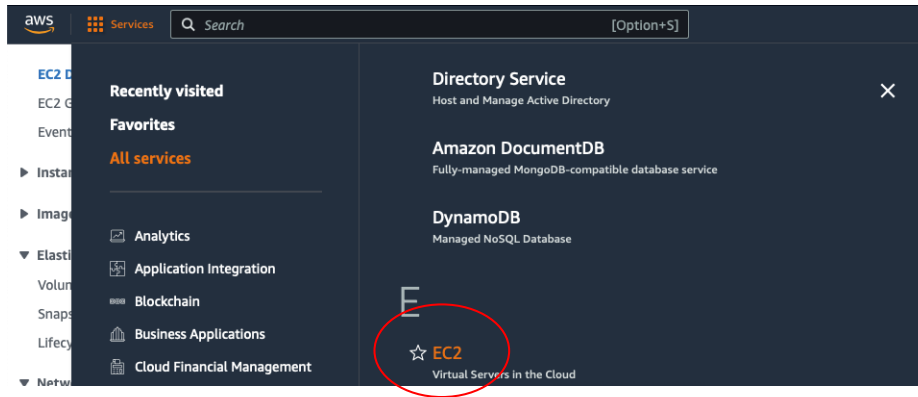


CPU Compile on AWS

Programming FPGAs for Economics:
An Introduction to Electrical Engineering Economics

Bhagath Cheela, Alessandro Peri, André DeHon, Jesús Fernández-Villaverde

Steps




1. **Log into your AWS account:**
2. Navigate to the Home Console
3. Select **EC2**
4. Launch Instance

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch instance



Migrate a server 

Steps: Name and tags

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name

[Add additional tags](#)

Select Amazon Linux Instance

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Q Search our full catalog including 1000s of application and OS images

Recents

Quick Start

Amazon Linux

aws

macOS

Mac

Ubuntu

ubuntu®

Windows

Microsoft

Red Hat

Red Hat

SUSE LI

SUS

Q

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI

ami-0eb5115914ccc4bc2 (64-bit (x86), uefi-preferred) / ami-0ca6dd60dbe07d5ad (64-bit (Arm), uefi)

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible ▼

Description

Amazon Linux 2023 AMI 2023.3.20240304.0 x86_64 HVM kernel-6.1

Architecture

64-bit (x86) ▼

Boot mode

uefi-preferred

AMI ID

ami-0eb5115914ccc4bc2

Verified provider

Select CPU Instance

▼ Instance type [Info](#) | [Get advice](#)

Instance type

m5n.large

Family: m5n 2 vCPU 8 GiB Memory Current generation: true

On-Demand RHEL base pricing: 0.179 USD per Hour

On-Demand Linux base pricing: 0.119 USD per Hour

On-Demand Windows base pricing: 0.211 USD per Hour

☒ All generations

[Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

Create all binaries from an **m5n.large** instance

Key pair

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

Proceed without a key pair (Not recommended)

Default value ▼



[Create new key pair](#)

For information on how to create a new key pair go [here](#)

Launch m5n.large Instance

Repeat the same for all other CPU instances

▼ Summary

Number of instances

Info

1

Software Image (AMI)

FPGA Developer AMI

ami-02ab431c7b3297b00

Virtual server type (instance type)

z1d.2xlarge

Firewall (security group)

New security group

Storage (volumes)

2 volume(s) - 125 GiB















Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel

Launch Instance

Review commands

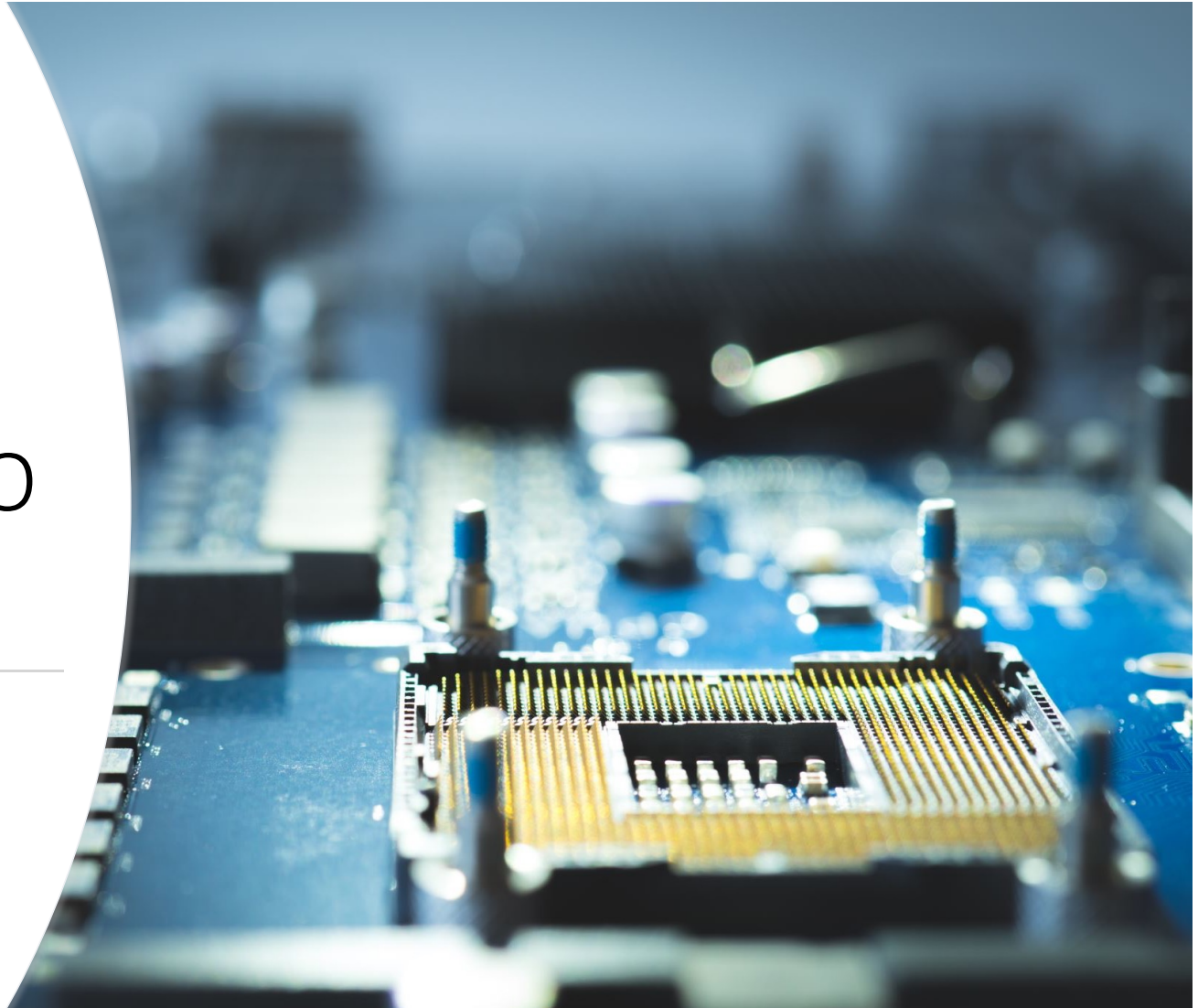
EC2 Instances

Instances (1/2) Info										Connect
<input type="text" value="Find Instance by attribute or tag (case-sensitive)"/>									Running ▼	
<input type="checkbox"/>	Name 	Instance ID	Instance state 	Instance type 	Status check	Alarm status	Availability Zone 	Public IPv4 DNS 		
<input type="checkbox"/>	3k-100-4	i-02362075ede6cd75e	 Running  	z1d.2xlarge	 2/2 checks passed	View alarms +	us-west-2a	ec2-54-188-219-255.us...		
<input checked="" type="checkbox"/>	cpu-run	i-00757b5113b27214b	 Running  	m5n.large	 Initializing	View alarms +	us-west-2a	ec2-35-88-29-46.us-we...		

- In the top-left menu, select 'Instances'
- Copy the public IPv4 address in Visual Studio code

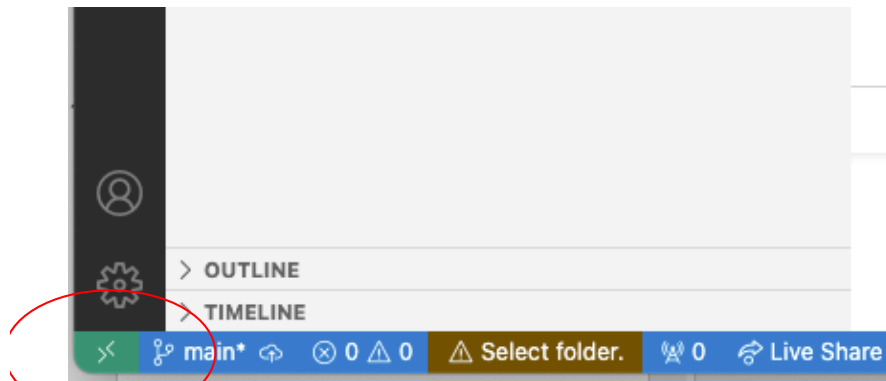


VISUAL STUDIO CODE



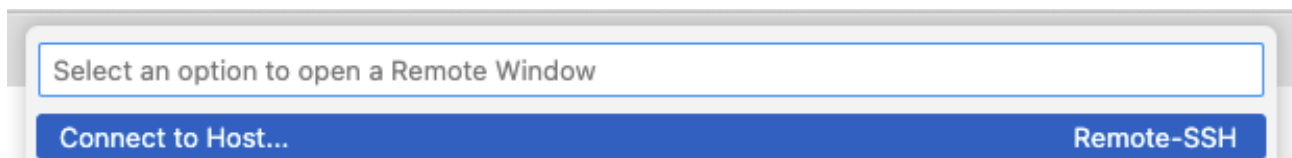
Open a Remote Window

- On the bottom-left corner of Visual Studio Code click on the green button 'Open a Remote Window'
- Click on Connect to Host
- Click on Configure SSH Hosts.
 - Give a name to the Host (e.g. aws-ec-cpu)
 - Copy the public IP address (red circle) and set the path to your key
 - Set the user to **ec2-user**
- Connect to aws-ec2.



CLICK the GREEN BUTTON HERE

```
# ----- CONNECT TO AMAZON AWS
Host aws-ec-cpu
HostName ec2-34-219-191-114.us-west-2.compute.amazonaws.com
User ec2-user
IdentityFile ~/[YOUR KEY PATH].pem
# -----
```



Compile the Binaries



Install utilities and Copy Repo

- Initiate a terminal session on the AWS instance and run the subsequent script to install the utilities

```
sudo yum install git -y  
sudo yum install make -y  
sudo yum install tmux -y  
sudo yum install wget -y
```

- Clone our GitHub repository into a directory of your preference (e.g., /home/ec2-user):

```
git clone https://github.com/AleP83/FPGA-Econ.git
```

AWS Configure

```
[centos@ip-10-0-1-68 ~]$ aws configure  
AWS Access Key ID [None]: █
```

1. Go to your aws account and set (one time thing):
 - AWS Access Key ID
 - AWS Secret Access Key
2. Go to the terminal in visual studio and type aws configure

```
aws configure  
AWS Access Key ID [*****xxxx]: <Your AWS Access Key ID>  
AWS Secret Access Key [*****xxxx]: <Your AWS Secret Access Key>  
Default region name: [us-west-1]: us-east-1  
Default output format [None]: json
```

3. Set:
 - AWS Access Key ID:
 - AWS Secret Access Key
 - Default Region name: us-west-2
Note: this depends in which region you launched your instance.
 - Default output format [json]: json

Install OpenMPI and set the Environment

- Install OpenMPI. Then, run the following script from the terminal

```
sh code/common/util/OpenMPI_install.sh
```

- To compile binaries for parallel execution set the OpenMPI environment by executing the following commands in the terminal from the parent directory:

```
export PATH=$PATH:$HOME/openmpi/bin  
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$HOME/openmpi/lib
```

Modify the Makefile

- Set the AWS S3 Bucket Name. Specify the S3 bucket name by replacing S3-NAME-GOES-HERE

```
S3_EXE_BUCKET_NAME := S3-NAME-GOES-HERE
```

Remark: The S3 bucket name must be globally unique within AWS. If an error occurs during bucket creation, it may be due to the name being already in use by another user.

- Select the AWS region of the S3 bucket (default is us-west-2):

```
AWS_REGION := us-west-2
```

Modify the
following files



Modify the files and Compile

- Open `/code/common/app.cpp` and set the number of models `N_MODEL` you want to compute (1,200 in our benchmark specification)

```
#define N_MODEL 1200 // total number of models
```

- Open `/code/common/definitions.h` and set the grid size

```
#define NKGRID 100 // grid points on individual capital grid  
#define NKM_GRID 4 // grid points on aggregate capital grid
```

- Open `/code/common/dev options.h` and select the interpolation-range search algorithm:

```
#define _LINEAR_SEARCH 0  
#define _BINARY_SEARCH 0  
#define _CUSTOM_BINARY_SEARCH 1
```

- In the terminal, compile the application for CPU execution (e.g. for benchmark model)

```
make cpu_to_s3 CPU_EXE=1200_100k_4km  
make openmpi_to_s3 OPENMPI_EXE=mpi_1200_100k_4km
```

Create all Binaries for
Sequential Execution

A yellow triangular graphic is located in the bottom right corner of the slide, pointing towards the center.

Create Binaries for Parallel Execution (MPI)

A yellow triangular graphic element is located in the bottom right corner of the slide, pointing towards the center.

Grid Sizes: 100-4, N. Economies: 1200

```
/common/app.cpp      #define N_MODEL 1200 // total number of models

/common/definitions.h #define NKGRID 100 // grid points on individual capital grid
                     #define NKM_GRID 4 // grid points on aggregate capital grid

/common/dev_options.h #define _LINEAR_SEARCH 0
                     #define _BINARY_SEARCH 0
                     #define _CUSTOM_BINARY_SEARCH 1

Terminal launch:      export PATH=$PATH:$HOME/openmpi/bin
                     export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$HOME/openmpi/lib
                     make clean
                     make openmpi_to_s3 OPENMPI_EXE=mpi_1200_100k_4km
```