LANCIS | Laboratorio Nacional de Ciencias de la Sostenibilidad

Manual de instalación y uso del paquete megadaptr(R)

Fidel Serrano Candela

diciembre 2021

Índice

1. En clusters de super cómputo

1.1. Requerimientos en el cluster

Para correr experimentos aprovechando la capacidad de cómputo en paralelo que ofrecen los clusters de super cómputo, se requieren únicamente dos dependencias: Singularity y Postgres. Singularity es una plataforma de contenedores que permite crear y ejecutar contenedores que empaquetan piezas de software de manera portátil y reproducible. El contenedor es un solo archivo, y elimina la necesidad de instalar todo el software que necesita en cada sistema operativo diferente. Mientras que Postgres es un poderoso sistema de base de datos relacional de objetos de código abierto con más de 30 años de desarrollo activo que le ha ganado una sólida reputación por su confiabilidad, solidez de funciones y rendimiento.

1.2. Configuración de experimentos

Para correr el modelo megadaptr en un cluster de super cómputo el primer paso es clonar el repositorio en la computadora personal que se conectará de forma remota al cluster. El repositorio del modelo está disponible en la siguiente url: https://github.com/comses/megadapt. Para clonar el repositorio se puede usar una herramienta especializada en conexión con repositorios y control de versiones llamada GitHub Desktop que está disponible en el siguiente url: https://desktop.github.com/ o desde la terminal con el comando:

```
git clone https://github.com/comses/megadapt
```

1.2.1. Conexión con el cluster

Para configurar y correr experimentos de megadaptr en un cluster de super cómputo se requiere poder copiar archivos en el cluster y poder usar la terminal del cluster de forma remota. Si la computadora remota tiene cualquier distribución de Linux o MacOS como su sistema operativo, estos dos requerimientos se logran directamente desde la terminal de la computadora remota, sin embargo si la computadora remota usa el sistema operativo Windows, se requiere instalar dos aplicaciones: Putty y WinSCP. Ambas aplicaciones se pueden descargar e instalar de forma gratuita en los siguientes urls:

- Putty - https://www.putty.org/
- WinSCP - https://winscp.net/eng/download.php

1.2.2. Estructura de carpetas

Para correr un conjunto de simulaciones dada una versión del modelo se requieren los siguientes archivos:

- db-postgres.json
- megadapt.sif

- megadapt_wgs84_v16.gpkg
- grid.json

Por razones de reproducibilidad es conveniente poner estos 4 archivos en una carpeta con el nombre del experimento a desarrollar. El archivo db-postgres.json contiene la información necesaria para que las simulaciones almacenen sus resultados en la base de datos.

```
{
    "driver": "postgres",
    "name": "megadapt",
    "host": "tawa",
    "port": 5432,
    "user": "fidel"
}
```

El archivo megadapt.sif es una imagen de un sistema operativo con todos los componentes necesarios para correr el modelo megadaptr y se genera con los siguientes comandos:

```
cd deploy
singularity build --fakeroot megadapt-base.sif megadapt-base.def
singularity build --fakeroot megadapt.sif megadapt.def
```

El archivo megadapt_wgs84_v16.gpkg es una capa geográfica en formato geopackage, que contiene los polígonos sobre los que correrá el modelo, así como los datos iniciales para estos polígonos. Por último el archivo grid.json contiene la información necesaria para generar el espacio de parámetros en el que se correrá el modelo, es decir los rangos de valores que se asignarán a cada parámetro del modelo.

1.2.3. Configuración paramétrica

La configuración del espacio de parámetros en los que se correrán las simulaciones de un experimento, así como el nombre del experimento, se deben especificar en el archivo grid.json. A continuación se muestra un ejemplo:

```
{
    "name": "test",
    "title": "test",
    "description": "stressing the model base with suitability normalized
    for all climate and urban growth scenarios, competing between flooding
     and scarcity",
    "author_name": "Fidel Serrano",
    "year": 2020,
    "strategy": "cartesian",
    "overrides": {
        "sacmex": {
```

```
            "distance_mode": "normalized",
            "constructor": ["system_budget","action_budget"],
            "budget": [1000],
            "maintenance_effectiveness_rate": [0.2],
            "new_infrastructure_effectiveness_rate": [0.1],
            "infrastructure_decay_rate": [0.02]
        },
        "climate": {
            "id": [1,2,3,4,5,6,7,8,9,10,11,12]
        },
        "mental_models": {
            "strategy": ["constant"]
        }
    },
    "n_reps": 1,
    "n_steps": 40
}
```

Dentro de estas definiciones, cuando se asigna una lista a un parámetro por ejemplo en el caso del 'id' dentro de 'climate', significa que se correrán simulaciones asignando el valor de cada uno de los elementos de esta lista al parámetro en cuestión.

1.2.4. Cómputo en paralelo usando el espacio de parámetros

Una vez creada la carpeta en la computadora personal con los archivos necesarios descritos anteriormente, se procede a copiar esta carpeta al cluster de super cómputo. En el caso de que el sistema operativo de la computadora personal sea Linux o MacOS esto se logra directamente desde la terminal con el siguiente comando:

```
scp -r ./test/ fidel@patung.mine.nu:~/megadapt_runs
```

En el caso de que la computadora personal tenga el sistema operativo Windows, esto se realiza a través de la aplicación WinSCP, previamente instalada. Una vez copiada la carpeta al cluster, hay que conectarse al cluster de manera remota para correr comandos en la terminal del cluster. En el caso de que el sistema operativo de la computadora personal sea Linux o MacOS esto se logra con el siguiente comando:

```
ssh patung.mine.nu
```

En el caso de que la computadora personal tenga el sistema operativo Windows, esto se realiza a través de la aplicación Putty, previamente instalada. Una vez conectado remotamente a la terminal del cluster, se corre el siguiente comando, que usa los 4 archivos previamente preparados en la computadora remota y copiados al cluster:

```
./megadapt.sif --db-config db-postgres.json grid setup
--experiment-config  grid.json --study-area megadapt_wgs84_v12.gpkg
```

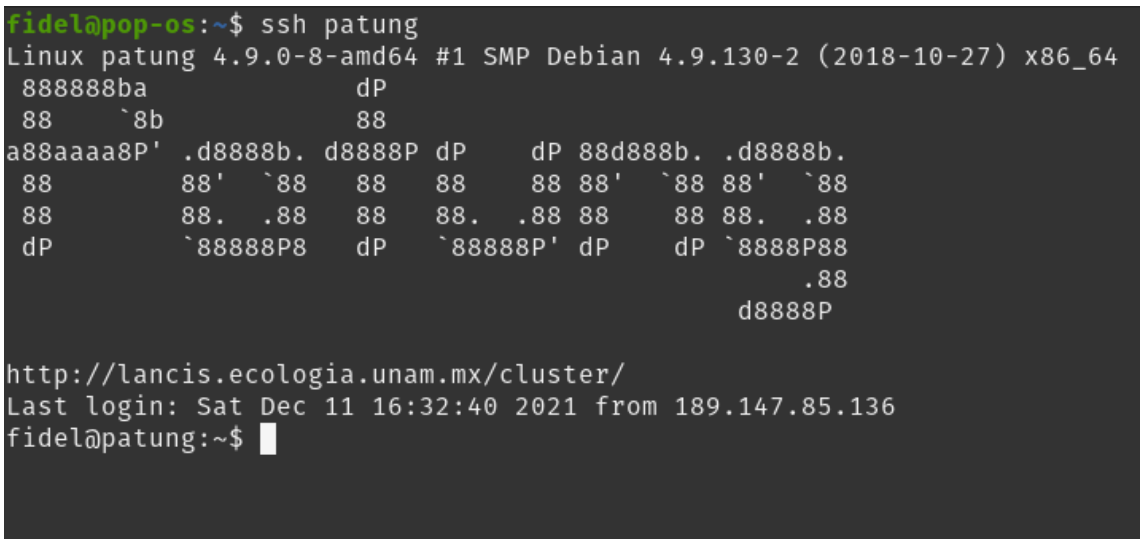

Figura 1: Terminal remota de patung

El resultado será la creación de una tabla de parámetros en la base de datos, cuyo nombre será el nombre del experimento definido en archivo grid.json con el sufijo _param, en el caso del ejemplo descrito anteriormente sería test_param. Esta tabla contiene todas las combinaciones de parámetros que se correrán en paralelo generando tantas simulaciones como la cantidad de estas combinaciones, multiplicada por la cantidad de repeticiones (n_rep). Adicionalmente en este paso se genera el script de HTCondor para correr las simulaciones en paralelo. Este script tendrá el nombre del experimento definido en el archivo grid.json, con la extensión .sub. En el caso del ejemplo descrito anteriormente sería test.sub. Para correr el script de HTCondor recién generado simplemente se ejecuta el siguiente comando:

```
condor_submit test.sub
```

El resultado de este comando será que HTCondor, repartirá a los nodos disponibles del cluster las distintas simulaciones que forman parte de este experimento y cada una de ellas irá guardando los resultados en la base de datos, quedando relacionados con el identificador de la combinación específica de parámetros correspondiente en la tabla de parámetros test_param. Para cerciorarse de que el cluster concluyó con las simulaciones del experimento se puede correr el siguiente comando en la terminal del cluster:

```
watch condor_q
```

1.3. Consulta de resultados

Para consultar los resultados almacenados en la base de datos, de uno o varios experimentos se puede hacer una consulta a la base de datos. Para hacer la consulta es necesario entender la estructura de la base y configurar un puente desde la computadora personal hasta la base de datos a través del cluster.

1.3.1. Estructura de la base de datos

Como se describió anteriormente al correr las simulaciones de un experimento se generan dos tablas relacionadas entre si en la base de datos, una con los parámetros

(test_param) en donde cada renglón corresponde a una de las combinaciones de parámetros y otra de resultados del modelo (test_result) que contiene los valores de las variables del modelo para cada unidad espacial del geopackage inicial y para cada año de simulación. Las variables que son almacenadas en la tabla de resultados son:

```
"censusblock_id",
"runoff_presence",
"sewer_infrastructure_age",
"potable_water_infrastructure_age",
"resident_reports_ponding_count_mean",
"resident_reports_flooding_count_mean",
"sewer_system_storm_drain_count",
"sewer_system_capacity",
"household_potable_system_lacking_percent",
"household_sewer_system_lacking_percent",
"household_days_no_potable_water_per_week_mean",
"resident_income_per_capita",
"resident_asset_index",
"flooding_event_count",
"flooding_index",
"ponding_event_count",
"ponding_index",
"scarcity_index_exposure",
"scarcity_index_sensitivity",
"household_potable_water_sensitivity",
"household_water_storage_tank_percent",
"household_sewer_sensitivity",
"household_resilience",
"household_potable_water_vulnerability",
"household_sewer_vulnerability",
"household_sewer_intervention_count",
"household_potable_water_invention_count",
"sacmex_potable_maintenance_intervention_presence",
"sacmex_sewer_maintenance_intervention_presence",
"sacmex_potable_new_infrastructure_intervention_presence",
"sacmex_sewer_new_infrastructure_intervention_presence",
"non_potable_maintenance",
"non_potable_new_infrastructure",
"potable_maintenance",
"potable_new_infrastructure"
```

Esto es importante porque esas serán las variables que se puedan consultar, con un query a la base de datos, una vez concluidas las simulaciones.

1.3.2. Conexión la base de datos

La base de datos con los resultados de los experimentos en el cluster de supercómputo está instalada en uno de los nodos del cluster que se llama tawa.mine.nu

que por razones de seguridad no es accesible desde Internet, por lo que para conectarse con la base de datos de manera remota desde una computadora personal, es necesario hacer un puente que comunique la computadora personal con tawa a través de patung. Para hacer este puente se requiere editar o crear el archivo ~/.ssh/config en la computadora personal, con el siguiente contenido:

```
Host patung
    Hostname patung.lancis.ecologia.unam.mx

Host tawa
    HostName tawa
    ProxyJump patung
    ForwardAgent yes
```

Una vez hecho esto, hay que crear un túnel entre un puerto local en la computadora personal que se redirija al puerto de la base de datos en tawa, eso se logra con el siguiente comando:

```
ssh -L 2222:localhost:5432 tawa
```

1.3.3. Consulta desde R

Con el puente establecido, se pueden hacer consultas en el lenguaje R, usando la bibliotecas RPostgreSQL y DBI, de la siguiente manera:

```
library("RPostgreSQL")
library("DBI")
drv <- dbDriver("PostgreSQL")
conn <- dbConnect(drv, dbname = "megadapt",
port=2222,
user="fidel",
host="localhost")
query <- "select censusblock_id,
        avg(scarcity_index_exposure) as scarcity_exposure
        from test_result
        inner join test_param as p on p.id = test_result.param_id
        where test_result.year = 2060
        and p.climate__id = 1
        and p.sacmex__budget = 1000
        and p.sacmex__constructor = 'action_budget'
        group by censusblock_id;"

df <- dbGetQuery(conn,query)
write.csv(df,"test_2060.csv", row.names = FALSE)
```

Este ejemplo de consulta obtiene el promedio por AGEB de la exposición a escasez para el año 2060, para el presupuesto con valor de 1000 y escribe un csv con los resultados.

1.4. Representación geográfica

Por razones de optimización de espacio de almacenamiento, la base de datos de resultados no contiene la geometría de las unidades espaciales, por lo que para

generar una representación geográfica de los resultados es necesario unir la tabla resultante con la capa geográfica usada en la inicialización del modelo. Esto se puede hacer fácilmente en QGIS mediante la unión por el campo censusblock_id.

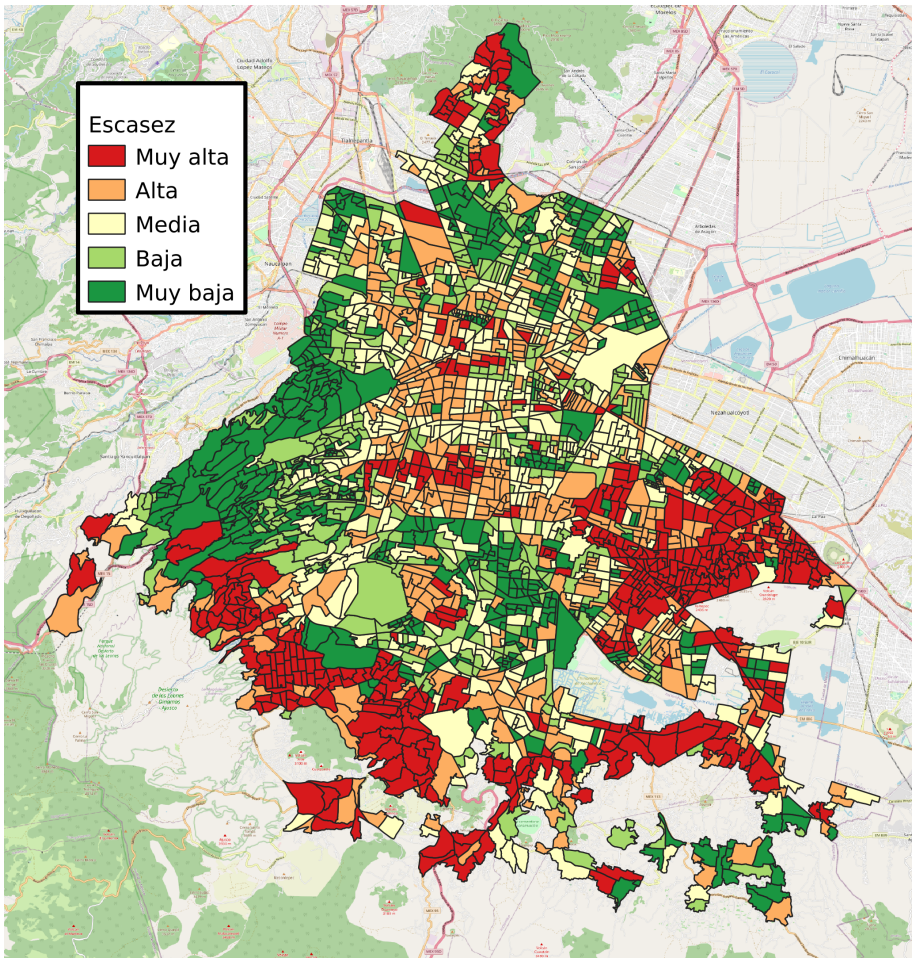

Figura 2: Representación espacial de la consulta

2. En computadoras personales

2.1. Instalación del paquete megadaptr(R)

Para correr el modelo megadaptr en una computadora personal el primer paso es clonar el repositorio del modelo. El repositorio del modelo está disponible en la siguiente url: https://github.com/comses/megadapt. Para clonar el repositorio se puede usar una herramienta especializada en conexión con repositorios y control de versiones llamada GitHub Desktop que está disponible en el siguiente url: https://desktop.github.com/ o desde la terminal con el comando:

```
git clone https://github.com/comses/megadapt
```

La forma más sencilla de correr el modelo es abrir el proyecto de R del modelo en RStudio. RStudio es un conjunto de herramientas integradas diseñadas para potenciar el trabajo con scripts de R y Python. Incluye una consola, un editor de resaltado de sintaxis que admite la ejecución directa de código y una variedad de herramientas sólidas para trazar, ver el historial, depurar y administrar el

espacio de trabajo. RStudio está disponibla para su descarga en el siguiente url: https://www.rstudio.com/products/rstudio/download/. Para abrir el proyecto del modelo se abre el archivo /src/r/megadaptr.Rproj en RStudio. Una vez abierto el proyecto, aparecerá en la parte superior derecha de RStudio la palabra megadaptr, simbolizando que el proyecto está abierto. Para compilar el paquete de R que permite correr el modelo se hace click en el menú Build/More/Clean and Rebuild como se muestra en la figura 3.

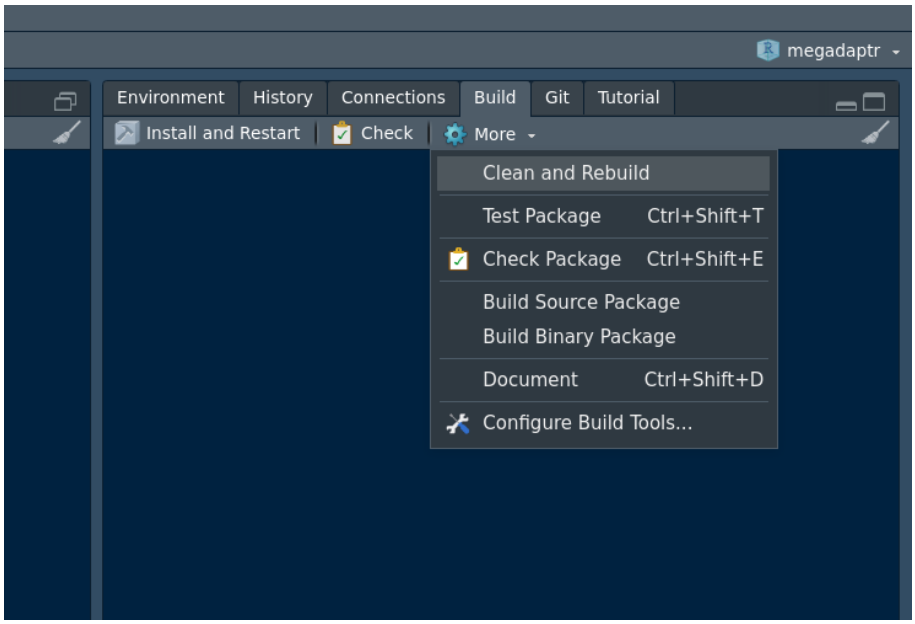

Figura 3: Compliación del paquete de R que contiene el modelo megadaptr

2.2. Configuración de los parámetros

Para correr una simulación del modelo megadaptr con un conjunto específico de parámetros, se puede editar y correr el archivo /src/r/scenarios/main.R para especificar los parámetros y componentes de la simulación. En el siguiente ejemplo se correrá la simulación utilizando las funciones de valor que emulan una reducción del caudal recibido por la CDMX del sistema Cutzamala y que se encuentran en la carpeta reduc_cutza, empezando en el año 2020 y corriendo el modelo por 20 años.

```
library(megadaptr)
library(magrittr)
source('../scenarios/util.R')

overrides = list(
        sacmex = list(distance_mode = "normalized"),
        value_functions = list(function_folder = "reduc_cutza"))
megadapt <- megadaptr:::megadapt_deserialize(
        config = overrides,
        study_area_path = megadaptr:::data_dir('censusblocks',
         'megadapt_wgs84_v16.gpkg'),
        year = 2020,
        n_steps = 20
)

result_config <- megadaptr:::megadapt_config_create(overrides)
```

```
megadaptr:::config_flatten(result_config)
new_results <- simulate(megadapt)
write.csv(new_results,"test_local_2040.csv", row.names = FALSE)
```

El resultado de este código será un archivo csv con los resultados para el último año de la simulación. Cabe mencionar que esta forma de correr el modelo tiene varias limitaciones, por ejemplo a diferencia de los experimentos corridos en el cluster solo se puede correr con una combinación de parámetros a la vez y no se almacenan los años intermedios de la simulación.

2.3. Representación geográfica

De forma similar que los resultados obtenidos de simulaciones en el cluster, el archivo csv del ejemplo anterior no contiene la geometría de las unidades geográficas del modelo. Por lo que para generar una representación geográfica es necesario unir estos resultados con la capa geográfica con la que se inicializó el modelo. Esto se puede realizar en QGIS importando tanto la capa inicial como el csv con el resultado de la simulación para último año, y aplicando un join usando el campo censusblock_id.