

Guia do Usuário

Dinamo

dinamo.gruponano.br

UFOP

versatushpc.com.br

A Versatus

Com foco exclusivo em Computação de Alto Desempenho (HPC), a Versatus é uma empresa Brasileira especialista em soluções de HPC. Utilizando software e hardware de ponta combinados com uma grande gama de serviços profissionais, ajudamos nossos clientes a criar soluções de HPC de alta qualidade, eficiência e confiabilidade.

Nossa missão é permitir que pesquisadores e instituições de pesquisa possam focar exclusivamente nos seus objetivos e produção científica, fornecendo o que há de melhor em termos de equipamentos, softwares e gerenciamento completo de ambientes computacionais.

A Versatus constrói clusters de alto desempenho sob medida para seus clientes. Com uma abordagem holística que analisa desde as técnicas e algoritmos computacionais que seus clientes utilizarão até a infraestrutura elétrica e térmica do recinto em que os sistemas repousarão, a Versatus

projeta o cluster certo para resolver o seu problema computacional em pesquisa ou produção. Isto garante que você adquira um sistema tecnicamente robusto, ecologicamente responsável, com preço competitivo e um suporte de alta qualidade. Este guia tem o objetivo de oferecer um início rápido aos usuários do sistema computacional Dinamo-UFOP descrevendo brevemente o cluster e o sistema instalado, além do seu uso.

COPYRIGHT

© 2020 VersatusHPC. Todos os direitos reservados; algumas partes deste manual podem ter direitos autorais protegidos por terceiros. O presente manual é distribuído em sua íntegra e não é permitida nenhuma alteração, cópia, distribuição ou criação de trabalhos derivados dos conteúdos deste documento, total ou parcialmente, sem a expressa autorização por escrito da VersatusHPC.

Conteúdo

Descrição do Cluster

- Hardware

 - Headnode

- Partições

- Arquivos e Armazenamento

- Nós de Processamento

- Forma de Acesso

Softwares Instalados

- Compiladores

 - GNU (gcc, g++, gfortran)

 - Intel Parallel Studio

- Bibliotecas de Transferência de Mensagem

 - Intel MPI

 - MPICH

 - OpenMPI

- Bibliotecas Matemáticas e Auxiliares

 - FFTW

 - Intel MKL

 - HDF5

 - OpenBLAS

 - PnetCDF

 - NetCDF

 - ScaLAPACK

- Softwares Científicos

 - GROMACS

 - LAMMPS

 - ORCA

 - Quantum ESPRESSO

 - SIESTA

Variáveis de Ambiente

Utilizando o Cluster

- Sistema Operacional

- Alterando a Senha de Usuário

Submetendo e Gerenciando Jobs

Diretivas e Variáveis do Job

Filas

Tipos de Jobs

Batch Job

Exemplo 1

Exemplo 2

Job Interativo

Monitorando os seus Jobs

Cancelando Jobs

Problemas, Resoluções e Dúvidas

Obtendo Ajuda

Descrição do Cluster

Hardware

Headnode

1(um) Nó de Gerenciamento R640 com fontes redundantes e armazenamento RAID, configurado com:

- 1 (um) processador Intel Core i7 8700;
- 16 (dezesesseis) GB de RAM;
- 1 (um) SSD de 120GB;
- 3 (três) HDDs de 2TB em RAID5.

Partições

Headnode:

Sist. Arq.	Tipo	Tam.	Usado	Disp.	Uso%	Montado em
/dev/mapper/c1-root	xfs	50G	14G	37G	28%	/
/dev/sdc2	xfs	1014M	196M	819M	20%	/boot
/dev/sdc1	vfat	200M	13M	188M	7%	/boot/efi
/dev/md127	xfs	3,7T	18G	3,7T	1%	/home
/dev/mapper/c1-opt	xfs	53G	27G	27G	50%	/opt

Arquivos e Armazenamento

O diretório padrão do usuário é construído da seguinte forma:

/home/<usuário>/

Ex: */home/vhpc/*

Esta área possui **3,7 TB disponíveis** que são compartilhados entre todos os usuários.

Nós de Processamento

O cluster Dinamo possui 9 nós listados nomeados no formato "dinXX". Ex: "din01" (ver Nós Computacionais).

Forma de Acesso

O acesso remoto é feito através de SSH (Secure SHell):

```
$ ssh -p2707 <usuário>@200.239.138.112
```

Softwares Instalados

O sistema tem um grande conjunto de softwares instalados de acordo com as necessidades dos usuários. Os softwares disponíveis em produção deste guia estão listados abaixo.

Todos os softwares, compiladores e bibliotecas pré-instalados no cluster estão disponíveis em `'/opt/ohpc/pub/'`.

Compiladores

A família de compiladores Intel e GNU e suas ferramentas, estão disponíveis nas seguintes versões:

GNU (gcc, g++, gfortran)

Versão 8.3.0

Executáveis: `'/opt/ohpc/pub/compiler/gcc/8.3.0/bin/'`

Bibliotecas: `'/opt/ohpc/pub/compiler/gcc/8.3.0/lib/'`

Module: `'gnu8/8.3.0'`

Intel Parallel Studio

Versão: 2020.2

Executáveis: `'/opt/ohpc/pub/compiler/intel/bin/'`

Bibliotecas: `'/opt/ohpc/pub/compiler/intel/lib/'`

Module: `'intel/2020.2.254'`

Bibliotecas de Transferência de Mensagem

As bibliotecas MPI pré-instalados na workstation são as seguintes:

Intel MPI

Versão 2020.2:

Executáveis:

`'/opt/ohpc/pub/compiler/intel/compilers_and_libraries_2020.2.254/linux/mpi/bin/'`

Bibliotecas:

`'/opt/ohpc/pub/compiler/intel/compilers_and_libraries_2020.2.254/linux/mpi/intel64/lib/'`

Module: `'intel-mpi-2020.2.254-intel-20.2.254-gicawn6'`

MPICH

MPICH é outra biblioteca de troca de mensagens aberta existente

Versão 3.3.1 compilada com GNU 8.3.0

Executáveis: `'/opt/ohpc/pub/mpi/mpich-gnu8-ohpc/3.3.1/bin/'`

Bibliotecas: `'/opt/ohpc/pub/mpi/mpich-gnu8-ohpc/3.3.1/lib/'`

Module: **'mpich/3.3.1'**

OpenMPI

OpenMPI é uma biblioteca de troca de mensagens de código aberto implementada pela Open MPI Project.

Versão 3.1.4 compilada com GNU 8.3.0:

Module: **'openmpi3/3.1.4'**

Versão 3.1.6 compilada com GNU 8.3.0:

Module: **'openmpi-3.1.6-gcc-8.3.0-jpd76yz'**

Versão 3.1.6 compilada com Intel 2020.2:

Module: **'openmpi-3.1.6-intel-20.2.254-3jqvcl3'**

Bibliotecas Matemáticas e Auxiliares

Estão pré-instalados no cluster as seguintes bibliotecas:

FFTW

Essa biblioteca são sub-rotinas para cálculo das transformadas rápida e discreta de Fourier, além de transformadas discretas do seno e cosseno e transformadas de Hartley.

Versão 3.3.8 compilada com GNU 8.3.0:

Module: **'fftw/3.3.8'**

Versão 3.3.8 compilada com GNU 8.3.0:

Module: **'fftw-3.3.8-gcc-8.3.0-2bwctuf'**

Intel MKL

Versão 2020.2:

Executáveis:

`'/opt/ohpc/pub/compiler/intel/compilers_and_libraries_2020.2.254/linux/mkl/bin'`

Bibliotecas: `'/opt/ohpc/pub/compiler/intel/compilers_and_libraries_2020.2.254/linux/mkl/lib'`

Module: **'intel-mkl-2020.2.254-intel-20.2.254-l4g5557'**

HDF5

Hierarchical Data Format (HDF, HDF4, ou HDF5) nome para um conjunto de formatos de arquivos e bibliotecas criadas para organização e armazenamento de grandes quantidades de dados numéricos.

Versão 1.10.5 compilada com GNU 8.3.0:

Module: **'hdf5/1.10.5'**

OpenBLAS

O BLAS (Básicos Álgebra Linear Subprograms) são rotinas que fornecem blocos de construção padrão para a realização de operações básicas vetoriais e matriciais.

Versão 0.3.5 compilada com GNU 8.3.0:

Module: **'openblas/0.3.5'**

Versão 0.3.5 compilada com GNU 8.3.0:

Module: **'openblas-0.3.10-gcc-8.3.0-x6t26gh'**

PnetCDF

PnetCDF é uma biblioteca de E/S (entrada/saída) paralelo de alta desempenho para acessar arquivos compatíveis com NetCDF.

Versão 1.10.1 compilada com GNU 8.3.0:

Module: **'pnetcdf/1.10.1'**

NetCDF

Network Common Data Form (netCDF) é um conjunto de bibliotecas de softwares criadas para melhor organização de dados científicos.

Versão 4.6.3 compilada com GNU 8.3.0:

Module: **'netcdf/4.6.3'**

ScaLAPACK

O Scalable LAPACK é uma biblioteca de rotinas de alto desempenho para álgebra linear baseada no LAPACK.

Versão 2.0.2 compilada com GNU 8.3.0:

Module: **'scalapack/2.0.2'**

Softwares Científicos

Os softwares científicos instalados são mostrados em cada tópico abaixo com seus respectivos módulos de ambiente para carregamento via comando **"module add/load"**:

GROMACS

Versão 2020.3 compilada com GNU 8.3.0:

Module: **'gromacs-2020.3-gcc-8.3.0-rcu5hl2'**

LAMMPS

Versão 20190919 compilada com GNU 8.3.0:

Module: '**lammps-20190919-gcc-8.3.0-zpafs4z**'

ORCA

Versão 4.2:

Module: '**orca/4.2.1**'

Quantum ESPRESSO

Versão 6.4 compilada com GNU 8.3.0:

Module: '**quantum-espresso-6.4.1-gcc-8.3.0-lmg6hcd**'

SIESTA

Versão 4.0 compilada com GNU 8.3.0:

Module: '**siesta-4.0.2-gcc-8.3.0-dr3gfro**'

Versão 4.0 compilada com Intel 2020.2:

Module: '**siesta-4.0.2-intel-20.2.254-vcoiat3**'

Variáveis de Ambiente

Em ambientes de computação de alto desempenho é comum que haja diversos compiladores e bibliotecas diferentes para a mesma rotina. Por exemplo, para compilação de códigos C/C++, Fortran e MPI, temos disponíveis “gcc”/“gfortran” e “mpicc”/“mpifort”. Além dessa variação de softwares que desempenham a mesma função, também é comum “coleccionar” diferentes versões de compiladores, bibliotecas e softwares. Com isso, surge a necessidade de selecionar quais arquivos queremos que o sistema operacional disponibilize para o uso, de acordo com a atividade que iremos desempenhar. No cluster Dinamo está disponível uma lista com todas as bibliotecas e compiladores presentes na máquina, basta o usuário selecionar quais ferramentas ele irá utilizar. Essa seleção é feita por sessão, ou seja, a cada job, as ferramentas devem ser selecionadas, e ao final do job, elas voltam ao valor padrão de “nenhuma seleção”. O comando utilizado para este gerenciamento é o “module” [<http://lmod.readthedocs.io/en/latest/>], ele pode listar, carregar, descarregar ou substituir uma ferramenta dentre as disponíveis.

Os módulos de variáveis de ambiente que você necessita podem ser carregados manualmente na linha de comando. Ainda assim, é mais conveniente incluir os módulos para serem carregados nos scripts de submissão que você utilizar (veja [Submetendo e Administrando Jobs](#) e alguns exemplos que seguem). Outra possibilidade é modificar o seu “.bash_profile” para carregar os módulos automaticamente no login (não esqueça de descarregar os módulos se há algum conflito com algum outro módulo que você precisa para um software específico).

Os módulos de variáveis atualmente disponíveis são:

```
[vhpc@headnode ~]$ module avail
```

A sintaxe do module é:

- **module av, avail** - lista todos os módulos disponíveis;
- **module list** - lista todos os módulos atualmente em uso;
- **module show <caminho_do_módulo>** - mostra o conteúdo do arquivo do módulo;
- **module add, load <caminho_do_módulo>** - carrega as configurações e prepara o ambiente de acordo com o módulo selecionado;
- **module rm, unload <caminho_do_módulo>** - remove as configurações específicas deste módulo;
- **module purge** - remove todos os módulos carregados ;
- **module help** - exibe todos os subcomandos do module.

Referência completa (em inglês): [http://lmod.readthedocs.io/en/latest/010_user.html]

Ex:

Cenário 1: Compilar um programa utilizando o GCC 8.2, para esta compilação o usuário deve carregar o módulo do compilador com o comando:

```
[vhpc@headnode ~]$ module add gnu8/8.2.0
```

Cenário 2: O usuário deseja remover todos os módulos já carregados:

```
[vhpc@headnode ~]$ module purge
```


Utilizando o Cluster

Sistema Operacional

A versão do seu sistema operacional é o CentOS Linux 7.8 x86_64, distribuição baseada no Red Hat Enterprise Linux.

Alterando a Senha de Usuário

Para alterar a senha de seu usuário no sistema, use o comando:

```
[vhpc@headnode ~]$ kpasswd
```

Assim que você digitar este comando, o sistema irá iniciar automaticamente o processo onde você deverá:

- Digitar a senha atual do usuário;
- Digitar a nova senha do usuário;
- Digitar novamente a nova senha do usuário.

Submetendo e Gerenciando Jobs

O Dinamo conta com o sistema de filas instalado para submeter e administrar jobs via scripts.

Para que os recursos computacionais sejam divididos de maneira razoável entre todos os usuários, é necessária uma forma de organizar e priorizar as requisições de uso, comumente chamadas de “jobs”. Para isso, o cluster conta com o sistema de filas PBSPro [\[https://www.altair.com/pdfs/pbsworks/PBSUserGuide19.2.3.pdf\]](https://www.altair.com/pdfs/pbsworks/PBSUserGuide19.2.3.pdf) para gerenciar os jobs que serão executados, respeitando as políticas estabelecidas. O PBSPro trabalha com o conceito de filas, que são estruturas para classificar e agrupar os jobs e nós computacionais sob critérios como: número de processadores requisitados, quantidade de nós, quantidade de memória RAM, tempo de processamento e assim por diante. Na configuração atual, estão configuradas partições com ordem de prioridade e em que apenas alguns grupos de usuários têm acesso a algumas delas (na seção [Filas](#) temos mais detalhes).

É possível exibir as filas do sistema e os seus status com o comando:

```
[vhpc@headnode ~]$ qstat -q
```

Para exibir os nós computacionais presentes no cluster utilize o comando:

```
[vhpc@headnode ~]$ pbsnodes -a
```

Diretivas e Variáveis do Job

Dentro dos jobs existem algumas instruções que são dedicadas ao PBSPro, entre outras coisas para instruí-lo onde, como e por quanto tempo rodar. Essas instruções são chamadas diretivas, e estão nas linhas que se iniciam com: “**#PBS**”. O PBSPro irá considerar todas as diretivas até que encontre uma linha executável, depois deste ponto, será ignorada qualquer outra diretiva.

Diretivas comuns são:

#PBS -N <nome> → Nome do job

#PBS -l select=<número>:ncpus=<número> → Número total de nós e núcleos

#PBS -l walltime <dias-hh:mm:ss> → Tempo de execução (dias-hh:mm:ss)

#PBS -q <fila> → Fila onde o job será alocado

Além disso, quando o job é submetido, o PBSPro coloca a disposição do job algumas variáveis de ambiente que tem o intuito de facilitar algumas operações. Algumas delas, por exemplo são:

\$PBS_NODEFILE → esta variável contém o caminho do arquivo com a lista de nós e núcleos que foram alocadas ao job, é útil, por exemplo para ser usada como parâmetro para o “mpirun”:

```
mpirun -machinefile $PBS_NODEFILE meuprograma
```

\$PBS_WORKDIR → esta variável aponta para o diretório de onde o script foi submetido.

\$PBS_JOBNAME → contém o nome especificado na diretiva “**#PBS -N**”

A lista completa de diretivas, opções e variáveis pode ser encontrada em (em Inglês): [\[https://www.altair.com/pdfs/pbsworks/PBSUserGuide19.2.3.pdf\]](https://www.altair.com/pdfs/pbsworks/PBSUserGuide19.2.3.pdf).

Filas

As filas, também chamadas de partições, são o que permitem um bom desempenho, distribuição e agendamento de uso de recursos. Na cluster Dinamo temos apenas as filas configuradas listadas abaixo:

Fila	Tempo limite	Nó(s) associado(s)	Observação
medium	7 dias	din[01-42]	Fila padrão
long	14 dias	din[01-42]	
infinite	Sem limite	din[01-42]	

A política de filas está com o fairshare de 14 dias ativado, ou seja, o histórico de 14 dias de uso será utilizado para definir a prioridade dos jobs submetidos por cada usuário.

Tipos de Jobs

Quanto aos jobs, existem dois tipos mais comuns:

Batch Job

Um arquivo de script que contém comandos para executar aplicações específicas, usado pelo comando “qsub” para informar ao PBSPro as características do seu job. Exemplo de script:

```
#!/bin/bash
#PBS -l walltime=24:00:00
#PBS -l select=1:ncpus=8:mpiprocs=8
#PBS -N teste

cd $PBS_WORKDIR

#### Prepare o ambiente com os devidos módulos #####
module add openmpi3/3.1.4

#### Exporte as variáveis relevantes para o job #####3
export OMP_NUM_THREADS=1

#### Execução do job ####
mpirun -np $PBS_NTASKS prime 200000 > teste.out
```

- A primeira linha informa ao PBSPro que o Shell escolhido pelo usuário para rodar o job é o Bash (“/bin/bash”);
- A segunda linha indica, em horas reais, qual a duração do job;
- A terceira linha indica o número de nós e núcleos solicitados para este job;

- A variável **\$PBS_WORKDIR** tem por valor padrão o diretório onde o job foi submetido. Logo após seguem os comandos para a execução do job propriamente dito;
- Variáveis de ambiente necessárias à execução do job devem ser declaradas/exportadas antes do job.

Exemplo da submissão de um batch job:

```
[vhpc@headnode ~]$ qsub script.pbs
```

Abaixo são apresentados scripts de submissão com ideias que podem ser mescladas para se construir um script ideal para cada tipo de job.

Exemplo 1

O exemplo abaixo mostra um job simples de execução de um software genérico em 20 núcleos e walltime de 2 horas:

```
#!/bin/bash
#PBS -N TESTE
#PBS -l select=1:ncpus=20:mpiprocs=20
#PBS -l walltime=2:00:00

module load openmpi3/3.1.2

cd $PBS_WORKDIR

mpirun -np $PBS_NTASKS ./meuprograma
```

Exemplo 2

O job abaixo é um exemplo de execução do software em 16 núcleos de 1 nó e walltime de 30 horas:

```
#!/bin/bash
##
## Copyright (C) 2009-2018 VersatusHPC, Inc.
##
## select = quantidade de nodes; ncpus = qt de nucleos
#PBS -l select=1:ncpus=16:mpiprocs=16
##
## walltime = quantidade de tempo
#PBS -l walltime=30:00:00
##
## Nome do job . Aparece na saída do comando 'qstat' .
## E recomendado, mas nao necessario, que o nome do job
## seja o mesmo que o nome do arquivo de input
#PBS -N <Job_Name>
##
## Consulte [https://www.altair.com/pdfs/pbsworks/PBSUserGuide19.2.3.pdf] para mais
informacoes
## sobre as diretivas acima.
```

```

echo -e "\n## Job iniciado em $(date +%d-%m-%Y as %T') #####\n"

## Variavel com o diretorio de scratch do job
## Consulte [https://www.altair.com/pdfs/pbsworks/PBSUserGuide19.2.3.pdf] para mais informacoes
## sobre as variaveis de ambiente do PBSPro.
WRKDIR=$SCRATCH/$PBS_JOBID

## O nome dos arquivos de input e output sao baseados no
## nome do job (linha "#PBS --jobname=xxx" acima).
## Observe que nao e obrigatorio esta forma de nomear os arquivos.
INP=$PBS_JOBNAME".inp"
OUT=$PBS_JOBNAME".out"

## O diretorio onde o job sera executado sera apagado, por padrao, ao
## final do mesmo.
## Se desejar que nao seja apagado, substitua Y por N.
APAGA_SCRATCH=Y

## Informacoes do job impressos no arquivo de saida.
echo -e "\n## Jobs ativos de $USER: \n"
squeue -a --user=$USER
echo -e "\n## Node de execucao do job:          $(hostname -s) \n"
echo -e "\n## Numero de tarefas para este job: $PBS_NTASKS \n"

#####
##----- Inicio do trabalho      ----- #
#####

## Configura o ambiente de execucao do software.
module load @SOFTWARE_MODULE@

## Informacoes sobre o ambiente de execucao impressos no arquivo de saida.
echo -e "\n## Diretorio de submissao do job:   $PBS_WORKDIR \n"
echo -e "\n## Diretorio de scratch do job:        $WRKDIR \n"
echo -e "\n## Arquivo de input:                    $INP \n"

## Cria o diretorio de scratch para o job.
mkdir -p $WRKDIR

## Transfere os inputs e arquivos necessarios para o diretorio de scratch
cd $PBS_WORKDIR
cp $INP $WRKDIR/
cd $WRKDIR

## Execucao do software

mpirun ./mysoftware < $INP > $OUT

## Copia o diretorio de scratch para o diretorio original do job.

```

```

cp -r $WRKDIR/ $PBS_WORKDIR/

## Apaga o diretorio de scratch do job.
if [ x"$APAGA_SCRATCH" = x"Y" ]; then
    rm -rf $WRKDIR
else
    echo -e "\n0 diretorio \e[00;31m$WRKDIR\e[00m deve ser removido manualmente"
    echo -e "para evitar problemas para outros jobs e/ou usuarios. \n"
fi

echo -e "\n## Job finalizado em $(date +%d-%m-%Y as %T) #####"

```

Job Interativo

É executado como um “batch job”, porém o terminal do usuário é conectado ao host de execução, similar a uma sessão de login. A partir daí o usuário envia as opções do script de job como comandos individuais, o que facilita ao usuário depurar um script com problemas.

Para submeter um job interativo basta executar o comando abaixo:

```
[vhpc@headnode ~]$ qsub -I
```

Para encerrar o job interativo, basta sair do terminal com o comando “exit”.

Outras opções podem ser utilizadas no comando “qsub”, referência completa (em inglês): [\[https://www.altair.com/pdfs/pbsworks/PBSUserGuide19.2.3.pdf\]](https://www.altair.com/pdfs/pbsworks/PBSUserGuide19.2.3.pdf).

Monitorando os seus Jobs

Para verificar o estado do(s) seu(s) job(s), usa-se o comando “**qstat**”. Algumas das opções mais usadas são:

- **-u <usuário>** - mostra somente as informações referentes ao usuário “usuário”
- **-f <jobid>** - mostra informações completas referentes ao job com a id “jobid”

Referência completa (em inglês): [\[https://www.altair.com/pdfs/pbsworks/PBSUserGuide19.2.3.pdf\]](https://www.altair.com/pdfs/pbsworks/PBSUserGuide19.2.3.pdf)

Cancelando Jobs

O comando utilizado para apagar os jobs da fila é o “**scancel**”, a sintaxe é:

- **qdel <jobid>** - jobid é o número que o PBSPro retorna quando o job é aceito e enfileirado.
- **qdel --help** - mostra brevemente as opções do comando.

Referência completa (em inglês) [\[https://www.altair.com/pdfs/pbsworks/PBSUserGuide19.2.3.pdf\]](https://www.altair.com/pdfs/pbsworks/PBSUserGuide19.2.3.pdf)

Problemas, Resoluções e Dúvidas

A documentação oficial do PBSPro mantém uma compilação de problemas comuns e suas resoluções como também dúvidas mais frequentes sobre o sistema de filas.

Para problemas e resoluções (em inglês):

[\[https://www.altair.com/pdfs/pbsworks/PBSUserGuide19.2.3.pdf\]](https://www.altair.com/pdfs/pbsworks/PBSUserGuide19.2.3.pdf)

Para dúvidas e questões frequentes (em inglês):

[\[https://www.altair.com/pdfs/pbsworks/PBSUserGuide19.2.3.pdf\]](https://www.altair.com/pdfs/pbsworks/PBSUserGuide19.2.3.pdf)

Obtendo Ajuda

Em caso de contrato ativo de manutenção continuada os chamados podem ser abertos por telefone (**+55 11 3436-0664**) ou através de e-mail [**suporte@versatushpc.com.br**], fornecendo o ID do cliente e as informações abaixo:

1. O que você está tentando fazer (executar um programa, copiar um arquivo, acessar um diretório);
2. Passos que você está executando para reproduzirmos o problema;
3. Qual/quais as mensagens de erro estão sendo exibidas;
4. Quais os passos alternativos você já tentou e quais os resultados.

HAPPY
CLUSTERING

contato@versatushpc.com.br



Rua Estela, 515, 21B

Paraíso - São Paulo - SP

CEP 04011-002 - BR

Fone: +55 11 3436-0664 | 11 2275-9733