

Лаборатория по Високо Производителни Изчисления (ЛВПИ)
София ТехПарк
и
СУ “Св. Климент Охридски”
Физически факултет

Ръководство за работа със софтуерните пакети
UEMS и WRFv3.7.1

гл. ас. д-р Стоян Кръстев Писов

1 август 2017 г.

1 Въведение

Този документ е предназначен за вътрешна употреба и служи като инструкция за работа на потребителите на изчислителните кълъстери Nestum и Physon. Ръководството описва стъпките за инсталация на програмните пакети за метео прогнози UEMS и WRF и може да се използва потребителите които имат активен потребителски акаунт на единият от двата изчислителни кълъстера и са запознати с основните правила за работа с тях. За повече детайли можете да разгледате страниците, Nestum: <http://nestum.phys.uni-sofia.bg/jobs-submission/> и Physon: <http://physon.phys.uni-sofia.bg/work-with-sge-bg>. Самите примери по-долу потребителски акаунт `usrNN` където `NN` е номер на потребителя (`usr01`, `usr02`, `usr03`, ..., `usr32`) на изчислителният кълъстер нестум. Всеки потребител има домашна папка с префикс `/home/course/usrNN` с размер 5G и работна папка `/work/course/usrNN` с групова квота от 500 GB. Достъпът до системата се осъществява с помощта на SSH сесия с помощта на командата:

```
ssh -Y usrNN@nestum.phys.uni-sofia.bg
```

1.1 Система за настройка на софтуерните пакети (`modules`)

Базовата операционна система на машината е Ubuntu 16.04. Допълнителният софтуер е достъпен през системата `modules`. С показаните команди по-долу можете да:

- Покажете списък на достъпните модули или софтуерни пакети: `module avail`
- Покажете заредените от вашата текуща сесия: `module list`
- Заредите конкретен модул: `module load openmpi`
- От-заредите конкретен модул: `module unload openmpi`

1.2 Система за пакетно изпълнение на задачи (`slurm`)

Достъпът до изчислителните ресурси на кълъстера Nestum става посредством система за пакетна обработка на задачите (`batch system`). Конкретно в случая се използва `slurm`. Можете да стартирате интерактивни и неинтерактивни задачи. Интерактивните задачи позволяват да имате достъп до изчислителните ресурси чрез интерактивна терминална сесия където имате достъп до входа и изхода на изпълнените от вас приложения докато неинтерактивните сесии се изпълняват във фонов (`background`) режим като изхода на вашите приложения се пренасочва във файлове. Интерактивните сесии са подходящи за анализ, разработване на софтуер, тест на софтуера и кратко изпълнение на програмите ви. Неинтерактивните задачи се използват за продуктивни дълги пресмятания:

Изпълнение на интерактивни задачи: `srun -p interact.p --pty /bin/bash` Изпълнение на неинтерактивни задачи: Неинтерактивните задачи се изпълняват чрез подходящ скриптов файл наречен `job` файл. Този файл по същество е `shell` скрипт в който са включени допълнителни директиви към `slurm` мениджъра. По-долу е показано съдържание на един такъв файл `myTask.job`:

```
#!/bin/bash
#
#SBATCH -p medium.p # partition (queue)
#SBATCH -N 2 # number of nodes
#SBATCH -n 64 # number of cores
```

```
#SBATCH -t 0-2:00 # time (D-HH:MM)  
#SBATCH -o slurm.%N.%j.out # STDOUT  
#SBATCH -e slurm.%N.%j.err # STDERR  
#SBATCH --mail-type=<type> # notification trigger  
#SBATCH --mail-user=<user> # email address
```

```
module load openmpi
```

```
mpirun helloworld.x
```

Изпращането на задачата ви към `slurm` мениджъра става с помощта на командата: `sbatch myTask.job`. Няколко допълнителни команди които могат да ви бъдат полезни са:

- Показване на списък с чакащите и стартираните в момента задачи: `squeue`
- Прекъсване на задача: `scancel job_id`
- Състояние на опашките със задачи: `sinfo`

Всяка една задача се може да се асоциира към една от трите дефинирани опашки чийто ресурс е време за изпълнение и максимален брой процесорни ядра:

- `short.p` - до 4 часа на 736 ядра
- `medium.p` - до 48 часа на 736 ядра
- `long.p` - до 168 часа на 736 ядра
- `interact.p` - до 24 часа на 32 ядра

За допълнителна информация можете да разгледате страницата на клъстера: <http://nestum.phys.uni-sofia.bg/>

2 Софтуерът UEMS (Unified Environmental Modeling System)

Програмен пакет предназначен за автоматизация на процеса на прогнозиране на времето NWP (Numerical Weather Prediction) като се използва както WRF (Weather Forecast Model) ядро на националният център за атмосферни изследвания на САЩ (NCAR - National Center for Atmospheric Research) така и NMM модела на националният център за прогнозиране на околната среда (NCEP - National Center for Environmental Predictions). Самата инсталацията съдържа множество скриптове, програми и инструменти за визуализация като например:

- `dwiz` - инструмент с графичен интерфейс за създаване на изчислителни области (домейни) за симулация
- `ems_prep`, `ems_run` и `ems_autorun` скрипт за подготовка и изпълнение на симулацията на базата на входни данни от глобалния мезоскален модел
- прекомпилирани изчислителни ядра на `wrf` модела `arw` и `nmm`
- `ems_post`, `ncview`, `ncdump` помощни скриптове за `postprocessing` последващата обработка на данни

Софтуерният пакет е достъпен при заявка от страницата STRC (Science and Training Resource Center) <http://strc.comet.ucar.edu/software/uems/> след регистрация.

2.1 Системни изисквания

Инсталирането на софтуерният пакет става през интернет (онлайн) с помощта на скрипта `uems_install.pl` (в по-старите версии скрипта се нарича `ems_install.pl`) който се получава след регистрация и заявка. Системните изисквания за минимална инсталация са:

- Linux операционна система с ядро ≥ 2.6
- PC архитектура с Intel или AMD процесори с поддръжка на SSE инструкции
- Минимум 4G системна памет, препоръчително е системата да има поне 8G системна памет
- 120 GB дисково пространство за пълна инсталация на UEMS пакета
- Допълнително дисково пространство > 30 GB за съхранение на генерираните от симулацията данни (може да зависи от продължителността и разделителната способност)
- Инсталиран Perl интерпретатор верси > 5
- Инсталиран един от двата инструмента за сваляне на файлове в интернет `wget` или `curl`

2.2 Помощна информация

При стартиране на скрипта `uems_install.pl` с параметър `--help` можете да получите детайлна информация за възможните параметри

```
./uems_install.pl --help
```

2.3 Инсталиране

В случай че смятате да направите глобална инсталция на UEMS е необходимо да се избере някоя общо достъпна папка във файловата система. Например

```
/opt/uems  
/usr/share/uems  
/srv/uems
```

папката трябва да има права за четене и писане на потребителя който извършва инсталирането. Глобалната инсталация може да се извърши от системен администратор или човек който има достатъчно опит в инсталирането на EMS за нуждите на група от хора или потребители на изчислителният клъстер. В случай че извършвате локална инсталация в домашната или работната папка на вашият потребител можете да изберете папка която е част от вашата домашна директория. Настоящият пример ще използва работната папка на потребителя `/work/course/usrNN/opt/uems` като скрипта предполага че директорийната структура съществува до точката на последната директория, т.е. вече съществува. Командата по-долу ще инсталира последната най-актуална версия на UEMS в папката `/work/course/usrNN/opt/uems` без да включва подложните данни от WRF модела с цел да се съкрати времето за инсталация:

```
./uems_install.pl --install --nogeog\  
--emshome=/work/course/usrNN/opt/uems
```

забележка: пътят указан в параметъра `--emshome` трябва да бъде абсолютен

В случай че правите пълна инсталция опцията `--nogeog` трябва да се пропусне. След стартиране на инсталацията скриптър ще зададе няколко уточняващи въпроса на които е достатъчно да потвърдите отговорите натискайки клавиша **[Enter]**. Базовата инсталация без подложните данните отнема около 5 - 10 мин в зависимост от интернет свързаността. В края на инсталацията скрипта отпечатва съобщение в което се разяснява как да се настроят променливите на средата така че да можете да използвате UEMS модела. В случая преди да използвате UEMS модела в зависимост от това какъв тип shell използвате `bash` или `csh/tcsh` е необходимо да заредите:

```
In order for you to use the UEMS, the environment variables must be  
set; therefore, it is recommended that you place the following lines  
in your /home/course/usrNN/.bash_profile file:
```

```
# Set the UEMS V15.99.5 environment variables  
#  
if [ -f /work/course/usrNN/opt/uems/etc/EMS.profile ]; then  
./work/course/usrNN/opt/uems/etc/EMS.profile  
fi
```

bash:

```
./work/course/usrNN/opt/uems/etc/EMS.profile
```

csh

```
source /work/course/usrNN/opt/uems/etc/EMS.cshrc
```

2.4 Директорийна структура на софтуерният пакет UEMS.

Веднъж инсталиран в директория с корен `/work/course/usrNN/opt/uems` софтуерният пакет създава следната директорийна структура:

- `$EMS_ROOT/etc` - папка която съдържа конфигурационни скриптове които настройват променливите на UEMS средата. Най-важните скриптове са `EMS.profile` и `EMS.cshrc`
- `$EMS_ROOT/bin` - папка с изпълними файлове. Тя съдържа прекомпилирани статични версии на инструментите за предварителна обработка (preprocessing): `ungrib`, `metgrid`. Паралелни версии на програмите от модела WRF: `wrf_arw.exe` и `real_arw.exe`
- `$EMS_ROOT/strc` - папка съдържаща множество работни скриптове на езика Perl които са сърцето на пакета UEMS. Тук се намират и скриптовите за подготовка и изпълнение на симулацията: `ems_prep` и `ems_run`
- `$EMS_ROOT/runs` - основна папка в която подразбиране се създават работните директории на изчислителните домейн. Потребителя трябва да има права за четене и запис в нея папка. Тя може да се преконфигурира с помощта на променливата на средата `EMS_RUN`
- `$EMS_ROOT/data` - папка с данни необходими на моделите WRF, NMM и dwiz. Например в тази папка се съхраняват данни за подложната повърхност.
- `$EMS_ROOT/logs` - работна папка в която се съхранява информация за изпълнението на скриптовите и програмите от пакета UEMS

2.5 Създаване на изчислителни области (домейн).

След като сте инсталирали успешно UEMS модела можем да пристъпим към следващата стъпка създаване на изчислителен домейн забележка при последната стъпка от създаване на домейна ще получите съобщение за грешка че домейна не може да бъде локализиран ... заради липсата на данни за подложната повърхност. затова можем да създадем символична връзка към съществуваща папка с данните за подложката:

```
cd /work/course/usrNN/opt/uems/data
```

```
ln -s /opt/lx/ems/15.58.5/uems/data/geog
```

и да стартираме от командният ред `ems_domain`. За целта е необходимо да сменим текущата директория с папката в която се намира нашият ново създаден домейн `bulgaria`. Подразбиране тази папка е част от директорийната структура и се намира в папката

```
$EMS_ROOT/runs/bulgaria
```

или в нашият случай

```
cd /work/course/usrNN/opt/uems/runs/bulgaria
```

Когато работите на изчислителният клъстер Nestum или Physon добра практика е изпълнението на командите да става в интерактивна сесия на някой от изчислителните възли. За целта предварително изпълнете командата (Nestum):

```
srun -n 16 --pty bash
. /work/course/usrNN/opt/uems/etc/EMS.profile
ems_domain --ncpus 1 --localize
```

или ако ползвате Physon:

```
qlogin -l h_rt=8:0:0,h_vmem=2G
cd /work/course/usrNN/opt/uems/runs/bulgaria
. /work/course/usrNN/opt/uems/etc/EMS.profile
ems_domain --ncpus 1 --localize
```

Същинската симулация може се стартира по два начина. Единият от тях е напълно автоматизиран при който се изпълнява директно скрипта `ems_autorun`, който сваля граничните условия необходими за симулацията и стартира самата симулация. Ние ще разгледаме другият подход при който тези две стъпки се изпълняват самостоятелно:

2.6 Подготовка на началните данни.

Осъществява се със скрипта `ems_prep` който съществува като символична връзка в папката на домейна който сме създали:

```
/work/course/usrNN/opt/uems/runs/bulgaria/ems_prep ->
../../strc/ems_prep.pl
```

```
cd /work/course/usrNN/opt/uems/runs/bulgaria/
./ems_prep --dset gfs --length 24
```

горната команда ще подготви симулация за следващите 24 часа използвайки данните от GFS (Global Forecast System) който са публично достъпни онлайн. Командата ще сваля глобалните данните за следващите 24 часа от списък със сървъри които съдържат данните (www.nomads.ncep.noaa.gov). Други два важни параметъра са `--date YYYYMMDD` с помощта на който можете да укажете началната дата на симулацията и `--domains DOM1:SH1, DOM2:SH2`, например ако искате да моделирате метеорологично събитие в миналото, командата:

```
./ems_prep --dset gfs --date 20161015 --length 24 --domains 2
```

ще подготви симулация с начална дата 15/10/2016 с продължителност 24 часа за двата изчислителни домейна. Самият скрипт `ems_prep` последователно ще стартира инструмента `ungrib` който е част от `wrdda` който преобразува данните от `grib2` в `netcdf` формат:

```
/work/course/usrNN/opt/uems/bin/ungrib
```

След което ще стартира инструмента `metgrid` който подготвя граничните условия на симулацията в папката `/work/course/usrNN/opt/uems/runs/bulgaria/wpsprd/`

2.7 Стартиране на симулацията.

След успешно приключване на скрипта `ems_post` можем да стартираме самата симулация чрез `ems_run`. Пълен списък със възможните параметри на скрипта може да се получи с помощта на `--help`

```
./ems_run --help
```

В случая е достатъчно да се изпълни `./ems_run`. Подразбиране ако се стартира скрипта без никакви допълнителни аргументи ще се извърши симулация само за първият домейн. В случай че желаете да включите и останалите домейни и използвайте патаметъта `-d N`, където `N` е цяло число указващо броя на включените домейни. При изпълнението си скрипта отпечатва детайлна информация на всяка стъпка от симулацията. По същество скрипта изпълнява последователно прекомпилирани версии на програмите `real.exe` и `wrf.exe` който са част модела WRF версия 3.7.1 веднъж стартираната, еволюцията на симулацията може да бъде наблюдавана с помощта на командата:

```
tail -f /work/course/usrNN/opt/uems/runs/bulgaria/rsl.out.0000
```

Необходимо е да изчакате определен период от време до завържвашването на самата симулация през което време можете да се отдадете на четене на документация. След приключване на симулацията резултатите от пресмятанията се пазят в папката:

```
/work/course/usrNN/opt/uems/runs/bulgaria/wrfprd
```

2.8 Използване на глобална инсталация

За да се спестят ресурси и улесни поддръжката на версиите е добре да се използва глобална инсталация на UEMS модела която е споделена между повече от един потребителя. В този случай обаче е необходимо да се предефинират три работни папки:

- `UEMS_LOCAL` - коренът (`root`) на пътят до локалните папки `EMS_RUN` и `EMS_LOGS`
- `EMS_RUN=$UEMS_LOCAL/runs` - локалната папка в която се намират симулациите на потребителя
- `EMS_LOGS=$UEMS_LOCAL/logs` - пътят до локалната папка която съдържа логовете (информацията) от изпълнението на скриптовете и програмите

В този пример ще използваме глобалната инсталация на UEMS модела на изчислителният клъстер Nestum. За целта нека инициализираме работните променливи на средата, така че да използваме

```
mkdir -p /work/course/usrNN/uems_local
. /opt/lx/ems/15.58.5/uems/etc/EMS.profile
export UEMS_LOCAL=/work/course/usrNN/uems_local
export EMS_RUN=$UEMS_LOCAL/runs
export EMS_LOGS=$UEMS_LOCAL/logs
```

веднъж установили променливите на средата можем да повторим стъпките описани в точки 2.5, 2.6 и 2.7. Използването на глобална инсталация на UEMS е за предпочитане защото пести дисково пространство и улеснява обновяването и поддържането на софтуера.

3 Компилиране на модела WRFv3.7.1 от изходен код

Изходният код на WRF модела е публично достъпен на адрес след регистрация на страницата http://www2.mmm.ucar.edu/wrf/users/download/get_source.html. След като свалите изходният код можете да направите локална папка в която да разархивирате файла `WRFV3.7.1.TAR.gz`. За да спестим време в настоящият пример можем да копираме изходният код направо от файловата система с помощта на командите:

Nestum:

```
mkdir -p /work/course/usrNN/wrf/src
cd /work/course/usrNN/wrf/src
cp /work/course/examples/wrf/WRFV3.7.1.TAR.gz .
tar zxvf WRFV3.7.1.TAR.gz
```

3.1 Системни изисквания

Моделът WRF може да бъде компилиран върху множество архитектури като сериен или паралелен код. В този документ ще разгледаме примерния случай в който се компилира паралелна версия на кода за системи с разпределена памет (dm - distributed memory architecture) с помощта на OpenMPI среда с компилатори на Intel. Минималните изисквания необходими за компилирането на модела са:

- персонален компютър или изчислителен клъстер който работи под ОС Linux
- библиотека netcdf включващата и интерфейс за програмния език Fortran версия ≥ 3
- в случай че се компилира серийна версия. Компилатори на програмните езици C/C++ и Fortran, например: GNU компилатори `gcc/gfortran`, Intel компилатори `icc/ifort` или PGI компилатори `pgcc/pgf90`
- в случай че се компилира паралелна версия е необходим компилатор който поддържа стандарта OpenMP за паралелни архитектури със споделена памет (sm) и/или паралелен компилатор с поддръжка на библиотеката MPI за архитектури с разпределена памет.
- пакета `build-essential` на Linux който включва инструмента `make`

в случай че компилирате WRF модела на собственото си PC може да инсталирате необходимият софтуер чрез пакетния мениджър на операционната система. Например ако използвате някоя Debian базирана дистрибуция като Debian или Ubuntu е необходимо да инсталирате следният списък с пакети:

```
sudo apt-get update
sudo apt-get install gcc gfortran build-essential libnetcdf-dev libopenmpi-dev
openmpi-common openmpi-bin
```

ако компилирате модела WRF изчислителния клъстер Nestum или Physon. необходимите библиотеки са достъпни с помощта на системата `modules`. Необходимо е да се заредят следните модули в интерактивна сесия.

Nestum

```
srunc -p interact.p -n 2 --pty bash
module load netcdf
module load netcdf-fortran
module load openmpi
```

Physon

```
qlogin -lh_rt=8:0:0 -pe ipar 4
module load netcdf
module load netcdf-fortran
module load openmpi/1.8.3-intel-13.1
```

3.2 Компилиране на модела

Веднъж след като изходният код на модела бъде разархивиран и помощните софтуерни пакети са заредени можем да преминем към компилирането на модела. За целта можем да сменим текущата папка с работната директория в която се намира изходният код и да стартираме конфигурационният скрипт `configure` преди това обаче трябва да се настроят няколко променливи на средата:

Nestum:

```
cd /work/course/usrNN/wrf/src
cd WRFV3
export NETCDF=/opt/lx/u16.04/netcdf-fortran/4.4.4/gcc-os
export NETCDF4=1
export WRFIO_NCD_LARGE_FILE_SUPPORT=1
./configure
```

Physon:

```
cd /work/course/usrNN/wrf/src
cd WRF
export NETCDF=
export NETCDF4=1
export WRFIO_NCD_LARGE_FILE_SUPPORT=1
./configure
```

конфигурационният скрипт ще попита първо коя от предефинираните среди за компилиране избираме. В нашият случай избираме отговор 15 която отговаря на паралелна среда с разпределена памет (`dmpar` - distributed memory) и компилатори на Intel. В случай че компилирате кода на друга паралелна платформа например суперкомпютър с математични ускорители Intel Phi или домашно PC трябва да подберете подходящата за вас среда за компилиране. Параметъра `NETCDF4=1` разрешава поддръжка на версия 4 на библиотеката `netcdf`. Докато параметъра `WRFIO_NCD_LARGE_FILE_SUPPORT=1` разрешава поддръжката на големи файлове за библиотеката `netcdf`. Следващият въпрос:

Compile for nesting? (1=basic, 2=preset moves, 3=vortex following) [default 1]:

обичайно се отговоря подразбиране с 1. Самото компилиране се извършва със скрипта `compile` който приема като параметър името на модула който да бъде компилиран в нашият случай това е който включва в себе си `em_real` изпълнимите файлове: `wrf.exe`, `real.exe`, `ndown.exe` и `tc.exe`. В случай че разполагаме със ситема която притежава повече от един процесор/ядро можем да ускорим компилирането на модела с помощта на параметъра `-j n`, където `n` е броят на процесорните ядра които да се използват за компилиране на модела (подразбиране `n = 2`). Окончателно командата за компилиране на модела изглежда така:

```
./compile -j n em_real
```

забележка: поради особеност на скрипта, компилирането на `em_real` модела трябва да се извърши два пъти последователно. В случай на нужда да се изчисти работната

директория от компилираната версия (това се прави задължително в случай на преконфигурация) трябва да се изпълни скрипта `clean`

```
./clean
```

генерираните изпълними файлове след успешна компилация могат да се намерят в поддиректорията `main`:

```
./main/real.exe  
./main/wrf.exe  
./main/ndown.exe  
./main/tc.exe
```

нека копираме изпълнимите файлове в подпапка в която се намират нашите локални UEMS данни:

```
mkdir -p $UEMS_LOCAL/bin  
cp ./main/real.exe/ $UEMS_LOCAL/bin/real_arw.exe  
cp ./main/wrf.exe/ $UEMS_LOCAL/bin/wrfm_arw.exe
```

и настроим нова променлива на средата `WRF_HOME` така че:

```
export WRF_HOME=$UEMS_LOCAL
```

3.3 Интеграция на прекомпилираната версия на модела WRF с UEMS

Удобно е когато интегрирате вашата прекомпилирана версия на WRF модела със скриптовете на софтуерният пакет UEMS за да се възползват от възможността да създават изчислителни области с помощта на `dwiz` и стартират симулации на прогнози за определен интервал време с помощта на скриптовете `ems_port` и `ems_run`. Интегрирането се налага и в случаите когато искате да изпълнявате модела на повече от един изчислителен възел на паралелният клъстер. В случая ще демонстрираме как трябва да се модифицират скриптовете на UEMS за да се използва външно компилиран WRF модел. Все пак трябва да се отбележи че тези промени се правят за собствена сметка и нямаме гаранция че скрипта винаги ще работи коректно. За целта е необходимо да се редактира съдържанието на скрипта `WRFRun.pm` който се намира в:

```
$EMS_ROOT/strc/EMSrun/Models/WRF/WRFRun.pm
```

за целта нека първо направим резервно копие на оригиналния скрипт и модифицираме актуалния файл:

```
cp -a $EMS_ROOT/strc/EMSrun/Models/WRF/WRFRun.pm $EMS_ROOT/  
strc/EMSrun/Models/WRF/WRFRun.pm_orig  
vi $EMS_ROOT/strc/EMSrun/Models/WRF/WRFRun.pm
```

Необходимо е да се добавят два реда на редове 122, 248, 437 и 512:

ред 122 името на изпълнимият файл `real_arw.exe` се присвоява в променливата `$exename`

```
(my $exe = "$ENV{EMS_BIN}/${proc}_CORE.exe") =~ s/CORE/$core/g;  
(my $exename = "${proc}_CORE.exe") =~ s/CORE/$core/g;  
return 1 if &Esystem::checkBin($exe);
```

ред 248 в случай че променливата на средата `WRF_HOME` е дефинирана променяме командата за изпълнение на `real_arw.exe` с `srun $ENV{WRF_HOME}/bin/$exename`

```
my $command = "$mpixec_$exe";  
if (defined $ENV{WRF_HOME} and $ENV{WRF_HOME}) {  
    $command = "srun $ENV{WRF_HOME}/bin/$exename";  
}
```

ред 437 името на изпълнимият файл `wrf_arw.exe` се присвоява в променливата `$exename`

```
(my $exe = "$ENV{EMS_BIN}/${proc}_CORE.exe") =~ s/CORE/$core/g;  
(my $exename = "${proc}_CORE.exe") =~ s/CORE/$core/g;  
return 1 if &Esystem::checkBin($exe);
```

ред 512 в случай че променливата на средата `WRF_HOME` е дефинирана променяме командата за изпълнение на `wrfm_arw.exe` с `srun $ENV{WRF_HOME}/bin/$exename`

```
my $command = "$mpixexec_$exe";  
if (defined $ENV{WRF_HOME} and $ENV{WRF_HOME}) {  
    $command = "srun $ENV{WRF_HOME}/bin/$exename";  
}
```

за да опростим опростим процедурата по промяна на файла `WRFrunc.pm` ще използваме файл заготовка като първо направим копие на оригиналния файл и копираме новият на негово място:

```
cd $EMS_ROOT/src/EMSRUN/Models/WRF/  
mv WRFrunc.pm WRFrunc.pm_orig  
cp /work/course/examples/uems/scripts/WRFrunc.pm .
```

сега сме готови да продължим с играта като стартираме симулацията с външно компилираните версии и апрограмите `real_arw.exe` и `wrfm_arw.exe`

```
srun -n 32 --pty bash  
. $EMS_ROOT/etc/EMS.profile  
export WRF_HOME=$UEMS_LOCAL  
module load openmpi  
module load netcdf  
module load netcdf-fortran  
cd $UEMS_ROOT/runs/bulgaria  
./ems_prep --dset gfs --length 24 --domains 2  
./ems_run --domains 2
```

забележка: Файлът `WRFrunc.pm` ще бъде презаписан при всяко обновяване или преинсталация на UEMS пакета. Затова е добра практика да се пази резервно копие на файла.
забележка: Добра практика е да се копират оригиналните TBL файлове от дистрибуцията на WRF в папката `$EMS_ROOT/data/tables/wrf`

```
cd $UEMS_ROOT/data/tables/wrf/physics/lsm/  
mv VEGPARAM.TBL VEGPARAM.TBL_bck  
mv URBPARAM.TBL URBPARAM.TBL_bck  
mv MPTABLE.TBL MPTABLE.TBL_bck  
mv URBPARAM_UZE.TBL URBPARAM_UZE.TBL_bck  
mv grib2map.tbl grib2map.tbl_bck  
mv LANDUSE.TBL LANDUSE.TBL_bck  
mv SOILPARAM.TBL SOILPARAM.TBL_bck  
mv GENPARAM.TBL GENPARAM.TBL_bck  
  
cp /work/course/usrNN/wrf/src/run/VEGPARM.TBL VEGPARAM.TBL  
cp /work/course/usrNN/wrf/src/run/URBPARAM.TBL URBPARAM.TBL  
cp /work/course/usrNN/wrf/src/run/MPTABLE.TBL MPTABLE.TBL  
cp /work/course/usrNN/wrf/src/run/URBPARAM_UZE.TBL URBPARAM_UZE.  
TBL  
cp /work/course/usrNN/wrf/src/run/grib2map.tbl grib2map.tbl  
cp /work/course/usrNN/wrf/src/run/VEGPARM.TBL VEGPARAM.TBL  
cp /work/course/usrNN/wrf/src/run/SOILPARAM.TBL SOILPARAM.TBL  
cp /work/course/usrNN/wrf/src/run/GENPARAM.TBL GENPARAM.TBL
```

4 Визуализация и анализ на резултатите от симулациите

4.1 Бързо разглеждане на изходните данни в netcdf формат

Един полезен инструмент който позволява бързо разглеждане на входните данни както и експортирането на някои от полетата в текстови формат е `ncdump` той е дъстепен през модула `netcdf`. Например можем да разгледаме хедър на даден `netcdf` файл в който са записани глобалните параметри на симулация както и описание на метео полетата от `wrf` модела:

```
module load netcdf
ncdump -h $UEMS_ROOT/runs/bulgaria/wrfprd/wrfout_d02_YYYY-MM-
DD_hh:00:00 | less
```

4.2 Инструмент за последващата обработка на данните ARWpost и визуализация с GrADS

Инструмента `ARWpost` е публично достъпен от страницата на NCAR http://www2.mmm.ucar.edu/wrf/users/download/get_sources.html#post_processing след регистрация. В случая ще използваме копие на архива който се намира в папката `/work/course/examples/arwpost/ARWpost_V3.tar.gz`. По-долу е описана процедурата по компилиране:

```
mkdir -p /work/course/usrNN/arwpost/src
cd /work/course/usrNN/arwpost/src
cp /work/course/examples/arwpost/ARWpost_V3.tar.gz .
tar zxvf ARWpost_V3.tar.gz
cd ARWpost/
module load netcdf
module load netcdf-fortran
module load intel
export NETCDF=/opt/lx/u16.04/netcdf-fortran/4.4.4/gcc-os
./configure
```

необходимо е да се редактира файла `src/Makefile` на ред 19 така че да се смени рефренцията от `C` версията на `-lnetcdf` библиотеката към версията на Fortran `-lnetcdf`

```
-L$(NETCDF)/lib -I$(NETCDF)/include -lnetcdf
```

след което можем да продължим с компилирането като изпълним командата:

```
./compile
```

след успешно компилиране можем да копираме файла в `$WRF_HOME/bin`

```
cp src/ARWpost.exe $WRF_HOME/bin
```

следващата стъпка е да се направи локално копие на `namelist.ARWpost` и да се променят параметрите `start_date`, `end_date`, `input_root_name`, `output_root_name`, `output_type` и `fields`:

```
...
start_date = 'YYYY-MM-DD_hh:00:00',
end_date = 'YYYY-MM-DD_hh:00:00',
...
input_root_name = 'wrfprd/wrfout_d02'
output_root_name = 'extract'
...
fields = 'height,geopt,theta,tc,tk,td,td2,rh,rh2,umet,vmet,pressure,u10m,v10m,
wdir,wspd,wd10,ws10,slp,mcape,mcin,lcl,lfc,cape,cin,dbz,max_dbz,clfr'
```

```
output_type = 'grads'
...
```

и да се изпълни командата:

```
$WRF_HOME/bin/ARWpost.exe
```

резултатът от изпълнението ще бъдат два файла `extract.ctl` и `extract.dat` които могат да бъдат визуализирани с помощта на GrADS.

4.3 Визуализация с помощта на скрипт на python

Скриптовият език `python` има възможност да чете файлове в `netcdf` формат в комбинацията с `matplotlib` и `cartopy` позволяват бърза визуализация на данните. Скрипта визуализира няколко от изходните полета на WRF модела:

- `Temp` - температурно поле `T2`
- `Rain` - разликата в натрупаният валеж от полетата `RAINNC+RAINNC+RAINSH` между два последователни `nc` файла
- `Hail` - разликата в натрупаният “твърд” валеж от полетата `HAILNC+GRAUPELNC` между два последователни `nc` файла
- `Humid` - Относителна влажност на приземният слой

$$\begin{aligned}a_2 &= 17.2693882 \\a_3 &= 273.16 \\a_4 &= 35.86 \\svp &= 611 \times \frac{\exp(a_2 * (T2 - a_3))}{(T2 - a_4)} \\Humid &= Q2 \times (PSFC - svp)/(0.622 \times svp)\end{aligned}$$

- `Cloud` - не е имплементиран все още
- `Solar` - полето `SWDOWN`

Можем да копираме скрипта от работната папка `/work/course/examples/python/`, след като го разархивираме скрипта може да се изпълни с командата `python field2plot-0.3.py`. Той приема няколко входни параметъра:

- `-h` - отпечатва помощно меню
- `-b` - пътят до файловата система където се намират `wrfout_` изходните файлове на WRF модела
- `-p` - префикс на изходните файлове на модела (подразбиране `wrfout_d02`)
- `-f` - име на полето за визуализация (`Temp`, `Rain`, `Hail`, `Humid`, `Cloud`, `Solar`)

по-долу можете да използвате следните команди за разархивиране и генериране на изображения на температурното поле `T2`:

```
cd ~/work/python
cp /work/course/examples/python/field2plot.tar.gz .
tar zxvf field2plot.tar.gz
python field2plot-0.3.py -h
python field2plot-0.3.py -b /work/course/usrNN/opt/uems/runs/bulgaria/
    wrfprd -f Temp
```