Thread'lerin özellikle ilk blokda birçok nonce değerini aynı anda bulmasından dolayı ilk nonce değerleri programı her koştuğumuzda farklılık göstermektedir. Bu farklılık hash olarak farklı bir çıktı oluşturduğundan diğer bloklarıda etkilemektedir. Thread sayısını artırmakla genel olarak fayda sağlamış olsak bile bu sorundan dolayı hesaplama sürelerinde şans faktörüde büyük önem oynamakta.

Şans faktörününden bağımsız olarak thread sayısının hızlanmaya etkisini görmek için sabit bir string verisini istenilen sıfır değerini buluncaya kadar hash hesaplayan ikinci bir program daha yazıldı. Bu sayede thread sayısının benchmark etkisi daha anlaşılır oldu.

3.1. 1. Yöntem vs 2. Yöntem

Başlangıç String: 1Bize her yer Trabzon! Bölümün en yakışıklı hocası İbrahim Hoca'dır0000000000000000000000000000000000000000000000000000000000000000
Thrad Sayısı: 3200

1.Yöntem Sonuçları:

```
1. block found
Elapsed time: 1.95478 ms
Nonce: 11857797674881419339
Hash: 091f0c80da3303f71db3a3cc6fc62749f855f930b18187895b54d7cf0282be90
----------------------------------------------------------------

2. block found
Elapsed time: 0.268192 ms
Nonce: 3758524105018320569
Hash: 0044c98b8bf429132ad7506a17482f42c3312b7371ca9e207cf23ab989c87714
----------------------------------------------------------------

3. block found
Elapsed time: 0.387072 ms
Nonce: 1026100139100093653
Hash: 000d320e67a8a9b99688e7f3ee1e29d60ffc16030df5e0edb2bd061472c7cab4
----------------------------------------------------------------

4. block found
Elapsed time: 7.30522 ms
Nonce: 1723617649387235992
Hash: 0000c26b9f50b2a809f78c7c70ea1d6a097ee2f57d6a6d87ac204acd3ae63ac1
----------------------------------------------------------------

5. block found
Elapsed time: 95.5464 ms
Nonce: 8156919645093441432
Hash: 000006ae0a3815074c92f90b94657920d05c81162b0d5d3f65ee2b0c5a45cfe8
----------------------------------------------------------------

6. block found
Elapsed time: 2016.62 ms
Nonce: 14129053038956912613
Hash: 00000078368dda1ab9a627be8f2c582654cc4aef9d30b825c2b01400c49b22ba
----------------------------------------------------------------

7. block found
Elapsed time: 18743.4 ms
Nonce: 15852670688344185330
Hash: 00000008d2a5d9a74c5e840a37f5d6f85bbfb3f1571b62c2037227b3419d18cd
----------------------------------------------------------------

8. block found
Elapsed time: 54611.7 ms
Nonce: 17870283321406229023
Hash: 00000000d7c0a99319e77f3cb94be94578ab8e1e11b89fe4cc1970491d2c19cb
----------------------------------------------------------------
```

2.Yöntem Sonuçları:

```
1. block found
Elapsed time: 2.29446 ms
Nonce: 2183
Hash: 0b17befdb96823ca76273e69f229a583e9ef3bd955f01f314b9cf12da495d53b
-----------------------------------------------------------------

2. block found
Elapsed time: 0.201248 ms
Nonce: 3092
Hash: 0031db5bb823331341ff38ee94a3bda69ad2eb6f7d75a3b266dd6dabb2ee5865
-----------------------------------------------------------------

3. block found
Elapsed time: 0.615424 ms
Nonce: 5815
Hash: 000fb3046e725c1509953d90955f465376cf4afdaf49842cd4f45ae6b6c3deae
-----------------------------------------------------------------

4. block found
Elapsed time: 11.6906 ms
Nonce: 145593
Hash: 00007571eade2553ab686c6a13239392942cadd5f9ba294aed8ed4b91b7722dc
-----------------------------------------------------------------

5. block found
Elapsed time: 26.3033 ms
Nonce: 272219
Hash: 0000076cc1d4ba5893499225b5d08ef6fe1aabc8a3fa1d738a54dcee73db0b6c
-----------------------------------------------------------------

6. block found
Elapsed time: 4015.58 ms
Nonce: 44778384
Hash: 0000004d59d197764c1fe5af029b31af16891cfc5476bb2edc19450d1c23dfe8
-----------------------------------------------------------------

7. block found
Elapsed time: 456849 ms
Nonce: 10058842923
Hash: 000000071a03dc1b4eee46a9da66e7d051fde6d16a4ea364e6ec6fab2981001c
-----------------------------------------------------------------

8. block found
Elapsed time: 225588 ms
Nonce: 1716379724
Hash: 000000007fe8f0f53861bf2db027e4d40a1b0360efa7ff8443eb51a7fa89b92b
-----------------------------------------------------------------
```

Not: Sonuçlar https://emn178.github.io/online-tools/sha256.html adresinden kontrol edilebilir.

3.2. 9.Blok (1.Yöntem)

Başlangıç String: 1Bize her yer Trabzon! Bölümün en yakışıklı hocası İbrahim Hoca'dır0000000000000000000000000000000000000000000000000000000000000000
Thrad Sayısı: 2048

```
1. block found
Elapsed time: 1.8039 ms
Nonce: 15699548301013547314
Hash: 07dbbaf4993f5bc645063e10f4d679d32cc431a49cb94ffb7651d21199b676a6
----------------------------------------------------------------

2. block found
Elapsed time: 0.29184 ms
Nonce: 5674535530486824331
Hash: 0067f838d02f46edaee7ce4c8c0678569c9cb4b2a2be6316ced992dc57a5b2b5
----------------------------------------------------------------

3. block found
Elapsed time: 1.60154 ms
Nonce: 5845672316326903165
Hash: 000984e4e49b387ff67d7efdcd0744286e741e2d3ff3b8bfb941a11e950586b8
----------------------------------------------------------------

4. block found
Elapsed time: 22.6188 ms
Nonce: 4530621225134718553
Hash: 0000bd10e04e49ff0ac62949cc6292877c13913a63f5b6218073428a0563ef70
----------------------------------------------------------------

5. block found
Elapsed time: 126.391 ms
Nonce: 16996584993696250405
Hash: 00000220cbdab9dcb99ba80b212392b46a5166bcc92be45b8ee12ab89c7589a1
----------------------------------------------------------------

6. block found
Elapsed time: 3749.82 ms
Nonce: 12483978167071024534
Hash: 000000af07436cb608515e40bfc3b3d2ad86397d382a25de17cb1a9784a49205
----------------------------------------------------------------

7. block found
Elapsed time: 60903.4 ms
Nonce: 16140901064496027981
Hash: 00000002b6d128153342eb08b264fdb7468edde6b0835b53504da73c63b23d4d
----------------------------------------------------------------

8. block found
Elapsed time: 120700 ms
Nonce: 17519002550471534626
Hash: 00000000a0ba7c43ce60127f4bf4b464a05e5d8d3b55b9c832be2d3361c1fe70
----------------------------------------------------------------

9. block found
Elapsed time: 2.82883e+06 ms
Nonce: 14474569202375205115
Hash: 0000000004945cf8615604df6940d9534f82e0fd45898e5f2445ee80e11f3731
----------------------------------------------------------------
```

3.3. Tek Hash – 7 Sıfırlı Nonce Bulma – 2.Yöntem (Thread karşılaştırma)

Bu karşılaştırmada sabit "burak" string'i üzerinde 7 sıfırlı nonce aranmıştır. Yöntem olarak 2. Yöntem kullanılmıştır. Thread sayısı sürekli azaltılmış ve zaman incelenmiştir.

Başlangıç String: burak

Thrad Sayısı: 3200

```
Elapsed time: 1242.27 ms
[main]nonce: 137084089
```

Thrad Sayısı: 2560

```
Elapsed time: 1444.46 ms
[main]nonce: 137084089
```

Thrad Sayısı: 1920

```
Elapsed time: 1646.29 ms
[main]nonce: 137084089
```

Thrad Sayısı: 1280

```
Elapsed time: 1872.56 ms
[main]nonce: 137084089
```

Thrad Sayısı: 640

```
Elapsed time: 3347.36 ms
[main]nonce: 137084089
```

Thrad Sayısı: 320

```
Elapsed time: 6616.39 ms
[main]nonce: 137084089
```

3.4. 1.Yöntem - Nvidia Tesla k80 GPU

1.Yöntemi Microsoft Azure Öğrenci hesabından açılan GPU'lu sanal makinede koşuldu. Kendi ekran kartım GTX1050'de 3200 thread 2.yöntemde ve tek hashlerde verimli olurken serverda Tesla k80 ile 12800 thread arttırdığımda verim arttı. Bu yüzden thread sayısını serverda 12800 yaptım.

Server 4 saatlik sürede 10. Bloğu buldu. Daha sonra 11. Bloğu hesaplarken yaklaşık 21 saatlik bir süre geçmesine rağmen bulunamadı ve istemli olarak durduruldu.

Başlangıç String: 1Bize her yer Trabzon! Bölümün en yakışıklı hocası İbrahim Hoca'dır0000000000000000000000000000000000000000000000000000000000000000
Thrad Sayısı: 12800

```
Selected device: Tesla K80
Num of threads: 12800

1. block found
Elapsed time: 6.11994 ms
System Time: Wed Jun  9 21:06:38 2021
Full String: …
Nonce: 5020973152562816073
Hash: 0cd59583c994a51667d3d51284203e861f11516ee97c22c1d11bedad7a012cd1
----------------------------------------------------------------

2. block found
Elapsed time: 0.54512 ms
System Time: Wed Jun  9 21:06:38 2021
Full String: …
Nonce: 14977891496723693295
Hash: 00b8b063de5f29744bc847a1240a224aab1a8e74765cea47c54efde0954248a3
----------------------------------------------------------------

3. block found
Elapsed time: 1.25824 ms
System Time: Wed Jun  9 21:06:38 2021
Full String: …
Nonce: 9611041892778823303
Hash: 0002ebad115d3d1697107466a92f309f39b3dcebe2f721dc7ceff7f560ec8b1a
----------------------------------------------------------------

4. block found
Elapsed time: 3.95456 ms
System Time: Wed Jun  9 21:06:38 2021
Full String: …
Nonce: 14686778816810464582
Hash: 0000a84980827e5cf563f34bb174ef68d7ab56a47f6e64c9b93950625b220561
----------------------------------------------------------------

5. block found
Elapsed time: 7.18262 ms
System Time: Wed Jun  9 21:06:38 2021
Full String: …
Nonce: 15750348904810280389
Hash: 00000a4635c97b5082cee08f7743fca0c220617b7aa6d11fbdd864e4e6c0307c
----------------------------------------------------------------

6. block found
Elapsed time: 223.895 ms
System Time: Wed Jun  9 21:06:39 2021
Full String: …
Nonce: 18142661026869486875
Hash: 0000008c9c091d2f3a8019de93757954ce6993ec24a8164d7a38fc4adf0426a4
----------------------------------------------------------------
```

```
7. block found
Elapsed time: 16192.8 ms
System Time: Wed Jun  9 21:06:55 2021
Full String: …
Nonce: 11847709611716122781
Hash: 00000001d520434dc712392282fe40ea93eb61efb6a4c5b71959ab987bbc48c6
----------------------------------------------------------------

8. block found
Elapsed time: 513.977 ms
System Time: Wed Jun  9 21:06:55 2021
Full String: …
Nonce: 17799666879248950364
Hash: 00000000f1b5b7ca826818e1050bc8b14198dbcbc9ab2b98957119040760b7c2
----------------------------------------------------------------

9. block found
Elapsed time: 4.9318e+06 ms
System Time: Wed Jun  9 22:29:07 2021
Full String: …
Nonce: 8426415046799330048
Hash: 0000000008b950e9dfb67dd7e436219054524d4e96d1264f229e8dbf27521ee4
----------------------------------------------------------------

10. block found
Elapsed time: 7.92839e+06 ms
System Time: Thu Jun 10 00:41:15 2021

Full String: 10987654321Bize her yer Trabzon! Bölümün en yakışıklı hocası İbrahim
Hoca'dır0000000000000000000000000000000000000000000000000000000000050209731525628
160730cd59583c994a51667d3d51284203e861f11516ee97c22c1d11bedad7a012cd114977891496723693
29500b8b063de5f29744bc847a1240a224aab1a8e74765cea47c54efde0954248a39611041892778823303
0002ebad115d3d1697107466a92f309f39b3dcebe2f721dc7ceff7f560ec8b1a14686778816810464582000
00a84980827e5cf563f34bb174ef68d7ab56a47f6e64c9b93950625b22056115750348904810280389000 0
0a4635c97b5082cee08f7743fca0c220617b7aa6d11fbdd864e4e6c0307c18142661026869486875000000
8c9c091d2f3a8019de93757954ce6993ec24a8164d7a38fc4adf0426a411847709611716122781000000001
d520434dc712392282fe40ea93eb61efb6a4c5b71959ab987bbc48c617799666879248950364000000000f1
b5b7ca826818e1050bc8b14198dbcbc9ab2b98957119040760b7c284264150467993300480000000008b95
0e9dfb67dd7e436219054524d4e96d1264f229e8dbf27521ee42775658522347146907
Nonce: 2775658522347146907
Hash: 0000000000393ad213a4de82559d4730f47404ded2294b5fcfb5c702d7477016
burak@btvm:~/opencl/multiple-hash/outputs$ date
Thu Jun 10 21:07:52 UTC 2021
```

3.5. 2.Yöntem – Tek Hash (Nvidia Tesla K80 VS GTX1050)

Karşılaştırmanın doğruluğu açısından 2.yöntem ve tek hash hesaplanmıştır. Giriş stringi olarak "burak" kullanılmıştır. Nonce değerleri ile birlikte https://emn178.github.io/online-tools/sha256.html adresinden kontrol edilebilir.

GTX1050

Thrad Sayısı: 3200

```
Elapsed time: 91662 ms
[main]nonce: 10111180163
```

Tesla K80

Thrad Sayısı: 12800

```
Elapsed time: 56624.8 ms
[main]nonce: 10111180163
```

Tesla K80

Thrad Sayısı: 6400

```
Elapsed time: 72234.6 ms
[main]nonce: 10111180163
```

4. KAYNAKLAR

https://github.com/Fruneng/opencl_sha_al_im
https://github.com/okdshin/PicoSHA2
https://github.com/KhronosGroup/OpenCL-SDK
https://www.khronos.org/opencl/
https://www.khronos.org/registry/OpenCL/