

# 共有 Xeon マシン (RYUDO、COLORS 等用) 利用説明書

Ver.1.06a(2005/10/21)  
(計算プログラム事前インストール/計算時転送共通)

---

## 1. 概要

現在、共有 Xeon マシンには MaterialsStudio 用と COLORS 大規模計算用、およびその他 (RYUDO、COLORS 等) 用の 3 種類があります。この説明書で解説するのはその他用です (COLORS 大規模計算用も含む)。

- いずれも `akinux` 名が付いていますが、個人用ではありません。
- 管理者は磯本さんですが、三浦も支援しています。
- MaterialsStudio 用については別の説明書を参照してください。
- COLORS 大規模計算用の使用方法是この説明書と同じですが、`outer` を別名にして (4. 1 節参照) その他用とは区別するようにしてください。

Xeon マシンのスペックは下記の通り (マシンにより若干異なります)。

CPU : Pentium4 Xeon 2.8GHz 2CPU (HT Disable)  
RAM : DDR-SDRAM 4GB  
OS : RedHat Linux 8.0 (kernel:2.4.20-18.8smp)

実行する計算プログラムとしては、宮本研オリジナルのものを推奨します。

- RYUDO、および COLORS の動作実績があります。
- その他、ライセンスを必要としないプログラムなら、ほとんど動作すると思います。(当然 Linux で動作するものに限られますが。)
- ADF 等、ライセンスを必要とするプログラムについては、別途ご相談ください。  
#MXDORTO, MSPORT, MSP, TRIBOSIM は問題なく動作すると思います。  
#ADF も動作しますが、プログラムとライセンスのインストール作業が必要です。

計算を入れる (投入する) には、`outer` コマンドを使用してください。

- `telnet` などでログインして直接投入しないでください。  
(他の方の計算を圧迫します。)
- `outer` コマンドは三浦が開発したオリジナルプログラムです。  
`outer` コマンドを使用するには、クライアント (親機) となる `akinux` が必要です。  
(通常、各個人に割り当てられている `akinux` を使います。)
- #共有 Xeon マシンはサーバ (子機) と呼びます。

初めて共有 Xeon マシンを利用する方は、まずは次章以降の手順に従って、自分の `akinux` と、Xeon マシン (の自分のアカウント) の両方を、それぞれ設定してください。

新しく Xeon マシンが増えた時は、以前の Xeon マシンと同様に設定するほか、自分の `akinux` の `.outerc` にマシン名を追加してください（詳細後述）。

2005/10/20 現在、その他用共有 Xeon マシンは下記のとおりです。

`akinux262`～`271`

`akinux307`～`326`      (ただし `akinux309,314` は故障中)

`akinux342`～`351`

`akinux372`～`381`      (ただし `akinux372,380,381` は故障中)

参考：COLORS 大規模計算用としては `akinux394`～`403` の 10 台があります。

---

## 2. 設定方法

### 2. 1. クライアント側（親機、自分の `akinux`）の設定

以下は、自分のアカウント（一般ユーザ）で作業してください。

注：クライアント側にするマシンは必ず 1 台だけにしてください。複数のクライアントマシンを設定して使った場合、計算管理番号がバッティングして動作に異常を来す可能性が高いです。ご注意ください。（どうしても複数のクライアント機を使いたい場合は、`outer` の名前を `outer2` などに変更してご使用ください。詳細は後述）

#### 2. 1. 1. `outer` コマンドをインストールしてください

パスの通った場所（例えば `~/bin/` など）に、`outer` プログラム、`showr` プログラム、および、`upout` プログラムを置いてください。

`~/bin/`の中には、プログラムソースの入ったディレクトリではなく、プログラム本体を直接置くようにしてください。さもないと、パスが通らず不便です。

以下の説明はすべて `outer`、`showr` および `upout` プログラムにパスが通っていることを前提とします。もしパスを通さない場合は、各自の責任で適宜読み替えてください。

各プログラムのソースは、`akinux20` の `/home/lab/outers/`の中にあります。FTP など、`lab` アカウントでログインし、ソースを持ってきてコンパイルしてください。

`lab` アカウントでログインすると、ソースは `outers` ディレクトリの中です。

`outer` のソースは `outer.c`、`showr` のソースは `showr.c`、`upout` のソースは `upout.c` です。コンパイルするには `gcc outer.c -o outer` などとしてください。

#### 2. 1. 2. `.outerc` ファイルを作成してください

`vi` や `emacs` などで、下記のような内容の `.outerc` ファイルを、ホームディレクトリ (`~`) に作成してください。（ファイル名の頭にドット「`.`」を忘れないように）

akinix262	2
akinix263	2
akinix264	2
akinix265	2

(akinix262～265 までを使う場合。このあと必要なだけ追加してください。)

ここで各行の頭に空白はいりません。第1項がホスト名、第2項がCPU数です。ちなみにすべてのXeonマシンは2CPUなので、CPU数は必ず2にしてください。

おまけ：viの使い方 (viはLinux、UNIXなら必ず入っているので便利です)

上述の.outercファイルを作る場合

1. コマンドラインで「vi .outerc」と入力 (ホームディレクトリで)
2. viが起動する。入力モードにするため「i」キーを1回押す。
3. 上述の「akinix262・・・」をそのまま入力する。  
行の終わりはEnterキー。打ち間違えたらBack spaceキーで消す。
4. 入力が済んだら、入力モードを抜けるためEscキーを1回押す。
5. 保存・終了するため、まずコマンドモードに入る。「:」キーを1回押す。
6. 書き込み&終了の意味の「wq」と入力。Enterキーで決定。
7. ファイルが保存され、viが終了する。

(既存のファイルを修正する場合も、上記の使用方法に準じる。)

注：ファイル名がドット「.」で始まるファイルは、Linux(UNIX)では隠しファイルとなります。通常の「ls」コマンドでは見えませんのでご注意ください。確認する場合は、「ls」コマンドに「-a」オプションを付けてください (例：ls -a)。なお、隠しファイルであっても、moreや編集は普通のファイルと同様にできます。

## 2. 1. 3. .outerlogファイルを作成してください

viやemacsなどで、下記内容の.outercファイルをホームディレクトリ (~/) に作成してください。(ファイル名の頭にドット「.」を忘れないように)

```
0 outerlog
```

(この1行だけです。「0」の前に空白を入れないでください。)

クライアント側の設定は以上です。

## 2. 2. サーバ側 (子機、共有Xeonマシン) の設定

telnetなどでサーバ (共有Xeonマシン) にログインし、以下の作業をしてください。ログインアカウントは他のakinixと同様です (パスワードは初期値になっている可能性があります。必要に応じて変更してください)。

注意：下記の作業は上記.outercファイルに書いた全てのサーバに対して必要です。

## 2. 2. 1. .rhosts ファイルを作成してください

vi や emacs など、下記のような内容の.rhosts ファイルをホームディレクトリ (~/) に作成してください。(ファイル名の頭にドット「.」を忘れないように)

```
akinux20 miura
```

(akinux20 がクライアントで、miura がログインする場合)

(万一、複数の akinux から計算を投入する場合は、それぞれ行を分けて、複数行庭建て記述してください。)

.rhosts ファイルは、作成しただけでは有効になりません。下記の chmod コマンドを実行して、自分以外は不可視にしてください(セキュリティー対策のため)。

```
chmod 600 .rhosts
```

(属性の確認は、「ls -al」コマンドで可能です。.rhosts が-rw----- になっていることが必要です。)

## 2. 2. 2. bin ディレクトリを作成してください

ホームディレクトリ(~/)の中に bin ディレクトリを作成してください。ryudo などの計算プログラムを置くためです。

なお、もし後述の「計算を投入する際に計算プログラムも同時に投入する」方法を使う場合、以下の作業(bin ディレクトリの作成、および計算プログラムの送り込み)は不要になります。(ただし、後述の showr での計算途中結果の確認ができなくなります)

ディレクトリを作成するには、mkdir コマンドを使用してください。

例     mkdir bin

## 2. 2. 3. 計算プログラム等を bin ディレクトリに置いてください

ryudo や colors などの実行形式ファイルを、bin ディレクトリの中に置いてください。各サーバごと、個別に FTP など転送しても良いですし、あるいは、すべてのサーバで上述の設定をすべて終えてから、upout プログラムを利用しても良いです。

upout プログラムを利用して、一斉アップロード(すべてのサーバにファイル転送)を行う場合は、まずすべてのサーバで、上記の設定(.rhosts ファイル、および bin ディレクトリ)を完了してください。

ついで、クライアント(各自の akinux)で、ryudo など実行形式の置いてあるディレクトリに移動し、upout ryudo (ryudo プログラムを送り込む場合)などとして upout を実行してください。

(upout プログラムの使用方法の詳細は後述します。)

showr プログラムで計算の途中経過を見たい人は、同様に各サーバの bin ディレクトリ

に `ryuga` プログラムを置いてください。

## 2. 2. 4. bin ディレクトリにパスを通してください

上記までの作業を行った後、`rsh` コマンドなどでサーバ機にログインして `ryudo` を実行できれば、特に設定は必要ありません。

(このチェックで `ryudo` 用のファイルを用意する必要はありません。ファイルがない場合、`ryudo` は情報を表示、`cnd.rd` を出力して止まります。`cnd.rd` は消して置いてください。)

もし、`ryudo` が起動されず、`command not found` や `no suchfile or directory` などと表示された場合、`bin` ディレクトリにパスが通っていないことになります。

この場合、`.bashrc` ファイルを `vi` や `emacs` など編集し、下記の一行を追加してください。

```
export PATH=$HOME/bin:$PATH
```

その後、ログインし直せば、パスが通っているはずです。再度 `ryudo` 起動テストを試みてください。それでもなお `ryudo` が起動しない場合は、`.bashrc` を確認し直してから、管理者までご相談ください。

サーバ側の設定は以上です。ログアウト (`exit` コマンド) してください。

---

## 3. 日常の利用方法

日常の利用では、サーバの直接操作は不要です。サーバへログインする必要はありません。通常はクライアント側 (親機、各自の `akinux`) から `outer` プログラム、`showr` プログラム、`upout` プログラムですべてできます。

### 3. 1. サーバの空き状況をチェック

サーバの空き状況を調べるには、下記の手順に従ってください。

(注: あらかじめ空き状況を調べなくとも、計算投入の手続き (次節で説明) を始めることは出来ます。空きがなかったときは自動的に待機状態になるからです。)

クライアント側 (自分の `akinux`) で `outer` とだけ入力して実行してください。オプションや引数は付けなくてください。

`outer` プログラムが計算サーバを巡回し、結果を返してきます。`.outerc` に登録したサーバが多い場合、少々時間がかかりますので、しばらくお待ちください。

以下、表示例を示します。

```
Outer version 1.17 , 2005/6/18 by Ryuji Miura from 2003/6/12.
Host      Average User      Priority      Load          Process
akinux262 1.00   miura        19           99.6000      ryudo
```

```

akinux262  1.00  shimizu    19          99.6000    ryudo
akinux263  1.00  miura     19          99.5000    ryudo2
akinux264  0.00
akinux265  0.00
Loads
akinux262  s 2/2
akinux263  s 1/2
akinux264  0/2
akinux265  0/2
free cpu = 5

```

最初の行で、outer のバージョン番号などを示します。

- ・常に最新バージョンを使うように心がけてください。

Host 以下に、各サーバの状況を示します。

Host の下はサーバ名

Average の下は負荷数 (=計算数)

User の下はその計算を入れた人のアカウント名

Priority はその計算の優先度 (通常 19 以上、または 39 以上)

Load はその計算の負荷率

Process はその計算のプログラム名

(表示される計算は負荷率 25% 以上のものだけです)

Loads 以下に、各マシンの有線フラグ「s」および「計算数/CPU数」を列挙します。  
free cpu は空き CPU 数です。

もし空き CPU が無くても、下記の「計算の投入」作業は可能です。この場合、空き CPU が出るまで定期的にサーバをチェックし、待ちつづけます。

注：Host 表示以下における各サーバチェック時に下記の表示が出ることがあります。

No route to host      このサーバは停止しています (故障しています)

Permission denied.    このサーバはログインできません (サーバ名は次行)

No route to host の場合、管理者しか修復できませんので、ご了承ください。

Permission denied の場合、下記を確認して修正してください。

.rhosts ファイルを作成したか？。chmod 600 .rhosts を実行したか？

(rsh akinuxXXX として、パスワードなしでログインできれば OK です)

なお、問題のあったサーバは Loads 行以下の一覧に表示されません。

### 3. 2. サーバに計算を投入

サーバに計算を投入するには、以下の手順に従ってください。

- 1) カレントディレクトリに計算開始に必要なファイルのみを置いてください。
  - ・計算に関係ないファイルは置かないでください。
  - ・サブディレクトリは置かないでください。
  - ・計算投入と同時に計算プログラム本体も投入する場合は、計算ファイルと一緒にプログラム本体もカレントディレクトリに置いてください。

例えば RYUDO 計算の場合、def.rd と set.rd（場合によっては bondon.dat など）をカレントディレクトリに置いてください。計算プログラム本体も投入する場合は、ryudo も一緒に置いてください。

## 2) outer プログラムに計算プログラム名を付けて実行してください。

上記で用意したカレントディレクトリに於いて、outer プログラムを実行します。このとき、outer の後に空白を開けて計算プログラム名を付けてください。例えば RYUDO 計算の場合、

```
outer ryudo
```

などとしてください。

- 計算プログラム名は、サーバの bin ディレクトリに置いたファイル名にしてください（上述 2. 3. 3 参照）。たとえば「ryudo106」のようにバージョン名を付けたりした場合、「outer ryudo106」などのように実行してください。
- 計算プログラムの同時投入を行う場合、カレントディレクトリのプログラム名にアタマに「./」を付けてください。たとえばカレントディレクトリの「ryudo」を使う場合は、「outer ./ryudo」のように実行してください。

注：「nice: ryudo106g: No such file or directory」などと表示されて outer が止まってしまった場合、指定した計算プログラム（この場合は ryudo106g）にパスが通っていない可能性があります。サーバの bin ディレクトリにプログラムを置いてある場合は、サーバのホームディレクトリの .bashrc ファイルに「export PATH=\$HOME/bin:\$PATH」の行があるか確認し、無ければ追加してください。計算プログラムを同時に投入する場合は、カレントディレクトリに計算プログラムが置いてあるか、outer 実行時に計算プログラム名の前に「./」を付けてあるか確認してください。

あとは outer が自動的に空いている CPU を探し出して計算を投入します。計算が完了し、計算結果がクライアント側（自分の akinux）に戻ってくるまで、お待ちください（放置可）。

もし CPU が空いていない場合は、outer はしばらく待って（デフォルトでは 3 分）から、再び空き CPU を探し始めます。よって利用者は特に何をする必要もなく、待っていれば自動的に進行します。

## 3) 計算が終了すると、計算結果がカレントディレクトリに戻ってきます。

サーバでの計算が終了すると、計算ファイル一式が自動的にカレントディレクトリに戻ってきます。これは通常の計算結果と同様、RYUGA などでも普通に解析できます。

- 計算結果が戻ってくるとき、カレントディレクトリにあった同名のファイルは上書きされますので、ご注意ください。（カレントディレクトリのファイルを計算投入中に変更した結果は計算終了時に失われます。）
- ファイルのタイムスタンプは戻ってきた時の日時になります。もしファイルの日時を利用するツール類をご使用になる場合はご注意ください。

## outer 利用に当たっての注意事項(必ずお読みください)

- `outer` はサーバ機にテンポラリディレクトリを作成して、そこに計算ファイルを置き、計算プログラムを実行します。テンポラリディレクトリの名称は `outertmpXXXX` です (XXXX は 4 桁の計算管理番号)。もし、サーバ機に同名のディレクトリが存在した場合、`outer` が異常動作しますのでご注意ください。
- 計算管理番号は `outer` が計算を投入する際に画面に表示されますが、すぐに計算プログラムの表示で流されてしまうので、確認できないかも知れません。その場合、後述の `showr` プログラムで確認できます。
- 一度に多数の `outer` を開始しないでください。サーバ CPU の空き状況を見誤って過剰に計算を入れてしまう可能性があります。前に開始した `outer` プログラムがサーバ CPU の空き状況を調べ終わるまで、次の `outer` を実行しないでください。
- 計算中は、そのウィンドウを閉じないでください (最小化は可)。もし閉じてしまっても、おそらく計算は止まりませんが、計算結果が戻ってこなくなります。この場合、投入先のサーバから手動で計算結果を引き上げ、サーバ内の計算ディレクトリ (`outertmp~`) を削除してください (`outertmp~` は各自のアカウントのホームディレクトリ直下にあります。ログインしてすぐのところ です)。
- サーバに計算結果を残してはいけません。計算結果が残っていると、サーバのハードディスクが圧迫され、他の方の計算投入に支障が発生して大迷惑になります。その場合、名前を公表して警告のうえ、計算結果を強制的に削除することがありますので、あらかじめご了承ください。
- 計算が完了するまで、クライアント側の計算ディレクトリ内容は変化しません。計算していないものと勘違いして、再度計算を投入したりしないよう、ご注意ください。万一、二重投入してしまった場合、結果が保証できないだけでなく、サーバの CPU をよけいに消費することにもなりますので、大変好ましくありません。
- もし、計算条件を誤ったなどで、計算を途中で止めたい場合は、`outer` を実行したウィンドウで `CTRL-C` を押してください。計算プログラムが強制終了し、それを `outer` が正常終了したものと判断して、計算結果をクライアント側に戻し、サーバから計算結果をテンポラリディレクトリごと削除します。ただし、環境によっては、`CTRL-C` で終了できないことがありますので、ご注意ください。
- `SCREEN` コマンドの使用については当方分かりませんのでサポートできません。`SCREEN` コマンドを使用したい場合は、各自の責任で行ってください。

**厳重注意** : `outer` 実行中のウィンドウを閉じてしまうなど、計算結果が戻ってこなかったときは、サーバに計算結果が残っていることがあります (`showr` プログラムで確認できます)。この場合、手動で計算結果をクライアントに転送し、サーバから計算結果を削除してください (テンポラリディレクトリ (`outertmp~`) ごと)。なお、サーバに計算結果を残すことは**絶対に**避けてください。サーバのハードディスクが圧迫され、他の方の計算投入に支障が発生して大迷惑になります。その場合、名前を公表して警告のうえ、計算結果を強制的に削除することがありますので、あらかじめご了承ください。

### 3. 3. 投入中の計算を一覧表示 (`showr` プログラム)

現在、サーバに投入中の計算は、`showr` プログラムで確認できます。

クライアント機の適当なディレクトリで「showr」とだけ入力して実行してください。  
以下に表示例を示します。

```
showr ver.1.12, 2005/1/18, by Ryuji Miura
Listup processes
 12 akinux262 /home/miura/lab/test1/ryudo
 13 akinux263 /home/miura/lab/test2/ryudo106
```

各行は左から「管理番号、計算サーバ名、親機のディレクトリ／プログラム名」となっています。

もし実行中でない計算が表示された場合、そのサーバに計算結果が残っている可能性があります。そのサーバに rsh や telnet などログインし、当該テンポラリディレクトリを削除してください。テンポラリディレクトリ名は、outertmpXXXX です（XXXX は管理番号です。たとえば管理番号 12 番の計算は outertmp0012 のテンポラリディレクトリになります）。

サーバに計算結果を残すことは**絶対**に避けてください。サーバのハードディスクが圧迫され、他の方の計算投入に支障が発生して大迷惑になります。その場合、名前を公表して警告のうえ、計算結果を強制的に削除することがありますので、あらかじめご了承ください。

### 3. 4. 投入実行中の計算途中結果の RYUGA 表示

通常、計算結果は終了までクライアントに戻ってこないのので、計算途中の様子を確認することが出来ません。（showr プログラムを使って投入先のサーバ名を調べ、ログインして計算途中のファイルを見ることは出来ます。）

showr プログラムに計算管理番号を付けて起動すると、ryuga で計算途中を読み込み表示させることが出来ます。

計算管理番号はあらかじめ showr プログラムで確認してください。

showr の後に空白を開けてから管理番号を付けて起動してください。

例 showr 12

この機能を利用するには、サーバに ryuga がインストールされている必要があります（サーバに rsh などログインし、そのまま ryuga と打って起動すれば OK です。）

起動した ryuga は計算ディレクトリがカレントになっていますので、そのまま読み込みを開始すれば計算途中結果を視覚化できます。

注意：

この機能を使うには、手元のマシンで外部から ryuga の画面が出るようにしておく必要があります。

- ・手元のマシンが Windows の場合、astec-x を立ち上げておいてください。

- ・手元のマシンが Linux(UNIX)の場合、xhost + を実行しておいてください。

ネットワーク越しの連携が微妙に絡みますので、うまく動かないケースがあります。その際は三浦までご相談ください。

また、現在この機能では「ryuga」という名前しか起動できません。例えば ryuga60 など名前を変更した物は起動できませんのでご注意ください。逆に ryuga でないプログラムで

も、サーバの `bin` ディレクトリでとりあえず「ryuga」という名前にしておけば、`showr` から起動できます（この方法で `Momonga` などでも起動できると思いますが未確認）。

---

## 4. その他

### 4. 1. 複数の `outer`

サーバのリストを分けたい場合、複数のクライアントマシンを使いたい場合などは、`outer` を複製することで対応できます。（`COLORS` 大規模計算専用共用 `Xeon` マシンを使う場合も、この方法で行ってください。）

これにはまず、コンパイル後の `outer` を、数字を付けてコピーしてください。

例 `cp outer outer2`

名前が変わった `outer2` は、サーバリストファイルとして `.outerc2` を、計算ログファイルとして `.outerlog2` を使用します。それぞれ `.outerc`、`.outerlog` とは別に用意してください。

また、サーバの計算テンポラリディレクトリ名としては、`outer2tmpXXXX` という形式になりますので、従来の `outer` とは重複しなくなります。

名前を変えた後も、使用方法は従来の `outer` と同様です。

`showr`、`upout` についても、`showr2`、`upout2` とすることで `outer2` に対応できます。

注意： `outer` の名前を変えても、基本的な動作は一緒ですので、やはり短時間に多数の `outer`、`outer2` などを開始しないでください。理由は同じく、サーバ `CPU` の空き状況を見誤るからです。クライアントマシンが別の場合も同様です。逆に言うと、自分と他人とがたまたま同時に `outer` を開始した場合も同様の問題が発生します。現状でこれは不可避です。このため、混雑時はしばしば `CPU` 数以上の計算がサーバに投入されてしまうことがあります。ご了承ください。

参考： `outer` の名前を変えても、投入できる計算数が増えるわけではありません。理由はよく考えれば分かると思いますが・・・無理に `outer` の名前を増やさないでください。混乱しやすくなります。

### 4. 2. 特記事項

- 上記の利用方法は今後変更する可能性があります。当方のアナウンスにご注意ください。
- サーバマシンは、`OS` バージョンアップなどのため、再起動がかかることがあります。その際は、数日前に連絡しますが、あらかじめご了承ください。
- 上記説明でわからない点があれば、三浦までお問い合わせください。
- `outer/showr/upout` について、バグ（と思われる症状）を発見した場合は、三浦まで

お知らせください。