Public

Amazon Elastic Compute Cloud EC2 F1

From: https://aws.amazon.com && https://www.xilinx.com

Arranged by long.a.zhang@tieto.com

Public

小人谋生，君子谋国，大丈夫谋天下。

——鬼谷子

Overview of AWS F1 and SDAccel

About AWS

Over the past ten years, the typical business application architecture has evolved from a desktop-centric installation, then to client/server solutions, and now to loosely coupled web services and service-oriented architectures (**SOA**).

In 2006, Amazon Web Services (**AWS**) began offering **IT infrastructure services** to businesses in the form of **web services** -- now commonly known as **cloud computing**.

One of the key benefits of cloud computing is the opportunity to replace **up-front capital infrastructure expenses** with **low variable costs that scale with your business**.

Global Infrastructure

China

Beijing (2), Ningxia (2)

Benefits of AWS

Low Cost : No Upfront Investment; Low Ongoing Cost,economies of scale. *E.g. : GE Oil & Gas division has started migrating more than half of its core applications to AWS while achieving a 52 percent reduction in its total cost of ownership.*

Agility and Instant Elasticity : AWS provides a massive global cloud infrastructure that allows you to quickly innovate, experiment and iterate. *E.g. : With AWS, developers can deploy hundreds or even thousands of compute nodes in minutes, without having to talk to anyone.*

Open and Flexible : AWS is a language and operating system agnostic platform; Flexible Capacity.

Apps not Ops : Focused on projects that grow the business.

Secure : AWS is a secure, durable technology platform with industry-recognized certifications and audits: *PCI DSS Level 1, ISO 27001, FISMA Moderate, FedRAMP, HIPAA, and SOC 1 (formerly referred to as SAS 70 and/or SSAE 16) and SOC 2 audit reports.*

Public

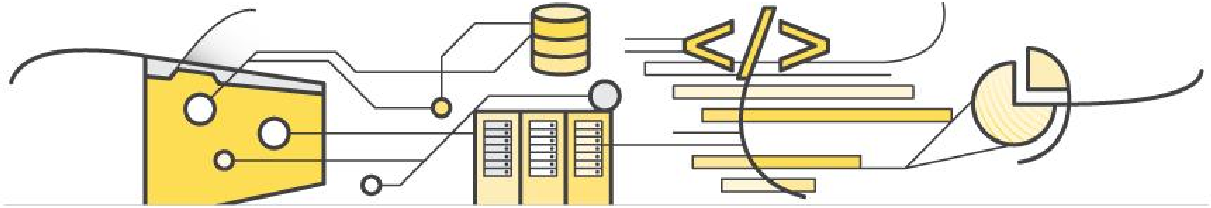
What is “Cloud Computing”?

Public

As defined by Gartner, “Cloud computing is a style of computing where scalable and elastic IT-enabled capabilities are delivered as a service to external customers using Internet technologies.”

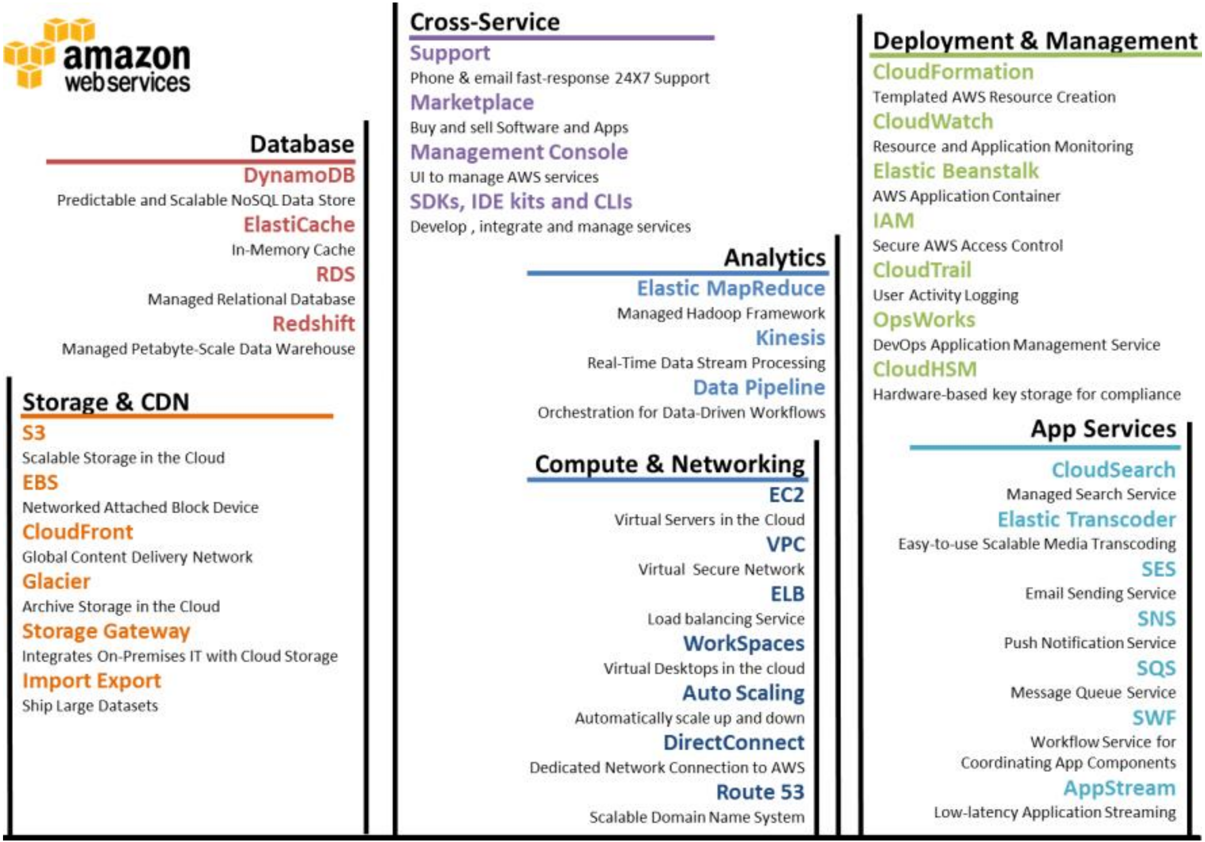
Cloud computing is the on-demand delivery of compute power, database storage, applications, and other IT resources through a cloud services platform via the internet with pay-as-you-go pricing.

Cloud computing has three main types that are commonly referred to as Infrastructure as a Service (**IaaS**), Platform as a Service (**PaaS**), and Software as a Service (**SaaS**).



AWS Cloud Platform

Public



Database

DynamoDB
Predictable and Scalable NoSQL Data Store

ElastiCache
In-Memory Cache

RDS
Managed Relational Database

Redshift
Managed Petabyte-Scale Data Warehouse

Storage & CDN

S3
Scalable Storage in the Cloud

EBS
Networked Attached Block Device

CloudFront
Global Content Delivery Network

Glacier
Archive Storage in the Cloud

Storage Gateway
Integrates On-Premises IT with Cloud Storage

Import Export
Ship Large Datasets

Cross-Service

Support
Phone & email fast-response 24X7 Support

Marketplace
Buy and sell Software and Apps

Management Console
UI to manage AWS services

SDKs, IDE kits and CLIs
Develop , integrate and manage services

Analytics

Elastic MapReduce
Managed Hadoop Framework

Kinesis
Real-Time Data Stream Processing

Data Pipeline
Orchestration for Data-Driven Workflows

Compute & Networking

EC2
Virtual Servers in the Cloud

VPC
Virtual Secure Network

ELB
Load balancing Service

WorkSpaces
Virtual Desktops in the cloud

Auto Scaling
Automatically scale up and down

DirectConnect
Dedicated Network Connection to AWS

Route 53
Scalable Domain Name System

Deployment & Management

CloudFormation
Templated AWS Resource Creation

CloudWatch
Resource and Application Monitoring

Elastic Beanstalk
AWS Application Container

IAM
Secure AWS Access Control

CloudTrail
User Activity Logging

OpsWorks
DevOps Application Management Service

CloudHSM
Hardware-based key storage for compliance

App Services

CloudSearch
Managed Search Service

Elastic Transcoder
Easy-to-use Scalable Media Transcoding

SES
Email Sending Service

SNS
Push Notification Service

SQS
Message Queue Service

SWF
Workflow Service for Coordinating App Components

AppStream
Low-latency Application Streaming

Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers.

Elastic Web-Scale Computing

Completely Controlled

Flexible Cloud Hosting Services: *You have the choice of multiple instance types, operating systems, and software packages.*

Designed for use with other Amazon Web Services

Reliable

Secure: VPC(Virtual Private Cloud), ACL(*Access Control List*)

Inexpensive

Easy to Start

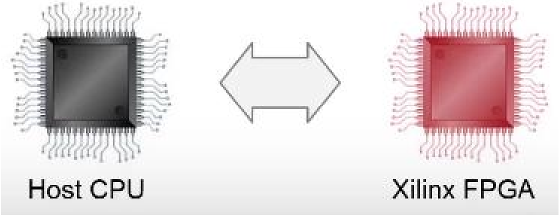
Features of Amazon EC2

- Virtual computing environments, known as ***instances***
- Preconfigured templates for your instances, known as Amazon Machine Images (***AMIs***), that package the bits you need for your server (including the operating system and additional software)
- Various configurations of CPU, memory, storage, and networking capacity for your instances, known as ***instance types***
- Secure login information for your instances using ***key pairs*** (AWS stores the public key, and you store the private key in a secure place)
- Storage volumes for temporary data that's deleted when you stop or terminate your instance, known as *instance store volumes*

- Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as ***Amazon EBS volumes***
- Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as ***regions*** and ***Availability Zones***
- A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using ***security groups***
- Static IPv4 addresses for dynamic cloud computing, known as ***Elastic IP addresses***
- Metadata, known as ***tags***, that you can create and assign to your Amazon EC2 resources
- Virtual networks you can create that are logically isolated from the rest of the AWS cloud, and that you can optionally connect to your own network, known as ***virtual private clouds*** **(VPCs)**

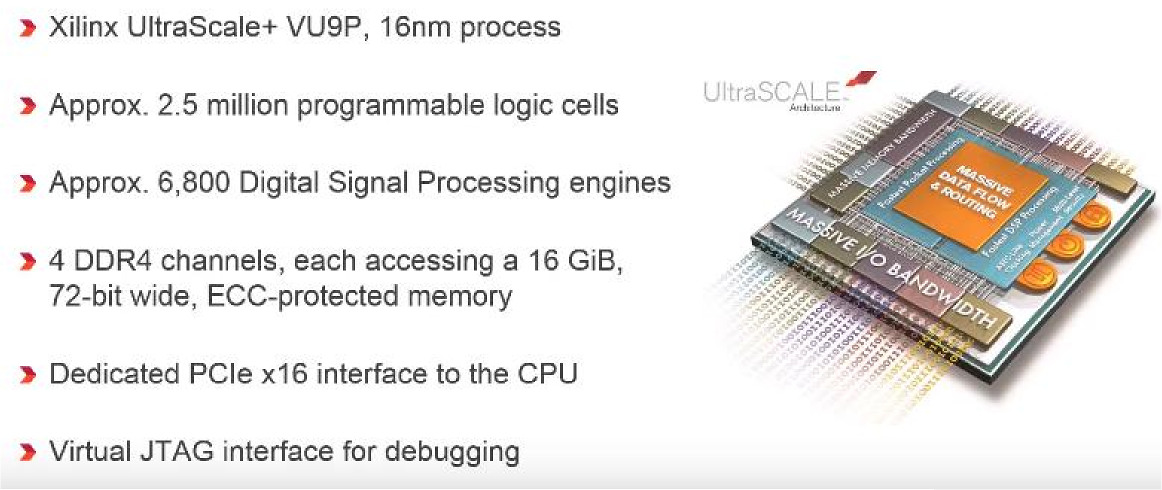
Amazon EC2 F1

- AWS F1 is an elastic cloud compute instance combining x86 CPUs and Xilinx FPGA is to create and run accelerated applications. The FPGA serves as a high-performance acceleration compute resource to the x86 CPU.
- F1 instances come pre-loaded with all the necessary development, simulation and debug tools to create optimized and programmed FPGAs.
- Once a FPGA design is complete, it can be registered as an Amazon FPGA Image(AFI), and it can be deployed in a few clicks on any number of FPGA instances.



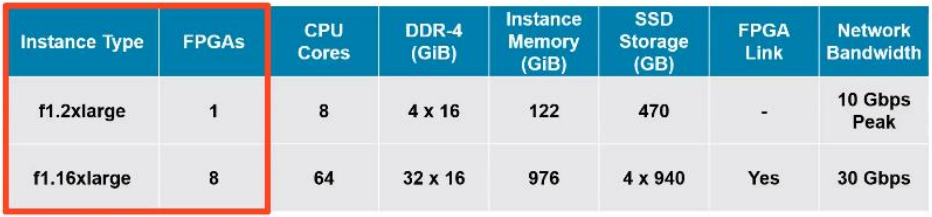
Genomics
Big Data Analytics
Financial Analytics
Security
Image and Video Processing
Machine Learning

The AWS-VU9P-F1 Hardware Platform



- Xilinx UltraScale+ VU9P, 16nm process
- Approx. 2.5 million programmable logic cells
- Approx. 6,800 Digital Signal Processing engines
- 4 DDR4 channels, each accessing a 16 GiB, 72-bit wide, ECC-protected memory
- Dedicated PCIe x16 interface to the CPU
- Virtual JTAG interface for debugging

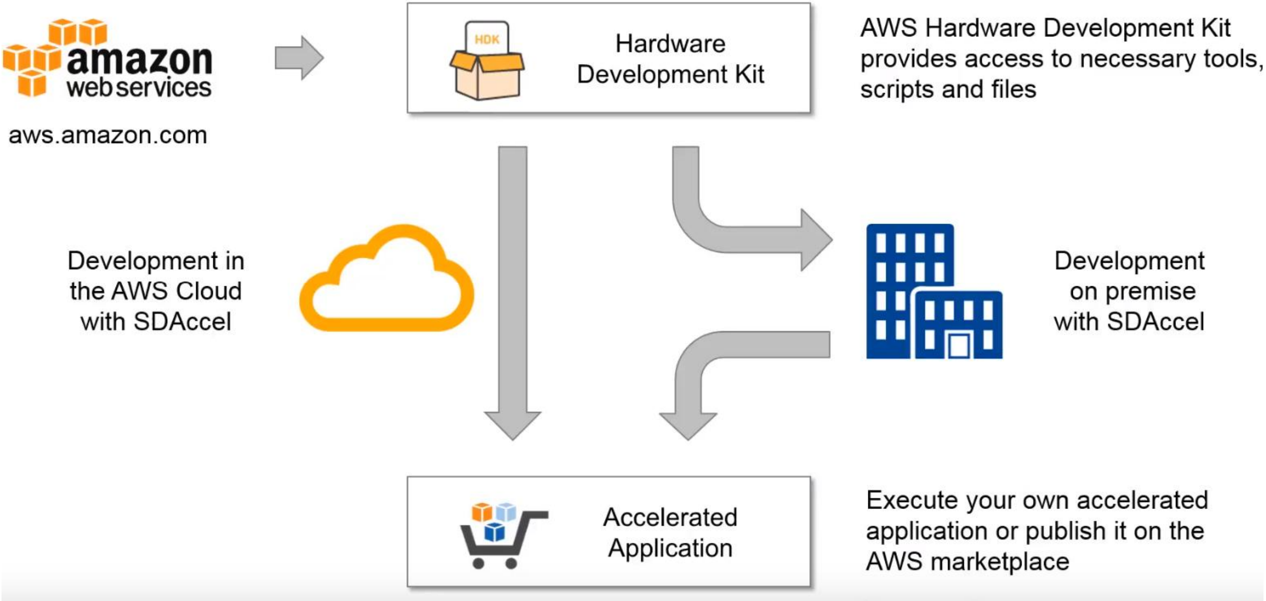
UltraSCALE
Architecture
MASSIVE DATA FLOW & ROUTING
MASSIVE I/O BANDWIDTH
MASSIVE MEMORY BANDWIDTH



Instance Type	FPGAs	CPU Cores	DDR-4 (GiB)	Instance Memory (GiB)	SSD Storage (GB)	FPGA Link	Network Bandwidth
f1.2xlarge	1	8	4 x 16	122	470	-	10 Gbps Peak
f1.16xlarge	8	64	32 x 16	976	4 x 940	Yes	30 Gbps

- Up to 8 high-density Xilinx UltraScale+ 16nm VU9P FPGAs

Amazon F1 Development Flow



aws.amazon.com

Hardware Development Kit

AWS Hardware Development Kit provides access to necessary tools, scripts and files

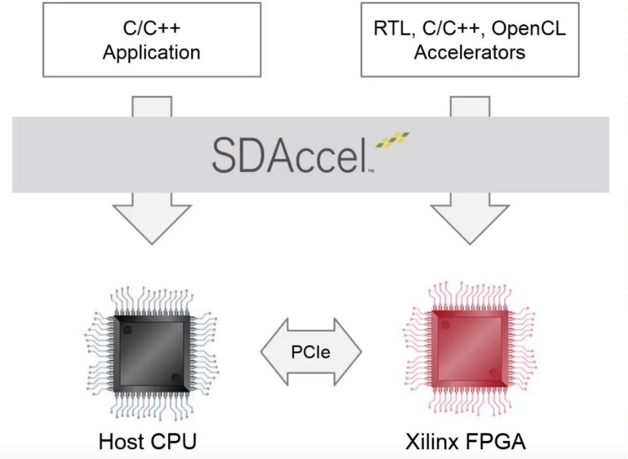
Development in the AWS Cloud with SDAccel

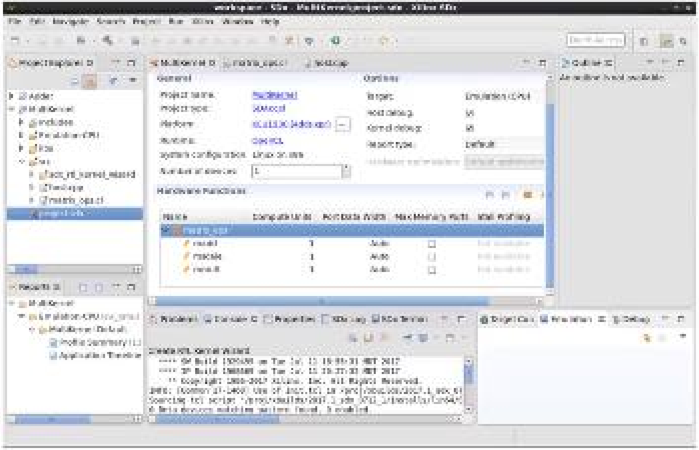
Development on premise with SDAccel

Accelerated Application

Execute your own accelerated application or publish it on the AWS marketplace

The SDAccel Development Environment

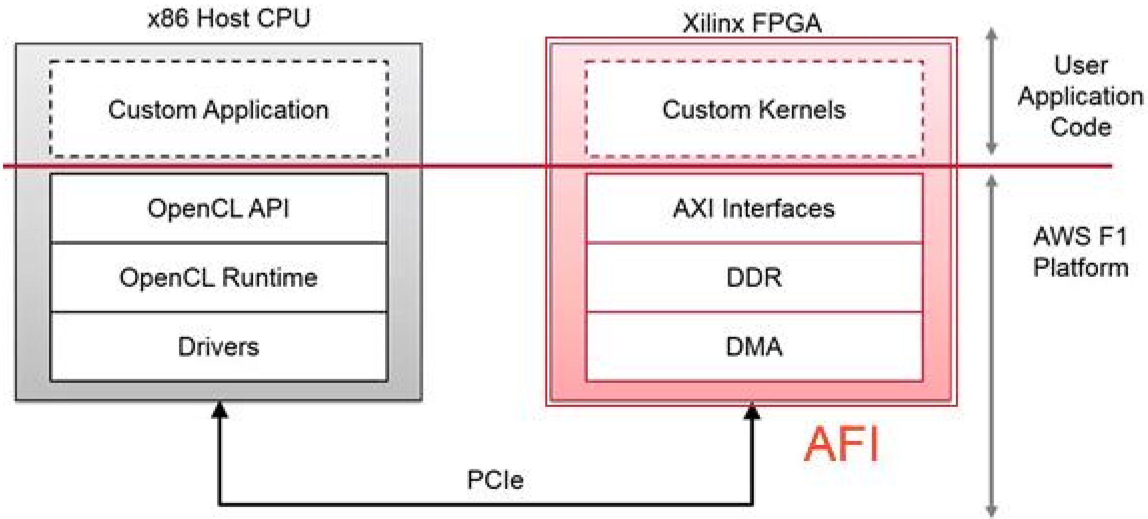




- Fully integrated Eclipse-based environment
- Develop, profile and deploy applications accelerated with Xilinx FPGAs
- Concurrent programming of the host application and FPGA kernels
- Automatic hardware execution flows
- Build-in debug, profiling and performance analysis tools

AWS F1 HW and SW Stacks

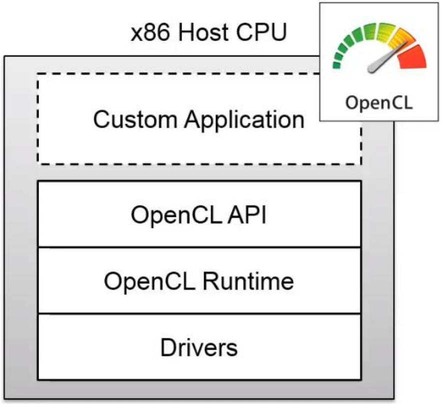
AWS F1 Platform Model



- Amazon FPGA Image(AFI) is the compiled registered design, securely stored
- AWS provide user APIs to create and manage AFIs
- Secured, encrypted and dynamically loaded in the FPGA – can't be copied or download, can be offered on AWS Marketplace associated with an AMI

The Host-Application Execution Stack

- Based on OpenCL - An open industry standard for parallel computing
  - – Standard maintained by Khronos Group (khronos.org)
- Master-slave model cleanly separates application code from kernel logic
- Host application submits work to FPGA kernels using standard OpenCL API
- OpenCL runtime and AWS drivers manage the communication with the FPGA hardware



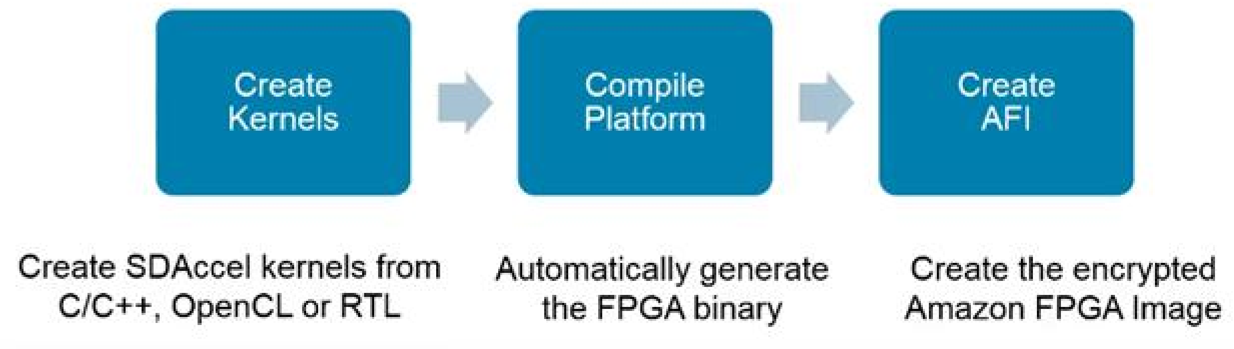
Benefits of OpenCL

- Platform independent programming model designed for heterogeneous computing
- Code portable across CPUs, GPUs, FPGAs etc.
- Easy to learn – many resources available online
- Faster results – vendor provided OpenCL Runtime manages and optimizes kernel communications
- Can swap and load different kernels dynamically
- Portable, open, royalty-free standard

*A **Programming Model** refers to the style of programming where execution is invoked by making what appear to be* library *calls.*

Creating Kernels and Compiling the Amazon FPGA Image(AFI)

AFI Creation Flow Overview


Create Kernels
Compile Platform
Create AFI
Create SDAccel kernels from C/C++, OpenCL or RTL
Automatically generate the FPGA binary
Create the encrypted Amazon FPGA Image

With F1, each FPGA is divided into two partitions:

- Shell (***SH***) – AWS platform logic responsible for taking care of the FPGA external peripherals, PCIe, DRAM, and Interrupts.
- Custom Logic (***CL***) – Custom acceleration logic created by an FPGA Developer.

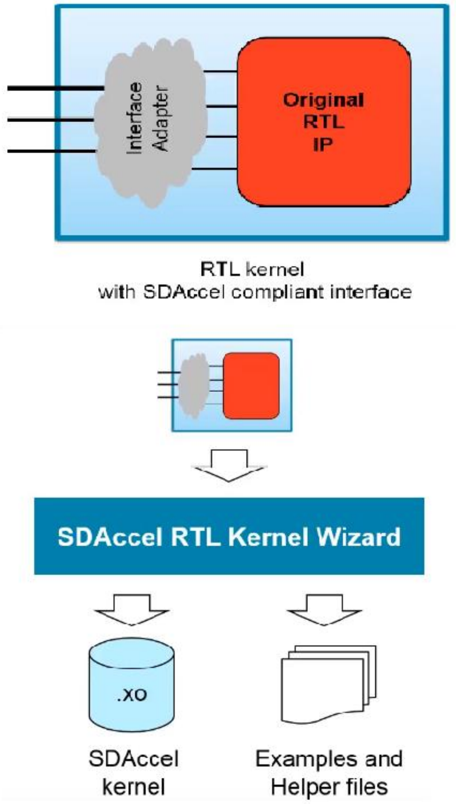
At the end of the development process, combining the Shell and CL creates an Amazon FPGA Image (AFI) that can be loaded onto EC2 F1 Instances.

Creating Kernels from RTL IP

- Custom RTL IP must be packaged as SDAccel "Kernels"
- Kernels must comply with SDAccel interface requirements
- Kernels should be designed with performance goals in mind
  - Interface bandwidth
  - Memory accesses
  - Physical design and timing closure
- SDAccel RTL Kernel Wizard assists in packaging existing RTL IP as Kernels
- Creates Kernel container file (XO file)
  - Kernel XML meta-data
  - RTL files
  - Vivado IP project

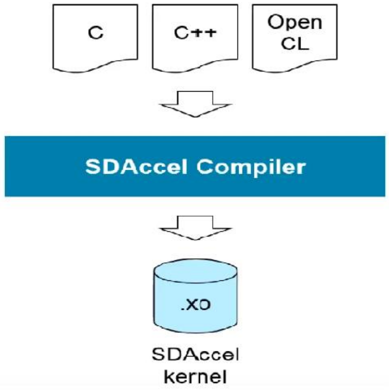
like C be compiled in .obj file and then link to .exe file

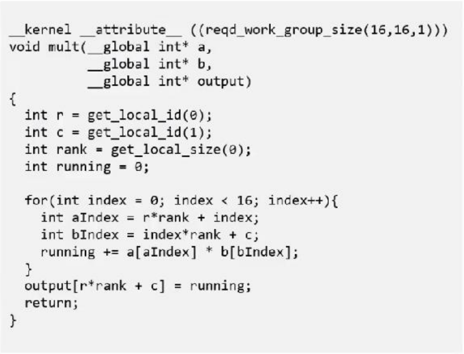
- XO files are the key 'building blocks' used by SDAccel to assemble the final FPGA design


Interface Adapter
Original RTL IP
RTL kernel
with SDAccel compliant interface
SDAccel RTL Kernel Wizard
.XO
SDAccel kernel
Examples and Helper files

Creating Kernels from C/C++, OpenCL

- Parallelizing compiler generates high-performance HW kernels from OpenCL, C, and C++
- Advanced optimizations tuned for Xilinx FPGA devices
  - Memory partitioning
  - DSP block inferencing
  - Loop unrolling, loop pipelining
- Creates HW kernel with necessary AXI interfaces
- Automatically generates SDAccel .xo file
- Comprehensive language support
  - OpenCL 1.0 embedded profile
  - OpenCL 2.0 Pipes
  - OpenCL 2.0 Image Objects
- N-dimensional kernel ranges
- SIMD with vector types
- Math library functions
- Rich set of examples on Github





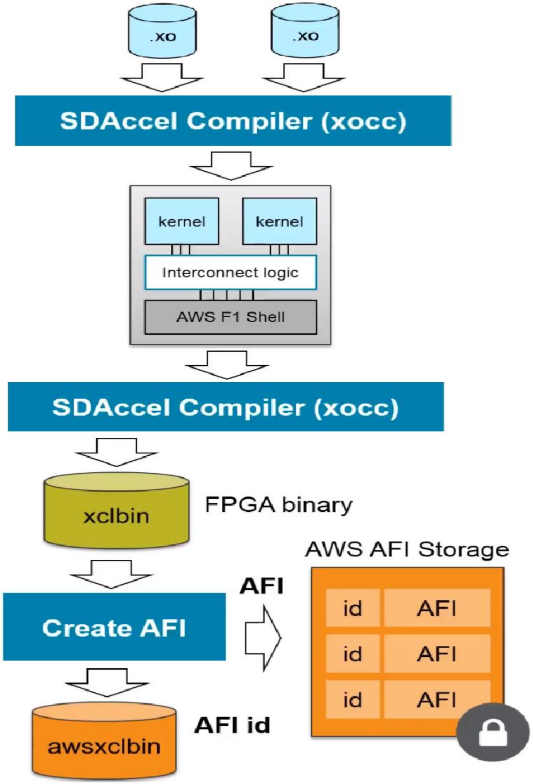
```
__kernel __attribute__ ((reqd_work_group_size(16,16,1)))
void mult(__global int* a,
          __global int* b,
          __global int* output)
{
  int r = get_local_id(0);
  int c = get_local_id(1);
  int rank = get_local_size(0);
  int running = 0;

  for(int index = 0; index < 16; index++){
    int aIndex = r*rank + index;
    int bIndex = index*rank + c;
    running += a[aIndex] * b[bIndex];
  }
  output[r*rank + c] = running;
  return;
}
```

OpenCL matrix multiplication example

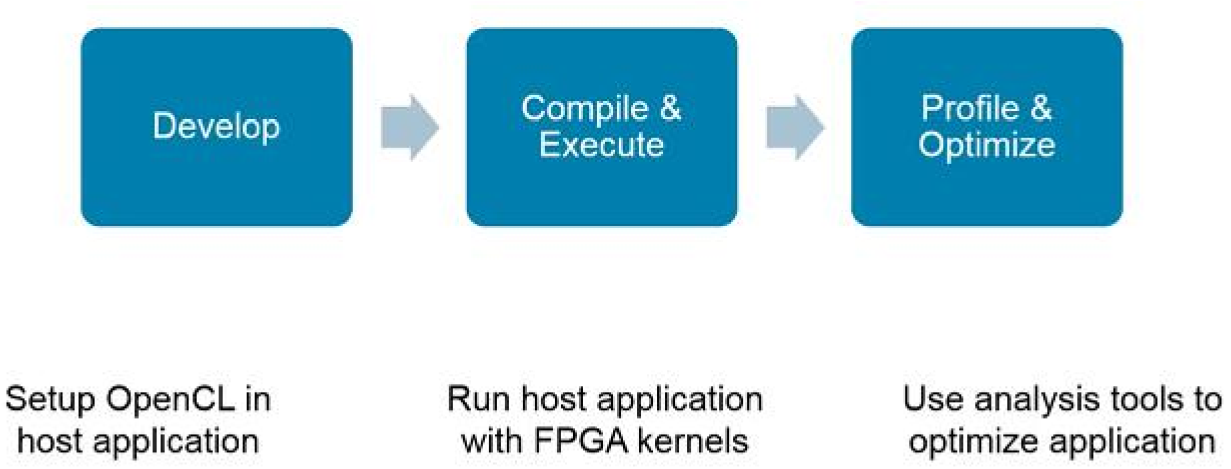
Creating an Amazon FPGA Image

- The SDAccel compiler assembles the FPGA design
- Automatically instantiates the kernels and F1 shell
- Automatically generates DDR interfaces and interconnect logic
- Makes all the necessary connections
- SDAccel runs synthesis and place&route on assembled FPGA design
- Generates FPGA binary (.xclbin)
- Multiple iterations might be required to meet timing goals
- For best results, Kernels should be designed with recommendations from the *UltraFast Design Methodology Guide for the Vivado Design Suite*
- AFIs are created and securely stored by an AWS backend service
- Distributable awsxclbin only contains the AFI id
- AFI id is used at runtime to download the AFI from the Vault into the FPGA
- Application developers have no access to acceleration RTL IP



Developing and Executing a Host Application on F1

Host Application Development Flow Overview



Develop → Compile & Execute → Profile & Optimize

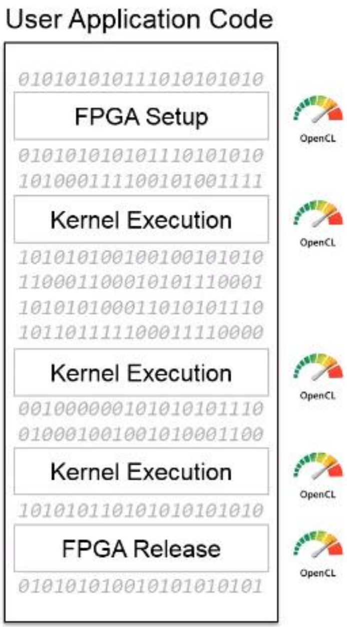
Setup OpenCL in host application

Run host application with FPGA kernels

Use analysis tools to optimize application

Host Application Development Flow Overview

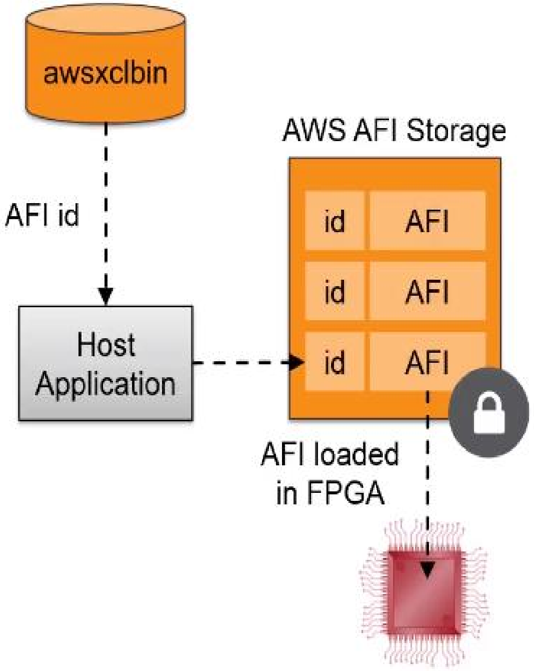
- Application written in C/C++, compiled with GCC
- OpenCL API used to communicate with FPGA
- OpenCL runtime and AWS drivers manage the communication with the FPGA hardware
- Host Application can take many forms
  - Standalone executable
  - Plugin, shared lib, etc…
  - Server for client-server system



User Application Code

Executing with the AFI

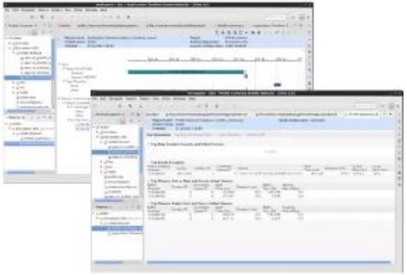
- Host application loads the AFI-id from the awsxclbin metadata
- Host application contacts the AWS storage with the AFI-id
- Backend service downloads the AFI into the FPGA
- Host application can dynamically swap and replace AFIs during runtime

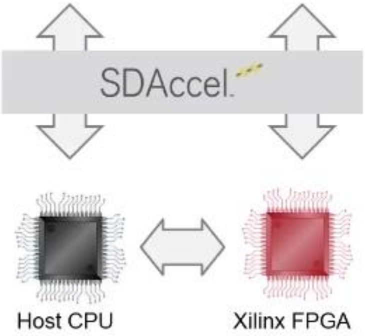


Public

SDAccel Testing, Profiling and Optimization

- SDAccel provides three different excution modes serving different testing needs, CPU Emulation, HW Emulation and HW Execution
- SDAccel provides comprehensive debug and analysis tools to assess health and performance of the system
- Profile Rule Checks highlight performance issues and provide improvement recommendations
- Visualize kernel execution and data transfer efficiency
  - Host and device events displayed on a common timeline
  - OpenCL API call sequence
  - Kernel execution sequence
  - FPGA trace data including AXI transactions, kernel start/stop, etc.

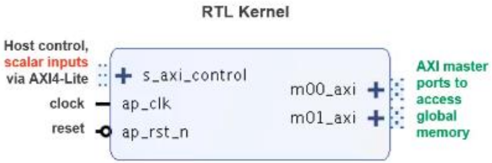




Packaging and Integration of RTL IP for AWS F1

RTL Kernel: Programming Paradigm

```
    void func( int length,
               int *a,
               int *b,
               int *output ) { … }
```



- SDAccel associates specific C function argument types (host-code) with specific HW ports types (RTL kernel)
- RTL kernel needs a AXI-Lite Slave port for scalars arguments
- RTL kernel needs a AXI MM Master port for pointer arguments
- Scalar arguments:
  - Inputs only
  - Written to the kernel via AXI4-lite interface
- Pointer arguments:
  - Inputs or outputs
  - Data resides in the global memory
  - Kernel is responsible for accessing the data through the AXI4 master interface
  - The base address of the memory is passed via the AXI4-lite interface
- The kernel is started and polled for completion status via AXI4-Lite

Public

大知闲闲，小知间间；大言炎炎，小言詹詹。

——庄子 • 齐物论

tieto

Long Zhang

Hardware Engineer
Tieto Oyj, ZSR Product Development Services /
long.a.zhang@tieto.com