



FireSim

Running a FireSim Simulation

<https://firesim.com>



[@firesimproject](https://twitter.com/firesimproject)

HPCA Tutorial 2023

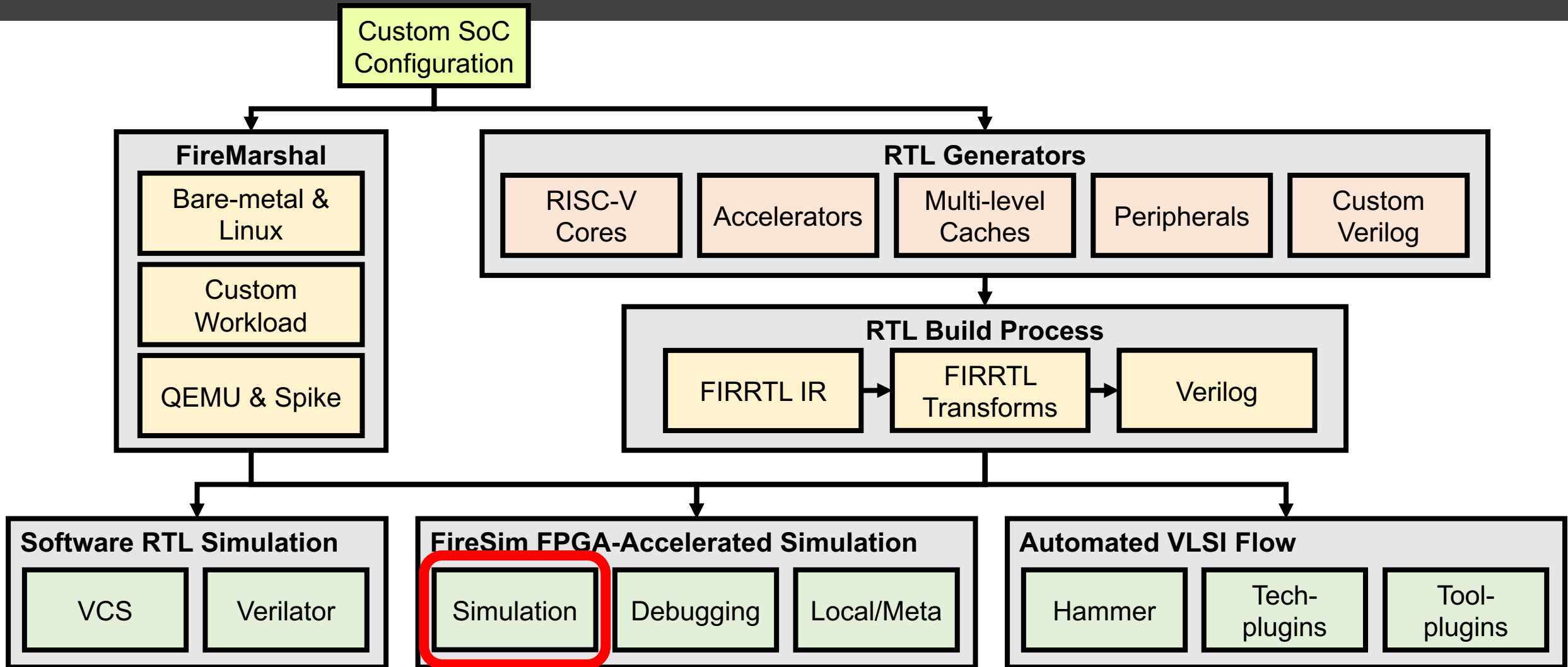
Abraham Gonzalez



Berkeley Architecture Research



Tutorial Roadmap





Agenda

- Configure and launch a simulation runfarm
- Boot Linux interactively on the target hardware
- Deploy new automated workloads



Prerequisites

- Interactive shell commands intended to be run during the tutorial are highlighted in blue blocks (prefixed by “\$”)
- Some simplifying assumptions about the shell environment:
 - The `$FDIR` variable refers to the top directory of FireSim
 - `env.sh` and `sourceme-f1-manager.sh` have been sourced

```
$ cd ~/chipyard-afternoon
$ source ./env.sh

$ cd $FDIR
$ source ./sourceme-f1-manager.sh
```



Prefetching

- We will later be setting up and launching simulations
- To hide setup latency, edit `$FDIR/deploy/config_runtime.yaml` to match the following settings:

```
run_farm:  
  recipe_arg_overrides:  
    run_farm_hosts_to_use:  
      - f1.2xlarge: 1  
  
target_config:  
  topology: no_net_config  
  no_net_num_nodes: 1  
  default_hw_config: firesim_rocket_singlecore_no_nic_l2_lbp
```



Prefetching

- We will later be setting up and launching simulations
- To hide setup latency:
 - Append the following entry to `config_hwdb.yaml`:

Make sure there are no duplicate entries

```
$ cd $FDIR/deploy
$ cat built-hwdb-entries/firesim_rocket_singlecore_no_nic_12_lbp >>
  config_hwdb.yaml
```

- Verify that it follows this format (with a unique AGFI ID):

```
firesim_rocket_singlecore_no_nic_12_lbp:
  agfi: agfi-0e27eb94672e2f5a9
  deploy_triplet_override: null
  custom_runtime_config: null
```

In case `firesim buildbitstream` did not finish in time, a pre-populated entry is provided for you to use



Prefetching

- We will later be setting up and launching simulations
- To hide setup latency, run the following commands:

- First verify that you aren't inside another `tmux` session.
- If so, detach from the existing `tmux` session using `Ctrl+b` then `d`.

```
$ tmux new -s sim  
$ firesim launchrunfarm && firesim infrasetup
```



What did we just do?



Runtime Configuration

What to simulate and what infrastructure is required is controlled by

`$FDIR/deploy/config_runtime.yaml`

- Target-level: Assemble a simulated system from components
 - FPGA images of SoC hardware designs
 - Network topology
 - Workload definition
- Host-level: Specify which EC2 instances to use



config_runtime.yaml

The `run_farm` section specifies the number, type, and other launch parameters of instances to be managed

```
run_farm:
  base_recipe: run-farm-recipes/aws_ec2.yaml
  recipe_arg_overrides:
    run_farm_tag: mainrunfarm
    always_expand_run_farm: true
    launch_instances_timeout_minutes: 60
    run_instance_market: ondemand
    spot_interruption_behavior: terminate
    spot_max_price: ondemand
    default_simulation_dir: /home/centos
```



config_runtime.yaml

The `run_farm` section specifies the number, type, and other launch parameters of instances to be managed

```
run_farm:
  base_recipe: run-farm-recipes/aws_ec2.yaml
  recipe_arg_overrides:
    # ...
  run_farm_hosts_to_use:
    - f1.16xlarge: 0
    - f1.4xlarge: 0
    - f1.2xlarge: 0
    - m4.16xlarge: 0
    - z1d.3xlarge: 0
    - z1d.6xlarge: 0
    - z1d.12xlarge: 0
```



config_runtime.yaml

The `target_config` section specifies the high-level configuration of the system to simulate

```
target_config:
  topology: example_8config
  no_net_num_nodes: 2
  link_latency: 6405
  switching_latency: 10
  net_bandwidth: 200
  profile_interval: -1
  default_hw_config: firesim_rocket_quadcore_nic_l2_1lc4mb_ddr3
  plusarg_passthrough: ""
```

`default_hw_config` references an entry from `config_hwdb.yaml`



config_runtime.yaml

The `workload` section specifies the software to be executed on the simulated nodes

```
workload:  
  workload_name: linux-uniform.json  
  terminate_on_completion: no  
  suffix_tag: null
```

Workload definitions live in `$FDIR/deploy/workloads/*.json`



config_runtime.yaml

Other miscellaneous sections:

- `metasimulation`
- `tracing`: TracerV trace port capture
- `autocounter`: Out-of-band performance counter collection
- `host_debug`: DRAM zeroing, synthesized assertions
- `synthprint`: Synthesized print statements

(These are used in debugging, see the FireSim docs for more info)



Testing the new AGFI

- By now, the `buildbitstream` run that you started at the very beginning of this tutorial should have finished
- Add the hardware entry to `config_hwdb.yaml`:

First remove the old entry, if any

```
$ cd $FDIR/deploy
$ cat built-hwdb-entries/firesim_rocket_singlecore_no_nic_12_lbp >>
  config_hwdb.yaml
```

- Verify that it follows this format (with a unique AGFI ID):

```
firesim_rocket_singlecore_no_nic_12_lbp:
  agfi: agfi-0e27eb94672e2f5a9
  deploy_triplet_override: null
  custom_runtime_config: null
```

In case `firesim buildbitstream` did not finish in time, a pre-populated entry is provided for you to use



Single-Node Simulation

What we modified in `config_runtime.yaml` earlier:

```
run_farm:  
  recipe_arg_overrides:  
    run_farm_hosts_to_use:  
      - f1.2xlarge: 1
```

```
target_config:  
  topology: no_net_config  
  no_net_num_nodes: 1  
  default_hw_config: firesim_rocket_singlecore_no_nic_l2_lbp
```

- Use a smaller f1.2xlarge instance (1 FPGA)
- Simulate one non-networked node without a switch model
- Load the single-core Rocket design without a NIC



Launching Simulation Instances

```
$ firesim launchrunfarm
```

Already running in a tmux session; re-attach with `tmux attach -t sim`

```
FireSim Manager. Docs: https://docs.firesim.com
```

```
Running: launchrunfarm
```

```
Waiting for instance boots: 0 f1.16xlarge
```

```
Waiting for instance boots: 1 f1.2xlarge
```

```
i-0c5c6894d0ac788af booted!
```

```
Waiting for instance boots: 0 f1.4xlarge
```

```
Waiting for instance boots: 0 m4.16xlarge
```

```
Waiting for instance boots: 0 z1d.12xlarge
```

```
Waiting for instance boots: 0 z1d.3xlarge
```

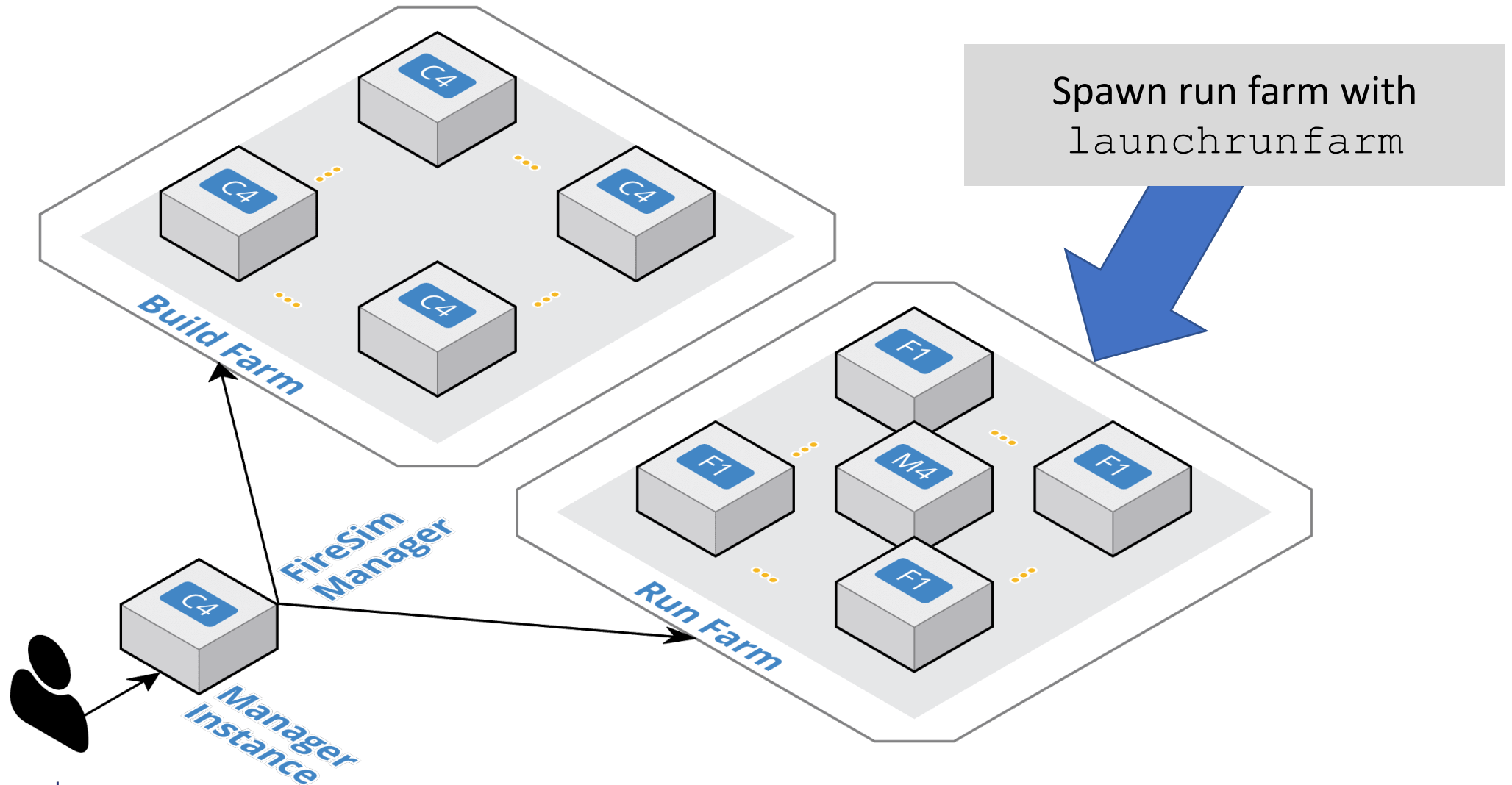
```
Waiting for instance boots: 0 z1d.6xlarge
```

```
The full log of this run is:
```

```
/home/centos/chipyard/sims/firesim/deploy/logs/2022-06-17--23-52-57-launchrunfarm-R50MKTLJ42O36MZZ.log
```



Launching Simulation Instances





Deploying Simulation Infrastructure

```
$ firesim infrasetup
```

Already running!

This deploys various software prerequisites:

- Builds host-side simulation drivers for the specific build triplet
- Builds the switch model executable (if enabled)
- Collects information about simulation instances and transfers files
- Programs the FPGAs with the desired AGFIs



Deploying Simulation Infrastructure

```
$ firesim infrasetup
```

Already running!

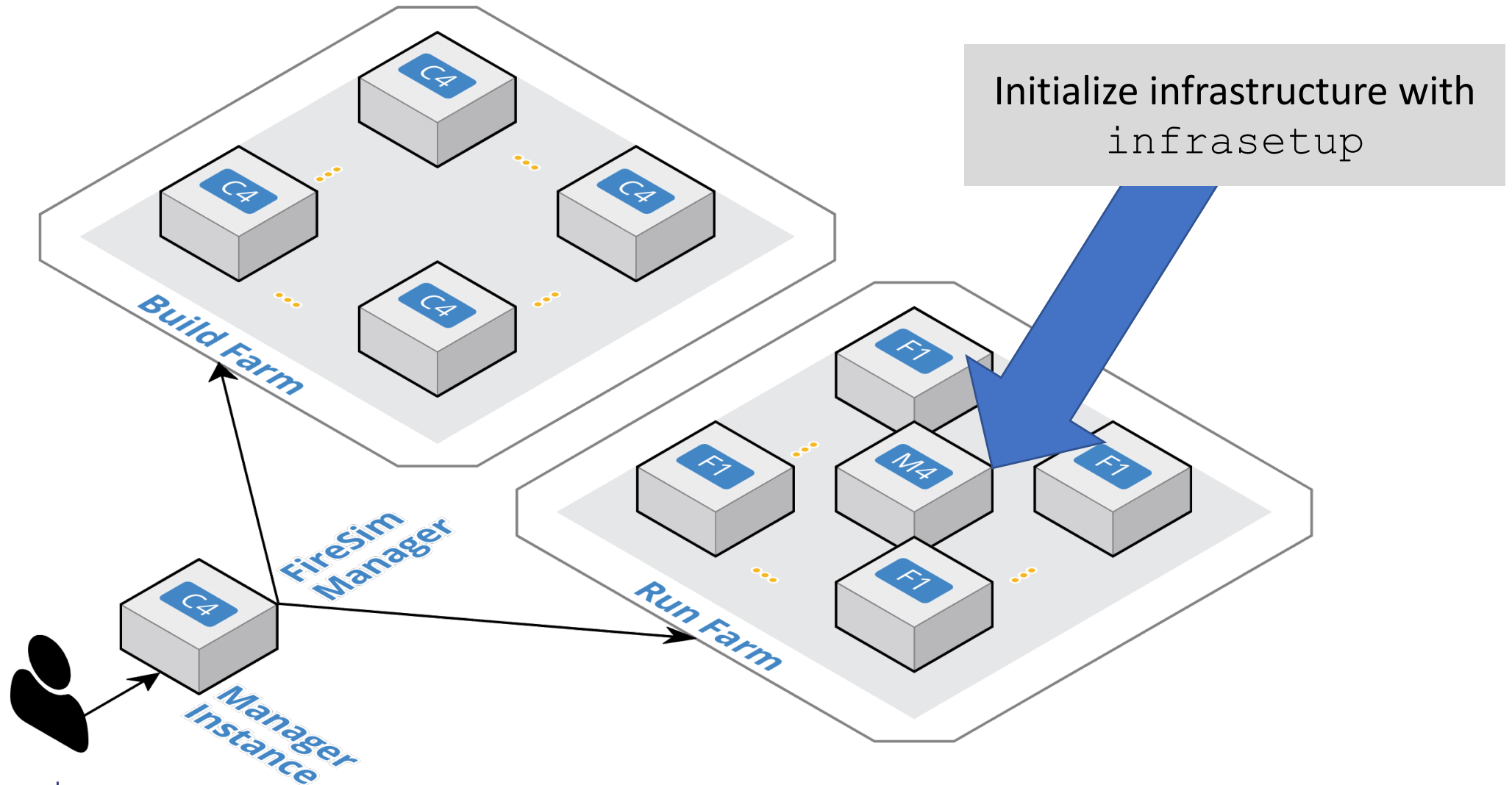
```
FireSim Manager. Docs: https://docs.firesim.com
Running: infrasetup

Building FPGA software driver for FireSim-WithDefaultFireSimBridges_WithFireSimHighPerfConfigTweaks_chipyard.RocketConfig-F90MHz_BaseF1Config
[192.168.3.52] Executing task 'instance_liveness'
[192.168.3.52] Checking if host instance is up...
[192.168.3.52] Executing task 'infrasetup_node_wrapper'
[192.168.3.52] Copying FPGA simulation infrastructure for slot: 0.
[192.168.3.52] Installing AWS FPGA SDK on remote nodes. Upstream hash: 1.12.0-72-gfed0aa6
[192.168.3.52] Unloading XRT-related Kernel Modules.
[192.168.3.52] Copying AWS FPGA XDMA driver to remote node.
[192.168.3.52] Unloading XDMA Driver Kernel Module.
[192.168.3.52] Loading XDMA Driver Kernel Module.
[192.168.3.52] Setting up remote node for qcow2 disk images.
[192.168.3.52] Loading NBD Kernel Module.
[192.168.3.52] Unloading NBD Kernel Module.
[192.168.3.52] Disconnecting all NBDs.
[192.168.3.52] Clearing FPGA Slot 0.
[192.168.3.52] Checking for Cleared FPGA Slot 0.
[192.168.3.52] Flashing FPGA Slot: 0 with agfi: agfi-0e27eb94672e2f5a9.
[192.168.3.52] Checking for Flashed FPGA Slot: 0 with agfi: agfi-0e27eb94672e2f5a9.
[192.168.3.52] Unloading XDMA Driver Kernel Module.
[192.168.3.52] Loading XDMA Driver Kernel Module.
[192.168.3.52] Starting Vivado hw_server.
[192.168.3.52] Starting Vivado virtual JTAG.
The full log of this run is:
/home/centos/chipyard/sims/firesim/deploy/logs/2022-06-18--00-13-05-infrasetup-SJJBKWPYO20THF4.log
```





Deploying Simulation Infrastructure





Running the Simulation

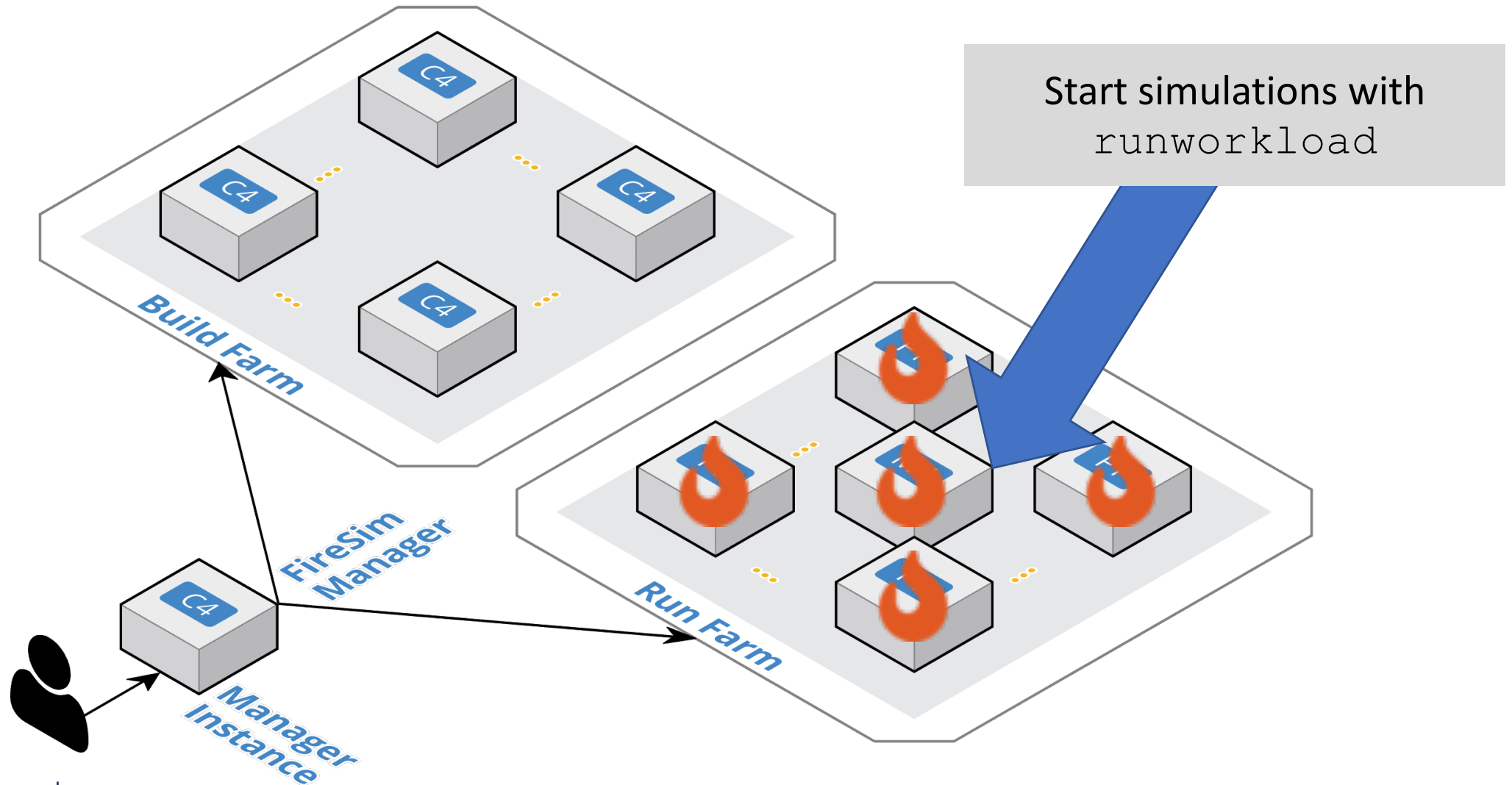
```
$ firesim runworkload
```

```
FireSim Manager. Docs: http://docs.firesim.im  
Running: runworkload
```

```
Creating the directory: /home/centos/chipyard/sims/firesim/deploy/results-  
workload/2022-06-18--00-16-00-linux-uniform/  
[192.168.3.52] Executing task 'instance_liveness'  
[192.168.3.52] Checking if host instance is up...  
[192.168.3.52] Executing task 'boot_switch_wrapper'  
[192.168.3.52] Executing task 'boot_simulation_wrapper'  
[192.168.3.52] Starting FPGA simulation for slot: 0.  
[192.168.3.52] Executing task 'monitor_jobs_wrapper'
```



Running the Simulation





Monitoring the Simulation

You should see a live status report that refreshes periodically:

```
FireSim Simulation Status @ 2022-06-18 00:17:10.188191
-----
This workload's output is located in:
/home/centos/chipyard/sims/firesim/deploy/results-workload/2022-06-18--00-16-00-
linux-uniform/
This run's log is located in:
/home/centos/chipyard/sims/firesim/deploy/logs/2022-06-18--00-16-00-runworkload-
NEZCRUKBA2M44B9M.log
This status will update every 10s.
-----
Instances
-----
Hostname/IP: 192.168.3.52 | Terminated: False
-----
Simulated Switches
-----
Simulated Nodes/Jobs
-----
Hostname/IP: 192.168.3.52 | Job: linux-uniform0 | Sim running: True
-----
Summary
-----
1/1 instances are still running.
1/1 simulations are still running.
-----
```





Interacting with the Simulation

Look for the run instance's IP address in the status:

```
FireSim Simulation Status @ 2022-06-18 00:17:10.188191
-----
This workload's output is located in:
/home/centos/chipyard/sims/firesim/deploy/results-workload/2022-06-18--00-16-00-
linux-uniform/
This run's log is located in:
/home/centos/chipyard/sims/firesim/deploy/logs/2022-06-18--00-16-00-runworkload-
NEZCRUKBA2M44B9M.log
This status will update every 10s.
-----
Instances
-----
Hostname/IP: 192.168.3.52 | Terminated: False
-----
Simulated Switches
-----
Simulated Nodes/Jobs
-----
Hostname/IP: 192.168.3.52 | Job: linux-uniform0 | Sim running: True
-----
Summary
-----
1/1 instances are still running.
1/1 simulations are still running.
-----
```





Interacting with the Simulation

- On the *manager* instance, `ssh` into the run farm instance:

```
$ ssh 192.168.3.52
```

```
┌───────────┐ ┌───────────┐ ┌───────────┐ ┌───────────┐
┌───┐ ┌───┐ ( ┌───┐ ┌───┐ ) ┌───┐ ┌───┐ v ┌───┐ ┌───┐ ┌───┐ ┌───┐
┌───┐ ┌───┐ ┌───┐ ┌───┐ ┌───┐ ┌───┐ ┌───┐ ┌───┐ ┌───┐ ┌───┐ ┌───┐
AMI Version:      1.11.4
Xilinx Version:   2021.1
Readme:           /home/centos/src/README.md
AMI Release Notes: /home/centos/src/RELEASE_NOTES.md
GUI/Cluster setup: https://github.com/aws/aws-fpga/blob/master/developer\_resources
```

- Then attach to the console of the simulated node:

```
$ screen -r fsim0
```



Logging Into the Simulated System

- Once Linux boots, the login prompt should appear over the console
- Log in as `root` with password `firesim` (password does not echo)

```
[    0.085714] EXT4-fs (iceblk): re-mounted. Opts: (null)
Starting syslogd: OK
Starting klogd: OK
Starting mdev... done.
Starting dropbear sshd: OK

Welcome to Buildroot
buildroot login: root
Password:
#
```



Logging Into the Simulated System

- Feel free to experiment with shell commands

```
# uname -a
# cat /proc/cpuinfo
# free -m
# vim
```

- When done, shut down the system

```
# poweroff -f
```

- This will also end the simulation

Finally, exit the `ssh` session with `Ctrl-d` to return to the manager instance



Custom FireSim Workloads

- *Workload*: Series of jobs (software configurations) assigned to run on individual simulations
- Two types of workloads:
 - Uniform**: Homogenous job run by all nodes in a simulated cluster
 - Non-uniform**: Each node is assigned a different job
 - Client/server configurations
 - Benchmark suites (SPEC17)



Workload Definitions

`linux-uniform`: Default workload to boot an interactive buildroot-based GNU/Linux distro on every node

```
{
  "benchmark_name" : "linux-uniform",
  "common_bootbinary" : "br-base-bin",
  "common_rootfs" : "br-base.img",
  "common_outputs" : ["/etc/os-release"],
  "common_simulation_outputs" : ["uartlog", "memory_stats.csv"]
}
```

`$FDIR/deploy/workloads/linux-uniform/br-base{-bin,.img}`
are symlinks to the FireMarshal-generated images



SPEC CPU2017

- 10 jobs – one per benchmark in the SPECrate Integer suite
- Build and install the workloads in [chipyard/software/spec2017](#) using FireMarshal
- Set up `config_runtime.yaml`
 - `f1_2xlarges: 10`
 - `topology: no_net_config`
 - `no_net_num_nodes: 10`
 - `workload name: spec17-intrate.json`
- Select the hardware config to benchmark, then run `firesim launchrunfarm / infrasetup / runworkload`

```
{
  "common_bootbinary" : "bbl-vmlinux",
  "benchmark_name" : "spec17-intrate",
  "deliver_dir" : "spec17-intrate",
  "common_args" : ["--copies 4"],
  "common_files" : ["intrate.sh"],
  "common_outputs" : ["/output"],
  "common_simulation_outputs" : ["uartlog"],
  "workloads" : [
    {
      "name": "500.perlbench_r",
      "files": ["500.perlbench_r"],
      "command": "cd /spec17-intrate && ./intrate.sh 500.perlbench_r",
      "simulation_outputs": [],
      "outputs": []
    },
    {
      "name": "502.gcc_r",
      "files": ["502.gcc_r"],
      "command": "cd /spec17-intrate && ./intrate.sh 502.gcc_r",
      "simulation_outputs": [],
      "outputs": []
    },
    ...
  ]
}
```





All done!

Don't forget to terminate your runfarms (otherwise, we are going to pay for a lot of FPGA time)

```
$ firesim terminatorunfarm
```

Type **yes** at the prompt to confirm