FDPS講習会

サンプルコード解説(C++)

野村昴太郎 / Kentaro Nomura 理研RCCS リサーチアソシエイト

2018年8月3日

コードの内容

- ヘッダーのインクルード(#include <particle_simulator.h>)
- 粒子クラスとそのメンバ関数の定義
- 相互作用カーネルの定義
- 時間積分
- I/Oと解析

紹介するサンプルコード

- ディレクトリは\$FDPS_DIR/sample/c++/nbody
- ファイルはnbody.cppとuser-defined.hpp
- ・重力多体問題コード
 - ・時間積分はleap-frog法
 - 初期条件はファイル読み込みではなく、その場で生成

user-defined.hpp全景

```
I/O用ヘッダークラス
  static Mosfé4 eps;
 PS::Pôles. geths?) cost f.
return pos:
                                     粒子クラス
Patrior BANKLE_RAWITON_GRAPS_X86
                                                                                  相互作用カーネル関数
```

I/O用へッダークラス (FileHeaderクラス)

```
2 class FileHeader{
3 public:
4     PS::S64 n_body;
5     PS::F64 time;
6     PS::S32 readAscii(FILE * fp) {
7          fscanf(fp, "%lf\n", &time);
8          fscanf(fp, "%lld\n", &n_body);
9          return n_body;
10     }
11     void writeAscii(FILE* fp) const {
12          fprintf(fp, "%e\n", time);
13          fprintf(fp, "%lld\n", n_body);
14     }
15 };
```

- ParticleSystem::writePraticleAscii/Binary で必要
- ・ 粒子データを単一/分散ファイルに書き込む
- readAsciiとwriteAsciiメンバ関数が必要

粒子クラス

(FPGrav)

```
class FPGrav{
   public:
       PS::S64
20
21
22
23
24
25
26
27
28
29
31
32
33
34
35
36
37
40
41
42
        PS::F64vec pos;
       static PS::F64 eps;
        PS::F64vec getPos() const {
            return pos;
        PS::F64 getCharge() const {
            return mass;
        void copyFromFP(const FPGrav & fp){_
            mass = fp.mass;
            pos = fp.pos;
        void copyFromForce(const FPGrav & force) {
            acc = force.acc;
43
44
45
46
            pot = force.pot;
       void clear() {
47
48
49
50
51
52
53
54
55
56
57
58
59
            acc = 0.0;
            pot = 0.0;
       void writeAscii(FILE* fp) const {
            this->id, this->mass,
                     this->pos.x, this->pos.y, this->pos.z,
                     this->vel.x, this->vel.y, this->vel.z);
       60
                    &this->id, &this->mass,
                   &this->pos.x, &this->pos.y, &this->pos.z, &this->vel.x, &this->vel.y, &this->vel.z);
61
62
63
64
65
```

- メンバ変数に粒子の持つ物理量を記述
- FDPSが粒子の情報を得るのに必要なメンバ 関数を定義(getPos, copyFromFP, copyFromForce, clearは必須)
- I/O用にreadAsciiとwriteAsciiメンバ関数を 定義
- 必要に応じてEssentialParticle(EP)クラス やForceクラスを作ることが可能。本サンプ ルではFPGravが全てを兼ねている

相互作用力一ネル関数

```
void CalcGravity(const FPGrav * ep_i,
124
125
126
127
128
129
130
                           const TParticleJ * ep_j,
          PS::F64 eps2 = FPGrav::eps * FPGrav::eps;
          for(PS::S32 i = 0; i < n_ip; i++){
               PS::F64vec xi = ep_i[i].getPos();
131
132
133
134
135
136
137
138
139
140
141
                for(PS::S32 j = 0; j < n_jp; j++){
                                   r3_inv = rij * rij + eps2;
r_inv = 1.0/sqrt(r3_inv);
                     r3_{inv} = r_{inv} * r_{inv};
                              *= ep_j[j].getCharge();
                              -= r3 inv * rij;
                              -= r_inv;
142
143
144
                force[i].acc += ai;
                force[i].pot += poti;
```

- ・本来は普通の粒子(EP)と超粒子(SP) 用にそれぞれカーネル関数が必要だ が、本サンプルではテンプレート関 数化することで同じ関数を利用
- 引数は固定(EPI*, N_{EPI},EPJ*/SPJ*, N_{EPJ/SPJ}, Force*)
- getPos()メンバ関数で粒子の座標を 取得
- Force(ここではFPGrav)の配列に計算結果を格納

nbody.cpp全景

```
entitle distribution distribution
   ヘッダーのインクルード
     Machade "exer-defined.http"
accent(eng < 0.0);
                                      PS::HTTS mt:
nt:Hait_conrand(0);
name:PS::Est b = M; b = M_mart; last;
massill = mass_pto / m_stb;
const PS::RS4 redias + 3.6;
de d
                                                           PS::F64rec on pop = 6.bt
PS::F64rec on vol = 6.bt
PS::F64rec on vol = 6.bt
PS::P64:S22 | = 6: 1 = 1 test | == 10
cm_pop - + mass[s] + pop[s];
cm_pop + + mass[s] + pop[s];
cm_pop + + mass[s] + pop[s];
                   complatercians Topyon
unid patharticlescoldUniforniphers(Tocyc & pays,
read: myllycoln.gom,
Pic:550 & m locb &
                      %_lac = a_glb;_
pays.setNumberO(PartScleLocal(s_lac);
                   PS::P840 + mace - man PS::F54(alas);
PS::P84000 + mac - man PS::F54000 | Loc!;
PS::F64000 + v41 - man PS::F5400 | alas];
casca F6::F64 mate - 1.3;
c
                                                                                                                                                                                                                                                                時間積分
```

```
continues a state of the continues of th
                                                                                                                                                                          AUTHORN BURGINGSTON
                                                                                                                                                                                                                                                                                                                    時間積分
 reid calcomeration (project parts & system. 
PS:: FSA 5 corp. 
PS:: FSA 5 corp. 
PS:: FSA 5 corp. 
PS:: FSA 5 corp. 
Const bool Clear-true] (
                        if(clear){
   etat = axis = ass: = 0.0:
                       vaid printletp() {
    clotterveror ter see at output tontmore. //posit/freetottendig
    stotterveror ter see a defeat: 0.5)*-estditendig
    stotterveror time_ter (defaat: 0.5)*-estditendig
    clotterveror time_ter (defaat: 0.0)*-estditendig
    clotterveror time_ter (defaat: 1.0 / 120 m)*-estditendig
    clotterveror time_ter (defaat: 1.0 / 120 m)*-estditendig
    clotterveror time_ter (defaat: 1.0)*-estditendig
    clotterveror time_ter (defaat: 1.0)*-estditendig

vaid makeOutputDirectory(char + dir_name) {
    pints( stat st:
    PtintS( ret;
    if (PtintSomerapethank)) -- 0) {
        if instantial name, bot (= 0) {
            ret = mkdEr(dir_name, 0777);
        } else {
            ret = 0; // the directory names dir_name already evists.
            ret = 0; // the directory names dir_name already evists.
            ret = 0; // the directory names dir_name already evists.
            ret = 0; // the directory names dir_name already evists.
                           PSTICOMETHN/GAGESFLS/ref, 111
                       PS : FC+ FFGrevicepe + 1, 3/32 6
                                                                                                                                                                                                                                         main関数
    isr sqisiint angr, car sarge[]] {
    itd::cout--atd::astprecision(15);
    itd::corr--atd::stprecision(15);
                     P5::Initiatizefargt. argv1:
P6::F22 thata + b.5;
P5::F22 n_maxr_limar = 2;
P5::522 n group limit = 64:
P5::F22 time_end = 18.6;
P5::F22 dr = 1.8 / 122 8;
                         P5::F32 dt diag = 1.8 / 8.6:
P5::F32 dt amap = 1.8:
                                          :502 c;
intf(dir_mane,"./recult"]]
                         sprintffdir name.opiarpl:
```

```
stdingers on "theta of on theta on stdingerly Mean"
                                                                                                                       time_end = stofigstorg);

time_end = stofigstorg);

time_end = restricted = restricted = storiced;

storicted = restricted = restricted = storiced = stori
                                                                                                              specific analysis of the state of the side conditions of the state of the side conditions of the state of the side conditions of the side
                                                                                                                                                               dt diag = atof(optarg):
std::corr -x "dt_diag = " ex dt_diag -x std::gmdl;
areak)
                                                                                                              figh 'D':
    st_map + stof(optorg);
    std::onr += "et st_cmap += std::ondlp
west
                                                                                                           case "L":

| Leaf_limit + staf(optary);
| STATE | Continue | Conti
                                                                                                                                                               a_greep_itent = arenteprarate;
std::corr =v "n group limit = " =v n group limit =v std::cond)?
                                                                                                                                                               n_tot = atol(cotamp):
std::corr == "a_tot = " == a_tot == atd::condl;
areak)
                                                                                                        case 'h':
   Ef(PS::Conn::getRank() -- b) {
    printHelp();
                                                                                                                                                               PSc: Finalize();
return b;
                                                                                                                                                         If(P5::(com::getRank() == b) {
   ctd::der-<*No cuch aption! Available options are here.*c
   printe(p1);</pre>
                                                                                                                                                                  PSc:Aborof1;
                                                                  makeButautBirectory(dir_mame);
                                                                                                              The sort do[1804] to the control of 
                                                       province recommendation operation operation;
province province in the commendation operation operatio
                                                                     COURT PREFETT COST_680 = 8.35
                                                            PS::DomainEnfo dinfo;
dinfo.Entitaliza(conf.end);
dinfo.Entitaliza(conf.end);
dinfo.Entitaliza(conf.end);
dystem_grav.exthemperfortEcte(dinfo);
n_loc = system_grav.getHumberOfParticLetecal[]]
                 ASSOCIATION CHAPE NOS
                 g5_open();
g5_set_eps_ra_all(F9(squrreps);
#cmilf
```

```
tree_grav.caleForce*!!!!deriteback|CaleCroubby-EPGrav-,
                    PS:: F44 Epoid: Ehind. Elabb. Epoil: Ehinl. Etabl:
PS:: F44 Epoid: Ehind. Elabb. Epoil: Ehinl. Etabl:
p:: F44 Inc. Ening = 2.8;
PS:: F44 Inc. Ening = 2.8;
P
                                          caleinergy(cyctom_graw, Eteti, Ekini, Spoti);
                                         kick(system_grav, dt + 8.5);
                                         rime_sys as dry
drift(system_grav, dt);
'(n_loop & 4 mm Mp(
| dinto.decomposadonminAllisystem_provid
                                          tree_prov.catcPorceAttAndWriteBook(CatcDrawity=PPGrave.
CatcDrawity=PGravExecute
                                         kickinyatem_grav. dt # 0.51:
                                          a_loopees
         FEFFIN SELECT PRINTEN_GAMPS_KSS
        Pondid
                       PSTIPMBHITHING
```

FDPSヘッダーファイルの インクルード

```
#include<iostream>
#include<fstream>
#include<unistd.h>
#include<sys/stat.h>

#include<sparticle_simulator.hpp>
#ifdef_ENABLE_PHANTOM_GRAPE_X86
#include <gp5util.h>
#endif
#ifdef_ENABLE_GPU_CUDA
#define_MULTI_WALK
#include"force_gpu_cuda.hpp"
#endif
#include "user-defined.hpp"
```

- particle_simulator.hppをインク ルード
- 粒子クラス等の定義が別ファイル (user-defined.hpp)の場合はイン クルード

時間積分関数の定義

- Leap-frog法の速度と座標の更 新の関数を定義
- ParticleSystem::getNumberOf ParticleLocal()関数でプロセス が担当する粒子の数を取得
- []演算子で粒子データにアクセス

エネルギーの計算(総和など)

```
void calcEnergy(const Tpsys & system,
103
104
105
106
107
                          PS::F64 & etot,
                          PS::F64 & ekin,
                          PS::F64 & epot,
                          const bool clear=true){
          if(clear){
108
109
110
               etot = ekin = epot = 0.0;
          PS::F64 etot_loc = 0.0;
PS::F64 ekin_loc = 0.0;
112
          PS::F64 epot_loc = 0.0;
          const PS::S32 nbody = system.getNumberOfParticleLocal();
113
          for(PS::S32 i = 0; i < nbody; i++){}
114
               ekin_loc += system[i].mass * system[i].vel * system[i].vel;
epot_loc += system[i].mass * (system[i].pot + system[i].mass / FPGrav::eps);
115
116
117
          ekin_loc *= 0.5;
          epot_loc *= 0.5;
120
121
122
123
124
          etot_
                   PS::Comm::getSum(etot_loc);
                  PS::Comm::getSum(epot_loc);
```

- ポテンシャルや運動エネルギーを 計算する場合、プロセス間の総和 が必要
- PS::Comm::getSum()関数でプロセス間の総和をとることが可能
- MPIが使われない場合でもコード の変更が不要

main関数全景

```
nt main(int argc, char *argv[]) {
    std::cout<<std::setprecision(15);</pre>
                                                                                                                                                               std::ofstream fout_eng;
                                                                                                                                                                                                                                                                                                                                                                                                   _* << (Etot1 - Etot0) / Etot0 <
                                                                                                                                                              if(PS::Comm::getRank() == 0) {
    char sout_de[1024];
                 std::cerr<<std::setprecision(15);
                                                                                             初期化
                                                                                                                                              237
238
                                                                                                                                                                        sprintf(sout_de, "%s/t-de.dat", dir_name);
               Craft United Lizer and Crady (V)
                                                                                                                                                            fout_eng.open(sout_de);
fprintf(stdout, "This is a sample program of N-bod')
fprintf(stdout, "Number of processes! %d\n", P$::G')
fprintf(stdout, "Number of threads per process: %d')
ParticleSystem初期化(初期条件生成)
               F3::132 theta = 0.5;
P5::532 n_leaf_limit = 8;
P5::532 n_group_limit = 64;
P5::F32 time_end = 10.0;
                                                                                                                                                                                                                                                                                              369
318
                                                                                                                                                                                                                                                                                                                    時間発展①
                                                                                                                                                                                                                                                                                              311
312
                                                                                                                                                                                                                                                                                                                        klck(system_grav, dt * 0.5);
                PS::F32 dt = 1.0 / 128.0;
               PS::F32 dt_diag = 1.0 / 8.0;
PS::F32 dt_snap = 1.0;
thar dir_name[1024];
PS::S64 n_tot = 1024; 実行時オプション処理
                                                                                                                                                                                                                                                                                              314
315
316
                                                                                                                                                                                                                                                                                                                       time_sys += dt;
                                                                                                                                                             PS::ParticleSystem<FPGrav> system_grav:
system_grav.initialize();
PS::S32 n_loc = 0;
PS::F32 time_sys = 0.0;
if(PS::Comm::getRank() == 0) {
    setParticlesColdUniformSphere(system_grav, n_lot,
173
174
                                                                                                                                              246
247
248
                                                                                                                                                                                                                                                                                               317
318
                                                                                                                                                                                                                                                                                                                       if(n_loop % 4 == 0){
    dinfo.decomposeDomainAll(system_grav);
                sprintf(dir_name,"./result");
                                                                                                                                              249
258
251
                 opterr = 0;
                                                                                                                                                                                                                                                                                              320
321
                while((c=getopt(argc,argv,"i:o:d:D:t:T:l:n:N:hs:")) !=
    switch(c){
                                                                                                                                                                                                                                                                                                       system_grav.exchangeParticle(dinfo);
#ifdef MULTI_WALK
                                                                                                                                                                      system_grav.setNumberOfParticleLocal(n_loc);
                                                                                                                                              252
252
254
                                                                                                                                                                                                                                                                                              323
224
325
376
327
                                                                                                                                                                                                                                                                                                                        tree_grav.calcForceAllAndWriteBackMultiWalk(DispatchKernelWithSP,
                         case 'o':
                                  sprintf(dir_name, optarg);
                                                                                                                                                                                                                                                                                                                                                                                                                    RetrieveKernel,
                                                                                                                                                             const PS::F32 coef_ema = DomainInfo初期化
pS::DomainInfo dinfo;
dinfo.initialize(coef_ema);
dinfo.decomposeDomainAll(system_grav);
system_grav.exchangeParticle(dinfo);
183
                                break:
                                                                                                                                                                                                                                                                                                                                                                                                                    tag_max,
184
185
                                                                                                                                              256
257
258
259
                                                                                                                                                                                                                                                                                                                                                                                                                    system_grav,
                         case 't's
                                                                                                                                                                                                                                                                                                                                                                                                                    dinfo,
n_walk_limit,
                                 theta = atof(optarg);
                                                                                                                                                                                                                                                                                              328
329
330
331
                                 std::cerr << "theta =" << theta << std::endl;
                                                                                                                                                                                                                                                                                                                       相互作用計算
                                                                                                                                                                                                                                                                                                                                                                                                                    true!;
                                 break;
                                                                                                                                                              n_loc = system_grav.getNumberOfParticleLocal();
                                                                                                                                                                                                                                                                                                       #else
                         case 'T':
                                 rime_end = atof(optarg);
                                                                                                                                                                                                                                                                                                                       332
333
                                 std::cerr << "time_end = " << time_end << std:
198
                                                                                                                                                      #ifdef ENABLE_PHANTOM_GRAPE_X86
                                                                                                                                                     g5_upen();
g5_set_eps_to_all(FPGrav::eps);
g5_set_eps_to_all(FPGrav::eps);
                                 break;
                                                                                                                                                                                                                                                                                                                                                                                                 system_grav,
                         case 's':
                                                                                                                                                                                                                                                                                              334
                                 dt = atof(optarg);
std::cerr << "time_step = " << ct << std::end"</pre>
193
194
                                                                                                                                                                                                                                                                                                       #endif
                                                                                                                                                                                                                                                                                                                        kick(system_grav, nt * 0.5);
                                                                                                                                                              tree_grav.initialize(n_tot, theta, n_leaf_limit, n_gro
                                                                                                                                                                                                                                                                                                                                                                                               時間発展②
196
                                                                                                                                                                                                                                                                                              338
339
348
                         case 'd':
                                                                                                                                                            fdef MULTI_WALK
const P5::S32 n_walk_limit = 200;
                                 dt_diag = atof(optarg):
std::cerr << "dt_diag = " << dt_diag << std::e
                                                                                                                                                                                                                                                                                                                        n_loop++:
                                                                                                                                                              const PS::S32 tag max = 1;
tree_grav.calcForceAllAndWriteBackMultiWalk(DispatchKe)
                        break;
                                                                                                                                                                                                                                                                                                       #Ifdef ENABLE_PHANTOM_GRAPE_X86
                                                                                                                                                                                                                                                                                                      g5_close();
#endif
                                 dt_snap = atof(optarg);
std::cerr << "dt_snap = " << dt_snap << std::e
281
282
283
264
285
286
287
288
289
                                                                                                                                             272
273
274
                                                                                                                                                                                                                                                           RetrieveKe
                                                                                                                                                                                                                                                           τag_max,
                                                                                                                                                                                                                                                           system_gra
                                 break;
                                                                                                                                                                                                                                                           dinfo,
n_walk_lim
                                                                                                                                                                                                                                                                                                          PS:: Finalize(); 終了処理
                         case 'l':
                                                                                                                                                                                                                                                                                              345
347
348
                                 n_leaf_limit = atoi(optarg);
std::cerr << "n_leaf_limit = " << n_leaf_limit";</pre>
                                                                                                                                                              tree_grav.calcForceAllAndWriteBack(CalcGravity<FPGrav>
                                break;
                                                                                                                                                                                                                                        CalcGravitycPS::SP3
                         case 'n's
                                 n_group_limit = atoi(optarg);
std::cerr << "n_group_limit = " << n_group_limit = " << n_
                                                                                                                                                                                                                                         system_grav,
210
211
                                                                                                                                                                                                                                        dinfo);
                                                                                                                                                             break;
                                 n_tot = ato1(optarg);
214
                                 std::cerr << "n_tot = " << r_tot << std::endl;
                                                                                                                                              285
286
287
200
289
290
291
292
293
                                 break;
216
                         case 'h':
                                 Lf(PS::Comm::getRank() == 0) {
    printHelp();
217
218
228
221
222
                                 PS::Finalize();
                                  return 0:
                                                                                                                                                                               header.time = time_sys;
header.n_body = system_grav.getNumberOfParticl>
system_grav.writeParticleAscii(filename, heade>
223
224
                                 Lf(PS::Comm::getRank() == 0) {
    std::cerr<<'No such option! Available opt:
    printHelp();</pre>
                                                                                                                                              294
295
296
297
298
299
388
381
382
                                                                                                                                                                                time_snap += dt_snap;
                                 PS::Abort();
                                                                                                                                                                       calcEnergy(system_grav, Etot1, Ekim1, Epot1);
230
231
                nakeOutputDirectory(dir_name);
                                                                                                                                                                        if(PS::Comm::getRank() == 0){
                                                                                                                                                                                if( (time_sys >= time_clag) || ( (time_sys + d )
```

FDPSの初期化と終了処理

```
int main(int argc, char *argv[]) {
         std::cout<<std::setprecision(15);</pre>
163
164
         std::cerr<<std::setprecision(15);</pre>
165
        PS::Initialize(argc, argv)
166
        PS::F32 theta = 0.5;
167
        PS::S32 n_leaf_limit = 8;
168
        PS::S32 n_group_limit = 64;
169
        PS::F32 time_end = 10.0;
170
        PS::F32 dt = 1.0 / 128.0;
171
        PS::F32 dt_diag = 1.0 / 8.0;
172
        PS::F32 dt_snap = 1.0;
173
        char dir_name[1024];
174
        PS::S64 n_{tot} = 1024;
```

必ず最初にPS::Initialize()と最後に PS::Finalize()関数を呼ぶ

```
PS::Finalize();
return υ;
348 }
```

FDPS各クラスの初期化

```
PS::ParticleSystem<FPGrav> system_grav;
        system_grav.initialize();
        PS::S32 n_loc = 0;
246
        PS::F32 time sys = 0.0;
        if(PS::Comm::getRank() == 0) {
            setParticlesColdUniformSphere(system_grav, n_tot, n_loc);
250
        } else {
251
            system_grav.setNumberOfParticleLocal(n_loc);
254
        const PS::F32 coef_ema = 0.3;
       PS::DomainInfo dinfo;
256
       dinfo.initialize(coef_ema);
       dinfo.decomposeDomainAll(system_grav);
        system_grav.exchangeParticle(dinfo);
259
       n_loc = system_grav.getNumberOfParticleLocal();
    #ifdef ENABLE_PHANTOM_GRAPE_X86
        q5_open();
       g5_set_eps_to_all(FPGrav::eps);
        PS::TreeForForceLong<FPGrav, FPGrav>::Monopole tree_grav;
        tree_grav.initialize(n_tot, theta, n_leaf_limit, n_group_limit);
```

- 各クラスでは必ずInitialize()メンバ関数を呼ぶ必要がある
- ParticleSystemクラスは初期化後に 初期条件の生成/読み込みを行う
- DomainInfoクラスは初期化後に最初 の領域分割を行う (decomposeDomainAll)
- TreeForForceクラスは相互作用計算 の形式によって呼び出すクラスが変わる(今回はTree法を使ったモノポール の長距離相互作用計算を行うもの)

相互作用の計算

- TreeForForce::calcForceAllAndWriteBackメンバ関数を呼ぶ
- ・ 引数に普通の粒子の相互作用計算関数と超粒子の相互作用計算関数(本サンプルではテンプレート関数で両方に同じ関数を使用), ParticleSystem, DomainInfoを入れる
- その他オプションがあるがここでは割愛

まとめ

- 500行程度でI/Oまでを含むTree法を用いた重力多体問題シミュレーションコードを記述
- OpenMPやMPIの並列化を意識するところ(ほぼ)ない
- コンパイルオプションを変更することで、MPIやOpenMPを利用 可能