Yüksek Ölçekli Paralel Atomistik Bilgisayar Simülasyonu: Yüzey-Iyon Çarpismasi

Murat ATIS
Fizik Bölümü
Kirikkale Üniversitesi

atis@siber.cankaya.edu.tr

Cem ÖZDOGAN
Bilgisayar Mühendisligi Bölümü
Çankaya Üniversitesi
ozdogan@cankaya.edu.tr

Ziya B. GÜVENÇ
Elektronik ve Haberlesme Mühendisligi Bölümü
Çankaya Üniversitesi
guvenc@cankaya.edu.tr

ÖZET

Bu çalismada, Ar iyonu – Ni yüzeyi çarpismasini simüle eden bilgisayar programi paralellestirildi. Sistemdeki atomlar tamamen bilgisayarlara paylastirildi. Paralel kodda bilgisayarlar arasinda bilgi transferi için geçen sürenin minimum ve atom sayisinin artmasina ragmen sabit bir degerde tutulmasi, islem süresinin ise en az ve atom sayisi ile düzgün orantili hale getirilmesi amaçlandi. Bunun için her bilgisayarin yalniz ihtiyaci oldugu kadar bilgiyi, o bilginin bulundugu bilgisayardan almasi saglandi. Ayrica atomlarin sadece kendi komsuluk mesafesindeki diger atomlarla etkilesimlerinin hesabi yapildi. Bunun sonucu olarak seri kodun ulasabilecegi en yüksek atom sayisi 190 bin ve bu sayida tek bir zaman adiminin gerçeklestirilmesi için geçen süre 79.3 saniye iken, ayni sayidaki fiziksel yapi için tek bir zaman adimi 2 makineli paralel sistemimizde ≈32 saniye, 8 makinelik sistem için ≈8 saniye süreye düsmüstür. Paralel kod 8 bilgisayar ile 1.5 milyon atom sayili sistemleri (tek bir zaman adimini 61.6 saniyede tamamlayacak sekilde) çalisabilmektedir. Ayrica bilgi transferi için harcanan zamanin her atom ve bilgisayar sayisi için sabit olmasi saglandigindan dolayi, paralel kodun 8 PC' ye kadar islem süresindeki kazancinin devam ettigi ve islem süresinin atom sayisiyla yaklasik lineer oldugu görüldü. Bilgisayarlar arasinda bilgi alisverisini saglamak için "Paralel Virtual Machine" (PVM) programi kullanildi. Kullanilan paralellestirme yöntemi Single Instruction Multiple Data (SIMD) yapisindadir. Program Pointer destekli Fortran77 dilinde yazildi ve derlendi. Böylece derleme ve çalisma asamalarindaki sinirlamalardan büyük ölçüde kurtulundu. Ayrica hafizanin dinamik kullanilmasi saglanmistir. Paralel kod, herbiri PIII 1GHz islemcili, 256MB RAM'a sahip, 100Mbit/sec (fast) ethernetli, Linux (Slackware 2.2.19 kernel) kurulu 8 PC' lik dagitik sistemli, birbirlerine switcher ile bagli bizim tarafimizdan kurulan bir cluster üzerinde test edildi.

1. GIRIS

Teorik ve deneysel gelismelerin yaninda, bilgisayar dünyasindaki gelismeler, birçok arastirmanin bilgisayar ortaminda simülasyonlarla yapilmasi yolunu açmis, bilgisayar simülasyonlarina dayali hesaplama yöntemleri gelismistir. Ayrica deney ve teorinin yeterli olmadigi durumlarda bilgisayar benzetiminden faydalanmak kaçinilmaz hale gelmistir.

Fizik alaninda da, mikroyapidaki olaylarin ve özelliklerin simülasyon teknigi ile incelenmesi önemli bir adim olmustur. Atomlar arasindaki etkilesimleri modellemek için çok hassas kuantum mekaniksel hesaplama yöntemlerinden, daha basit yari deneysel yöntemlere kadar birçok hesaplama yöntemi gelistirilmistir. Fiziksel bir sistemin, bilgisayar ortaminda benzetimi onun küçük bir modelidir. Bu sisteme en yakin benzetimi elde edebilmek için, sistem gerçek sistemin makroskobik özelliklerini yansitacak kadar büyük olmali ve simülasyon zamani fiziksel özelikleri gözleyebilecek kadar uzun olmalidir. Bunu saglamak için limitler, sistem büyüklügü için 10^4 ile 10^6 atom arasi, simülasyon süresi için 10^{-12} ile 10^{-3} saniye araligidir. Bilgisayar dünyasinin hizli gelisimi bile bu ihtiyaci karsilayacak seviyeye henüz ulasamamistir. Bu yolda, yardimci olabilecek yöntemlerden biriside paralellestirmedir. Paralellestirme, birkaç bilgisayarin, bir problemi paylasarak ayni anda çözmeleri isidir. Bunun için incelen fiziksel sistem bilgisayarlara paylastirilir ve gerekli yerlerde bilgi transferi ile problemi birlikte çözmeleri saglanir. Bilgisayarlar arasindaki veri transferini saglamak için "Paralel Virtual Machine" (PVM) ve "Message Passing Interface" (MPI) en çok kullanilan iki yöntemdir. Fiyat/verim orani düsük, yüksek performansli bir paralel laboratuar için, dagitik sistemli Linux isletim sistemi kurulu ayni özelliklere sahip birkaç bilgisayardan olusmus bir paralel bilgisayar laboratuari uygun bir seçimdir.

En uygun paralellestirme, en az hafiza kullanimi ve bilgi transferi zamani ile programin en hizli sonuç verecegi algoritmayi saglayarak elde edilir. Çalistigimiz sistem için iki farkli paralellestirme yönteminden bahsedilebilir. Birincisi, bütün verilerin her bilgisayarda bulundugu, sadece yapilacak islerin

paylastirildigi yöntemdir (Atomic Decomposition). Bu yöntemde seri kodu paralellestirmek kolaydir ve islem zamanindan kazanç saglar. fakat hafiza kullanimi iyi degildir ve veri transferi için gecen süre nisbeten daha fazladir. Ikinci yöntem, sistemin tamamen bilgisayarlara bölündügü yöntemdir (Domain Decomposition). Uygulamasi zordur, fakat hem islem zamani hem de hafiza kullanimindan kazanç saglar. Bütün bilgi transfer edilmediginden iletisim zamani fazla degildir [2] .

2. MOLEKÜLER DINAMIK ADIM

Bu çalismada, bir Ar iyonunun Ni kati cisminin yüzeyine çarpmasi simüle edildi. Ni kati cismi hedef yüzeyi z=0'da bulunmak üzere, z-ekseni boyunca (46x46 atomluk) atom düzlemlerinden olusmaktadir. Ni kati cisminin en alt atom tabakasi, cismin dikdörtgenler prizmasi olan seklini, çarpisma sirasinda korumasi için sabit tutuldu. Ar iyonu bir baslangiç yüksekliginden, z=0'daki Ni yüzeyine bir baslangiç kinetik enerjisiyle gönderilmektedir. Çarpisma sonrasi Ar iyonu ile Ni atomlarinin durumlari incelenmektedir. 400 farkli çarpisma ile bütün yüzey ihtimaliyetlerinin taranmasi saglanmistir. Atomlar arasindaki etkilesmelerin hesabi için, Daw ve Baskes [1] tarafindan bulunan "Embedded Atom Model" (EAM) metodu kullanildi. Moleküler dinamik adimlar Haming'in Tahmin et Düzelt algoritmasiyla islettirildi. Herbir Moleküler dinamik simlasyon zaman adimi, yüzde 0.001'lik enerji korunumunu saglayacak sekilde, 45×10^{-16} saniye olarak ayarlandi.

2.1. Seri Moleküler Dinamik Adim

Seri kod Pointer destegi ile minimum ve dinamik hafiza kullanimi gözönünde bulundurularak optimize edildi. Bu sartlar altinda seri kodun maksimum 190 bin atomlu sistemi çalisabildigi görüldü. Bu seviyedeki islemlerin çok uzun sürmesi ve daha yüksek atom sayili sistemleri çalisabilmek için programin paralellestirilmesi geregi görüldü.

2.2. Paralel Moleküler Dinamik adim

Seri kodda simüle edilen sistem, tüm paralel bilgisayarlara paylastirilarak kod paralellestirildi. Her bilgisayarin yanliz kendi atom bilgisini bulundurmasi ve dizi boyutlarinin buna göre açilmasi saglandi. Islem zamanindan kazanç saglamak için, iki döngüden olusan (Sekil-1) ve islem sayisinin atom sayinin karesine bagli komsuluk hesabi $\left(O\left(N_{atom}{}^{2}\right)\right)$, üç döngüden olusan (Sekil-2) ve islem sayisi atom sayisi ile dogru orantili degisen komsuluk hesabi kulanildi $\left(O\left(N_{atom}\right)\right)$ (N_{atom} = Tabaka Sayisi * Tabakadaki Atom Sayisi olmak üzere). Bunun için, her atomun yanliz, kendi bulundugu atom tabakasindaki atomlarla ve etkilesme mesafesindeki komsu atom tabakalarindaki atomlarla etkilesebilecegi gözetildi. Geriye kalan atom bilgisi, komsuluk hesabinda kullanilmadi, bilgisayarlar arasinda transfer edilmedi ve hafizada tutulmadi. Gerekli atom bilgisi için, yanliz uygun bilgisayarla haberleserek alindi. Böylece gereksiz iletisimden kaçinildi.

```
DO I=1,ATOM SAYISI
   ...
   DO J=1,ATOM SAYISI
      ...
      ...
   ENDDO
...
ENDDO
```

Sekil-1. iki öngülü komsuluk hesabi.

```
DO I=1, TABAKA SAYISI
   ...
   DO J=1, TABAKADAKI ATOM SAYISI
      ...
      DO K=ETKILESME MESAFESINDE
            BULUNAN TABAKALARDAKI
            ATOM SAYISI
      ...
      ...
    ENDDO
   ...
   ENDDO
...
ENDDO
```

Sekil-2. Üç döngülü komsuluk hesabi.

3. SONUÇ VE TARTISMA

Paralel kod, Linux isletim sistemi kurulu, PVM (Paralel Virtual Machine) kütüphane fonksiyonlarini kullanan, birbirlerine fast ethernet (100Mbit/s) ve switcher ile bagli, dagitik sistemli, 8 adet Pentium-III 1000MHz islemcili PC'lerden olusan, bizim tarafimizdan kurulan bir paralel bilgisayar laboratuarinda test edildi. Paralel ve seri kodun hizlanma ve verim degerleri Tablo-1 ve Tablo-2'de verildi. Hizlanma degeri $Hizlanma = T_{seri}/T_{paralel}$ seklinde seri ve paralel kodun islem sürelerinin (T_{seri} ve $T_{paralel}$) orantilanmasi ile bulundu. Verim ise $Verim = Hizlanma/\left(Islemci\ say.\right)$ seklinde bulundu. Tablolardan görüldügü gibi, en yüksek Hizlanma degeri 9.25 olarak görüldü. Bu hizlanma degeri $46\times46\times152$ (herbir tabakada 46×46 atom ve 152 tabaka için 160816 atom) için 8 islemci ile elde edildi. En yüksek verim degeri ise, 1.38 olarak

belirlendi. Bu verim $46 \times 46 \times 36$ (38088 atom) için 6 islemci kullanilarak elde edildi.

Tablo-1. Farkli atom ve islemci sayilari için paralel kodun hizlanmasi.

Atom sayisi	2	3	4	5	6	7	8
10580	2,02	-	-	-	-	-	
21160	2,28	2,77	2,37	-	-	-	-
38088	2,45	3,40	4,65	5,72	8,28	5,13	-
69828	2,53	3,89	4,45	5,46	6,51	7,00	8,25
80408	2,38	3,53	4,16	5,66	6,14	5,72	7,00
101568	2,45	3,63	4,63	5,52	5,68	6,64	7,96
110032	2,48	3,52	4,63	5,27	6,79	6,34	8,91
131192	2,49	3,62	4,79	5,46	6,76	6,56	8,09
150236	2,43	3,82	4,58	5,05	6,71	7,69	7,94
169280	2,57	3,70	4,97	6,20	6,78	7,23	8,52
190440	2,55	3,68	4,74	5,51	6,83	7,70	9,12

Tablo-2. Farkli atom ve islemci sayilari için paralel kodun verimi.

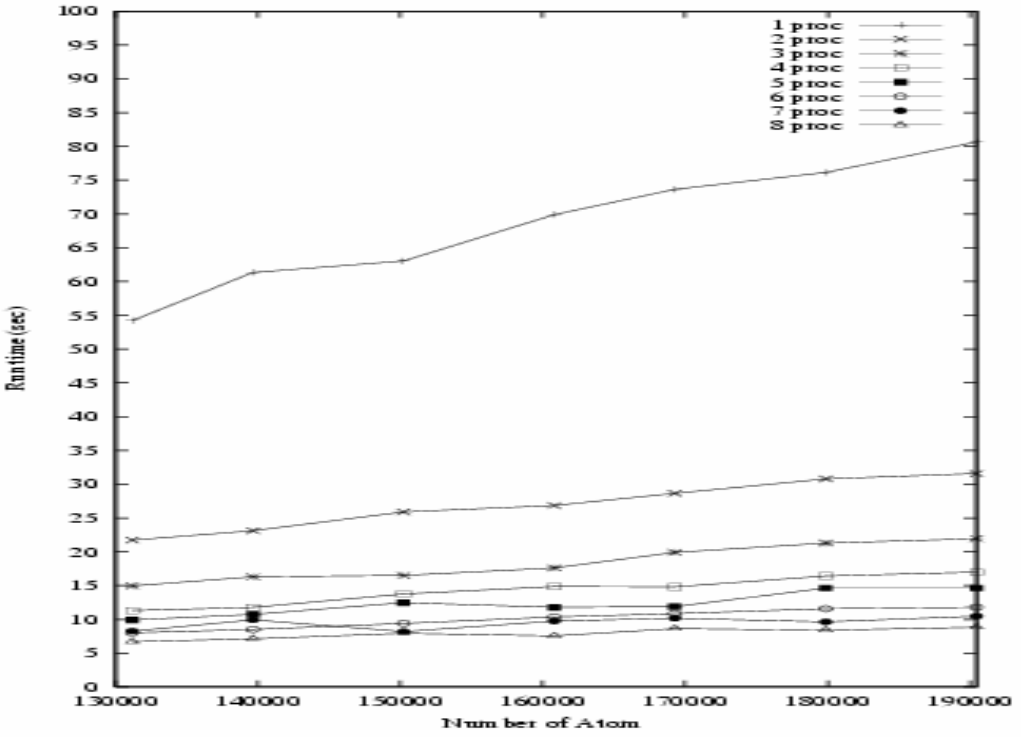
Atom sayisi	2	3	4	5	6	7	8
10580	1,01	-	-	-	-	-	
21160	1,14	0,92	0,59	-	-	-	-
38088	1,23	1,13	1,16	1,14	1,38	0,73	-
69828	1,26	1,30	1,11	1,09	1,09	1,00	1,03
80408	1,19	1,18	1,04	1,13	1,02	0,82	0,88
101568	1,23	1,21	1,16	1,10	0,95	0,95	0,99
110032	1,24	1,17	1,16	1,05	1,13	0,91	1,11
131192	1,25	1,21	1,20	1,09	1,13	0,94	1,01
150236	1,22	1,27	1,15	1,01	1,12	1,10	0,99
169280	1,28	1,23	1,24	1,24	1,13	1,03	1,07
190440	1,28	1,23	1,19	1,10	1,14	1,10	1,14

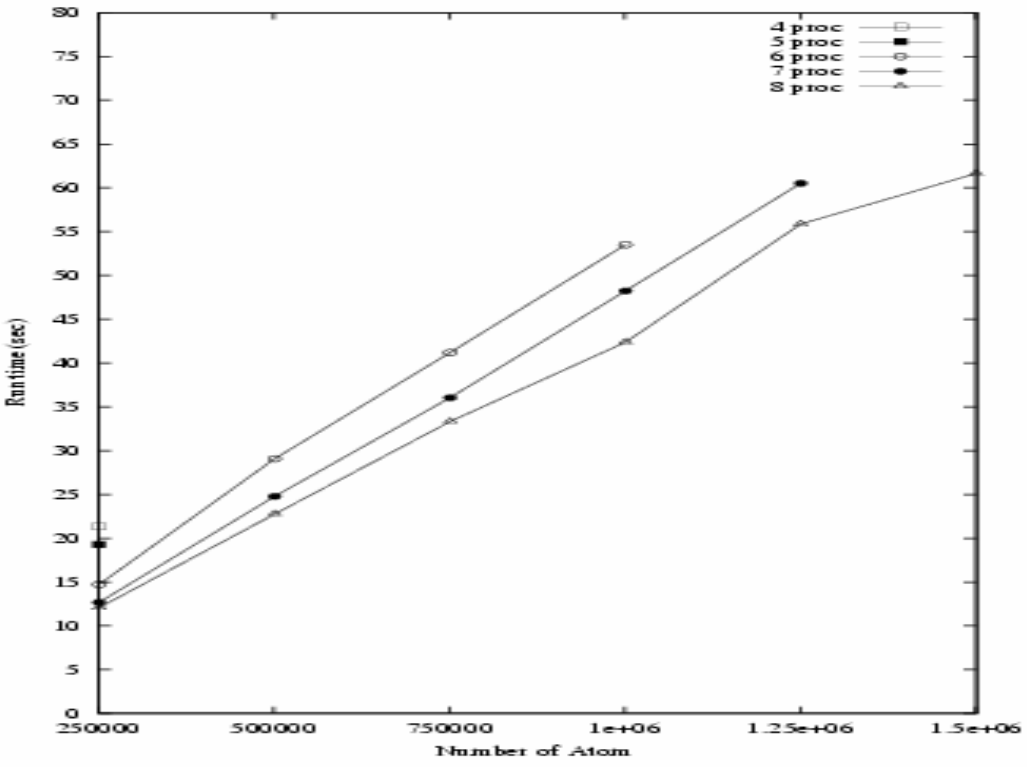
Yukarida verilen formüllerden anlasilacagi gibi, hizlanma için elde edilebilecek en yüksek deger sistemdeki bilgisayar sayisi ve verim için 1 olmasi gerekmektedir. Ama bizim verdigimiz tablolarda görülen durum, bu teorik degerlerden daha yüksek bir davranis göstermektedir. Bunun sebebi optimize edilmis olan seri koddaki komsuluk hesabi yapan kisimlarin paralel kodla ayni yapida olmasina ragmen tüm atom sayisi üzerinden hesaplarin tek makinede yapilmasidir. Seri kodun paralellestirilmesibilgisayar üzerine düsen atom sayisini önemli ölçüde azaltmasi hizlanma ve verim kavramlarindaki bu yüksek degerlerin sebebidir.

Farkli islemci sayilari için, sistemin atom sayisinin artmasi ile paralel kodun islem süresinin degisimi grafik halinde verildi (Sekil-3). Üç döngülü komsuluk hesabinin kullanilmasindan dolayi islem süresinin atom sayisi ile lineer degisimi beklenmekteydi. Sekilde islem süresinin atom sayisi ile dogru orantili degisimi görülmektedir

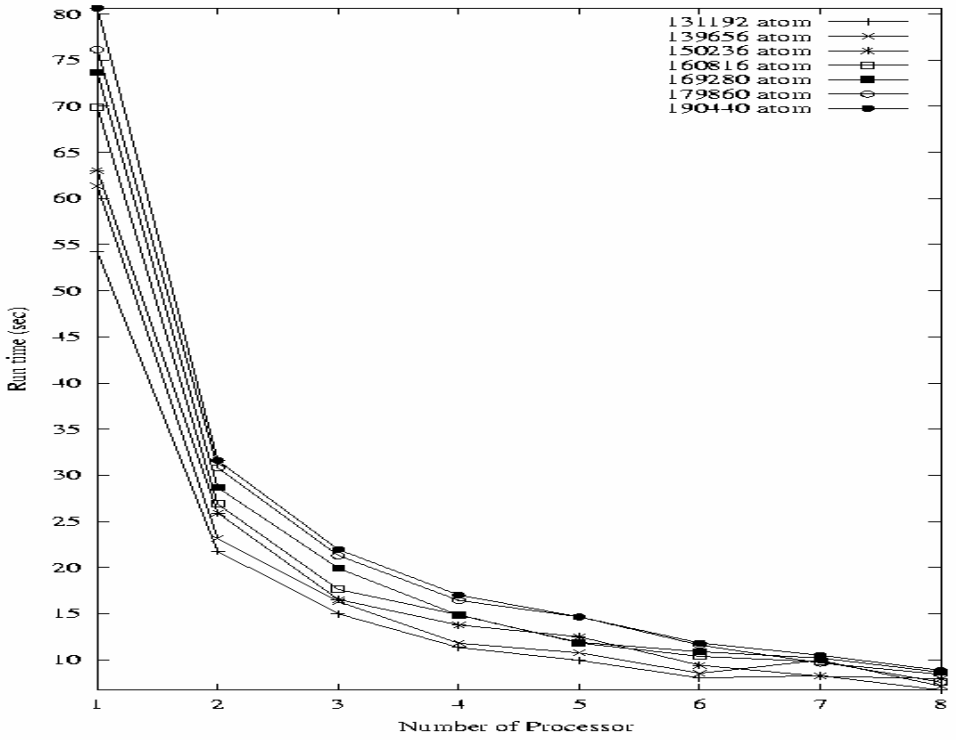
Algoritmanin, her bilgisayarin yanliz ihtiyaci oldugu kadar bilgiyi, yanliz gerekli bilgisayarla haberleserek alacak sekilde kuruldugundan, islemci sayisinin artmasina ragmen islem süresinden kazancin devam ettigi görülmüstür (Sekil-4). Ayrica, transfer edilen atom bilgisinin ölçüsü etkilesim mesafesi oldugundan ve etkilesim mesafesi sabit oldugundan, iletisim zamaninin sabit olmasi beklenmekteydi. Her bilgisayarin, iletisim süresinin ortalamasinin sabit oldugu görülmüstür (Sekil-5).

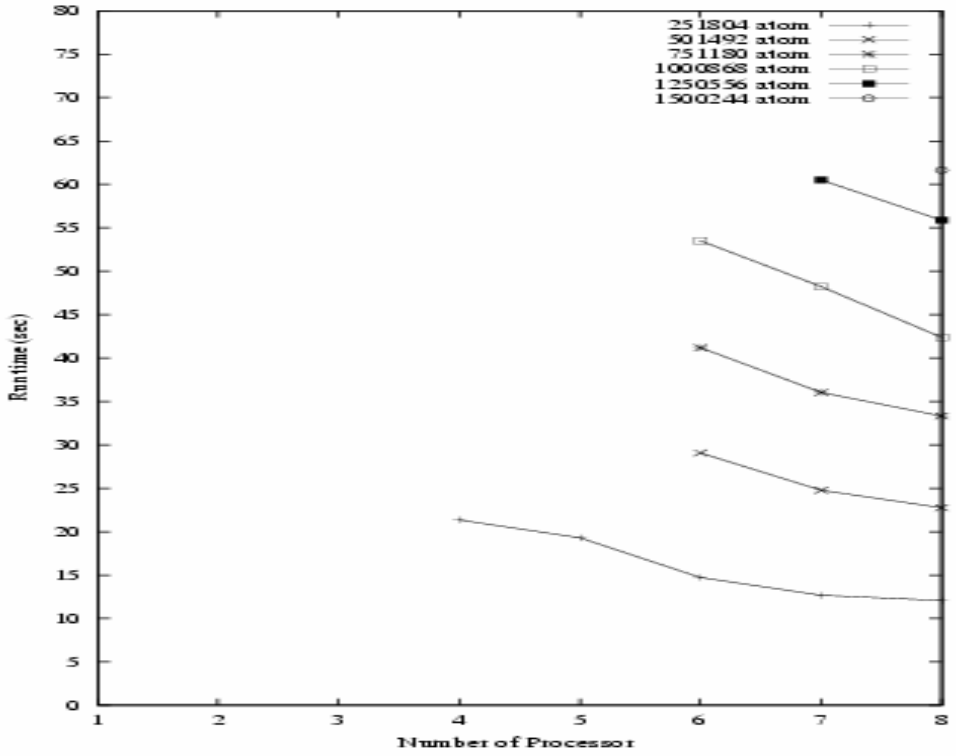
Bu çalismada kullanilan paralellestirme yöntemi, kullanilan komsuluk sayma sistemi ile gayet basarili bir paralellestirme yöntemi olarak alinabilir. Bu yöntem sayesinde, islem süresinin atom sayisi ile dogru orantili degisimi ve sabit iletisim zamani elde edilmistir. Bunu yaparken hafizaya ve islemciye ek sayilabilecek bir yük olmamasi da bir avantajidir. Yaptigimiz çalismada geldigimiz nokta olan 1.5 milyon atom sayisi elde hazir bulunan bilgisayar sistemlerinin donanim sinirlamasidir. Paralel çalisamalarda sisteme eklenen her yeni makinenin kazanç getimesi ama bir sinir degerde artik kazanç degil kayip getirmesi beklenir. Ama eldeki sonuçlar bu sinirlamanin asildigini ve oldukça büyük sistemleri kisa sayilabilecek sürede inceleyebilecegimizi göstermektedir.



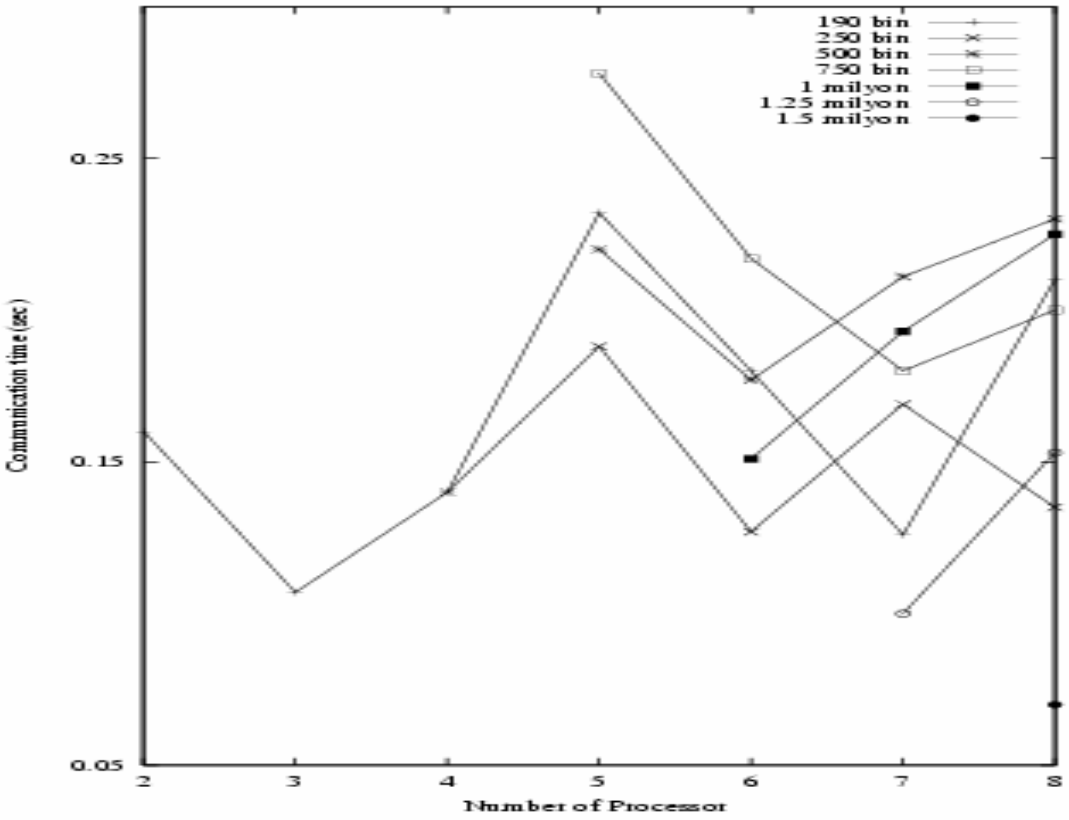


Sekil-3. Farkli islemci sayilari için Islem süresinin atom sayisina bagli degisimi.





Sekil-4. Farkli atom sayilari için islem zamaninin islemci sayisina göre degisimi.



Sekil-5. Farkli atom ve islemci sayilari için, bilgisayarlar arasinda veri transferi süresi.

4. KAYNAKLAR

[1] Daw, M.S. and M.I Baskes,., "Semiemprical Quantum Mechanical Calculation of Hyroggen Embrittlement in Metals ",*Phys. Rev. Lett.*, 50, 1983, pp.1285

[2] Özdogan, C., G. Dereli ve T. Çagin, "O(N) paralel tight binding molecular dynamic simulation of carbon nanotubes", Computer Phys. Comm., 148, 2002, pp188-205.