HPFワークショップ

平成15年9月25日

地球シミュレータセンター

SX-7の特長を生かした粒子シミュレーションHPFコード

核融合科学研究所理論・シミュレーション研究センター

石黒静児

協力いただいた方：

核融合科学研究所

渡邉國彦、大谷寛明、堀内利得、岡本正雄

NEC　林康晴、末広謙二、堀内紳年

中部大　高丸尚教

核融合科学研究所理論・シミュレーション研究センター大型シミュレーション研究用解析装置の概要

- NEC製 SX-7 を中核としたシステム

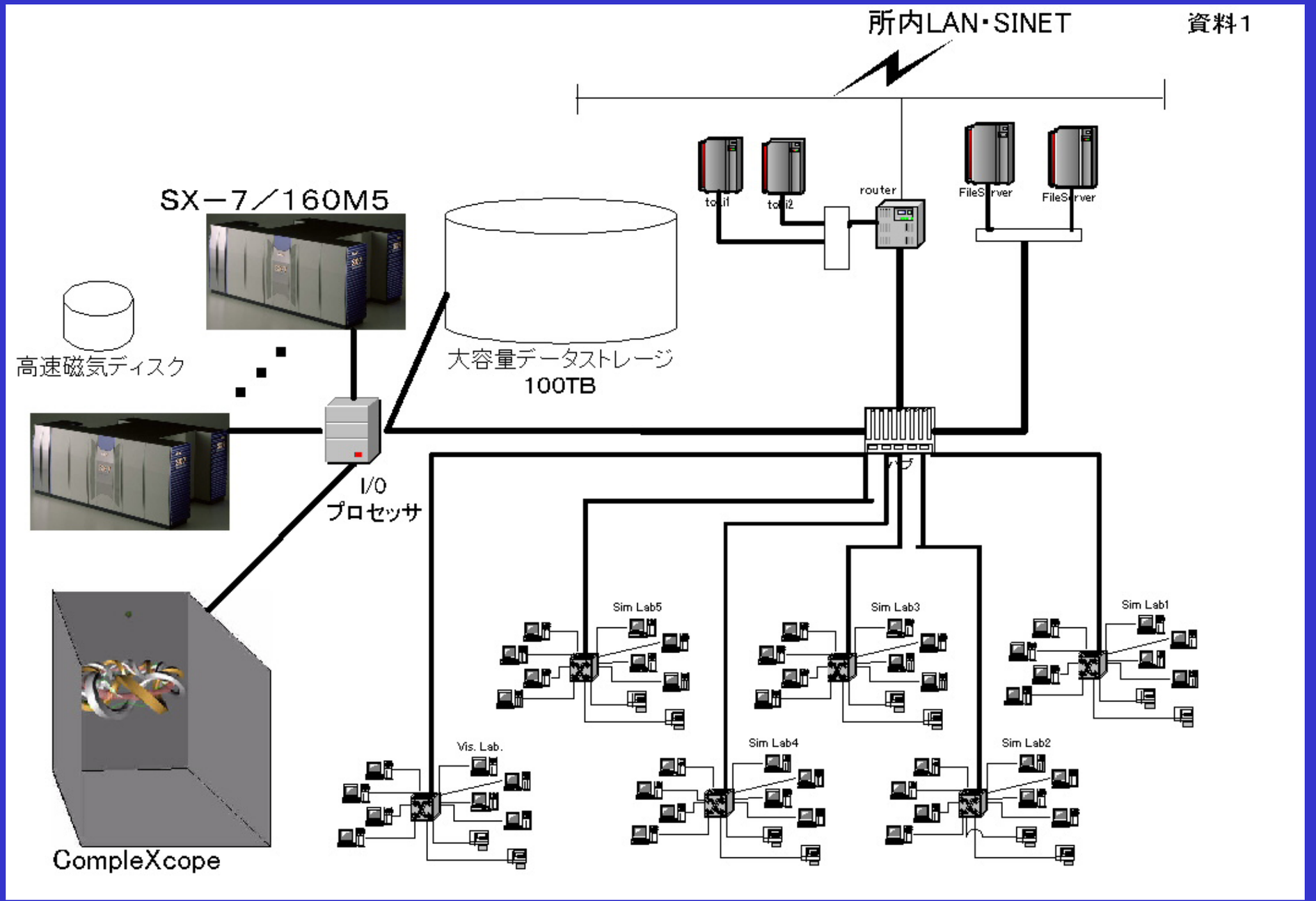
主システム概要

機種名	SX7/160M5
アーキテクチュア	共有メモリ型ベクトル並列
総主記憶容量	1280GBytes
総合処理速度	1440GFlops
ノード数	5
1ノードあたりのPE数	32
1PEあたりのベクトルパイプライン数	4
ノード間転送速度（単方向）	8GBytes/sec
大容量データストレージ	100TBytes

性能（測定値）

- Linpack測定結果
  200,000元の実行で1.378TFLOPS
  (peak性能の97.54%)
  top500 リストの３８位相当

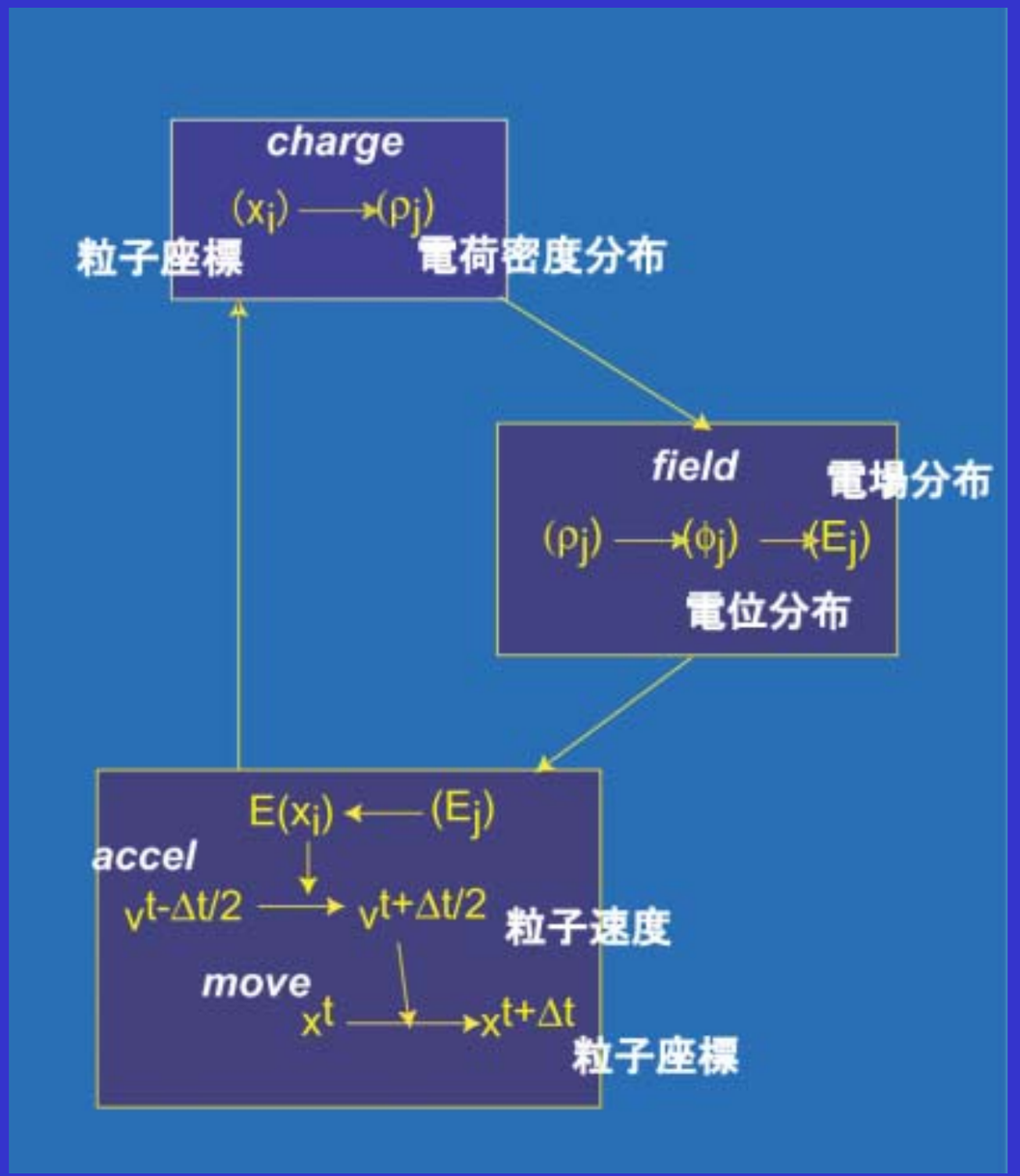
大型シミュレーション研究用解析装置ネットワーク

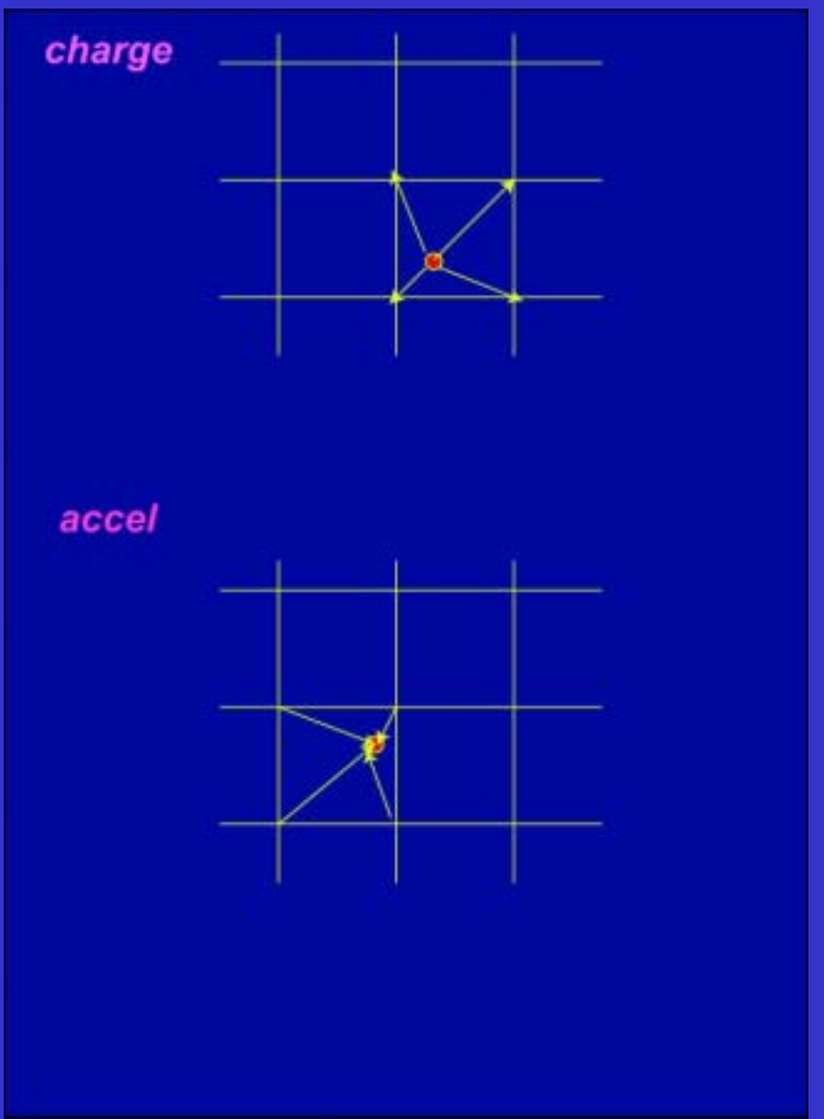


核融合研のシステムの特長

- １ノードあたり主記憶容量が大きい(256GBytes)
- 共有メモリ型ベクトル並列
- 少数のノードで構成（５ノード構成）

プラズマ粒子シミュレーション（静電コード）
Particle-in-Cell(PIC)法





Sxf90 (1node)での実行時間(elapse time)比率

4PEのケース

Charge	48%
Accel	45%
Move	4%
Field	2%

32PEのケース

Charge	75%
Accel	16%
Move	2%
Field	2%

HPF化の方針

- 共有並列とHPFによる分散並列を共用する
- 領域の分割は行わない
- 場の量は各HPFプロセスで重複計算を行う

配列（3次元周期境界静電コード）

- 粒子関係(座標、速度)　--分散

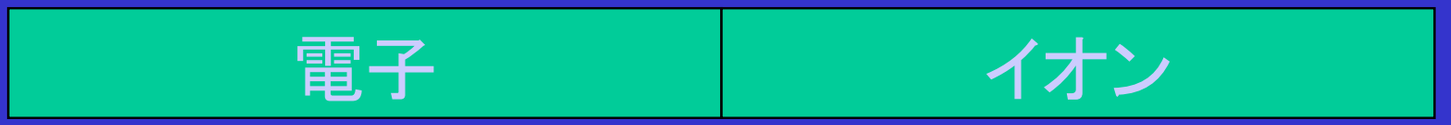
  x(:), y(:), z(:), vx(:), vy(:), vz(:)
- 場の量--分散させない(各HPFプロセスで同じものを持つ)

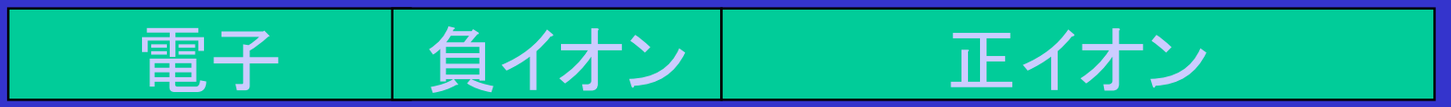
  rho(:,:,:), phi(:,:,:), ex(:,:,:), ey(:,:,:), ez(:,:,:)
- 作業用配列(charge でベクトル、並列実行のため)--分散

  work(:,:,:,:,:)

HPF分散

x,y,z,vx,vy,vz - 通常は1次元配列





要請：x,y,z,vx,vy,vzに各々1つの配列

分散方法（1次元配列）:

1. サイクリック分散

　　現状のsxhpfでは共有並列、ベクトルとの併用がうまくいかない

2. ブロック分散

　　ロードバランスが悪くなる、アルゴリズムの変更大

HPF分散(続き)

- x,y,z,vx,vy,vz　を2次元配列にし、2次元めでブロック分散

1次元目

二次元目 pe, hpf

電子	イオン

プログラム

```
module
      real (kind=kreal),save,dimension(npm,npe)::x,y,z,vx,vy,vz
c-----work arrays
      real (kind=kreal), save,
     &             dimension(0:ngxmx,0:ngymx,0:ngzmx,lr,npe)
     &                                          :: workp
!hpf$ processors p(nhpf)
!hpf$ distribute (*,block) onto p :: x,y,z,vx,vy,vz
!hpf$ align workp(*,*,*,*,i) with x(*,i)
```

charge

```
!hpf$ independent,new(ipe,j,kk,il,iu,n1,nl,lr1,ist,
!hpf$&  jx,jy,jz,ddx,ddy,ddz,nadd)
      do ipe=1,npe
        il = itopp(ipe)
        iu = iendp(ipe)
       -------省略----
!cdir novector
        do j=1,nl
          -------省略----
!cdir nodep
          do kk= 1, lr1
            ii = ist + kk
            jx =x(ii,ipe)
            ddx=x(ii,ipe)-jx
           ----省略 ----
            workp(jx,jy,jz,kk,ipe)
    &             = workp(jx,jy,jz,kk,ipe)
    &              +(1.0d0-ddx)*(1.0d0-ddy)*(1.0d0-ddz)
          enddo
        enddo
      enddo
    end
```

```
!hpf$ independent, new(ipe,l,k,j,i),reduction(rho)
      do ipe = 1,npe
        do l=1,lr,4
          do k=0,ngz
            do j=0,ngy
!cdir nodep
              do i=0,ngx
                rho(i,j,k)=   rho(i,j,k)
    &               + qdxdydz*(workp(i,j,k,l,ipe)
    &               +          workp(i,j,k,l+1,ipe)
    &               +          workp(i,j,k,l+2,ipe)
    &               +          workp(i,j,k,l+3,ipe))
              enddo
            enddo
          enddo
        enddo
      enddo
```

accel

```
!hpf$ independent,new(ipe,i,jx,jy,jz,vxo,vyo,vzo,ddx,ddy,ddz
!hpf$&                ,aax,aay,vzn,vyy,vxx)
      do ipe=1,npe
        do i=itopp(ipe),iendp(ipe)
          jx  = x(i,ipe)
          vxo = vx(i,ipe)
          ddx = x(i,ipe)-jx
          aax =
     &        ex(jx,jy,jz)    *(1.0d0-ddx)*(1.0d0-ddy)*(1.0d0-ddz)+
     &        ex(jx,jy+1,jz)  *(1.0d0-ddx)*ddy*(1.0d0-ddz)+
     &        ex(jx+1,jy,jz)  *ddx*(1.0d0-ddy)*(1.0d0-ddz)+
      ----省略-----
          aay =
     &        ey(jx,jy,jz)    *(1.0d0-ddx)*(1.0d0-ddy)*(1.0d0-ddz)+
          vyy = vy(i,ipe) + aay
          vxx = vx(i,ipe) - t*vyy + aax
          vyy = vyy + s*vxx
          vxx = vxx - t*vyy
          vx(i,ipe) = vxx + aax
          vy(i,ipe) = vyy + aay
          vz(i,ipe) = vzn
        enddo
      enddo
```

性能比較（1ノード及び2ノードジョブ）

128x128x128grids, 64particles/cell for each species,
total number of particles 268,435,456, 500 time steps

Pe数	Node数	Hpfプロセス数	コンパイラ	経過時間 (sec)	sec/step	nsec/step /particle
32	1		sxf90	635	1.27	4.73
32	1	1	sxhpf	646	1.29	4.81
32	1	2	sxhpf	625	1.25	4.66
32	2	2	sxhpf	567	1.13	4.22
32	1	4	sxhpf	612	1.22	4.56
32	2	4	sxhpf	561	1.12	4.18
32	1	8	sxhpf	1016	2.03	7.57
64	2	2	sxhpf	472	0.94	3.52
64	2	4	sxhpf	440	0.88	3.28
64	2	8	sxhpf	432	0.86	3.22
64	2	16	sxhpf	626	1.25	4.66

まとめ

- １ノードあたりの主記憶容量が大きいというSX-7の特長を生かした粒子コードのHPF化を行った。
- 粒子座標等の配列を２次元化し、２次元目をブロック分散することにより効率的に、ベクトル、共有並列、分散並列を併用することが可能なコーディングができた。
- 同じpe数の場合、HPF分散でノード越えを行うことによる性能低下は見られなかった。