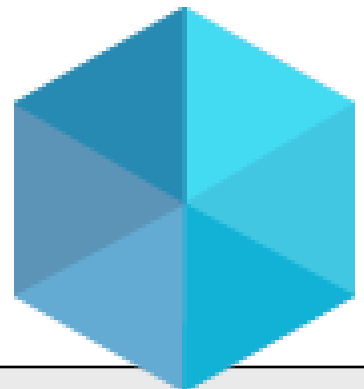




The CAPI SNAP Framework for Programmers

Bruce Wile (bwile@us.ibm.com)

Frank Haverkamp (haverkam@de.ibm.com)



CAPI SNAP Framework Motivations

SNAP = Storage, Networking, and Analytics Programming

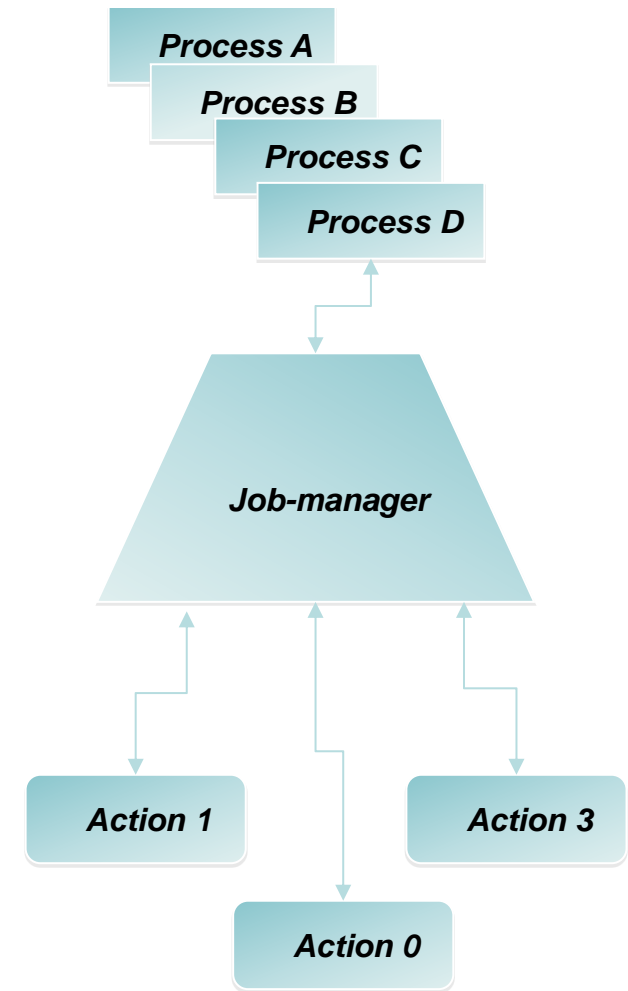
- Target Application Developers with two directives:
 1. The framework must make it easy for programmers to call accelerators and write their own acceleration IP.
 2. The framework must be open source to enable continued enhancements and cross company collaboration.
- Leverage the high IO bandwidth of FPGA cards to place compute closer to the data.
- Leverage POWER8's CAPI interface for its thread programmability model and high performance exchange between host and FPGA.
 - Architect for future generations (CAPI 2.0, OpenCAPI)

Content

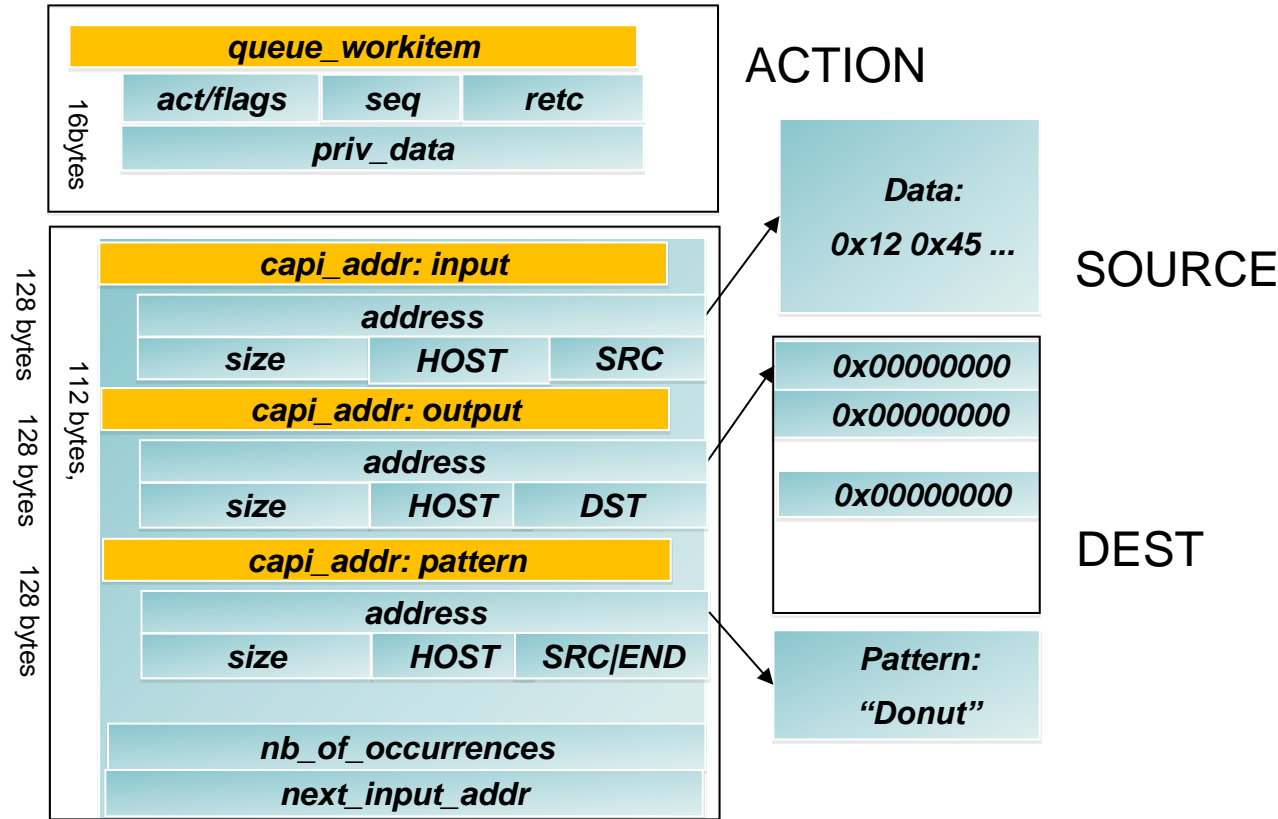
- Donut Job-Queue Mode
- Donut Fixed-Action Mode
- Donut Direct-Access Mode

Donut Job-Queue Mode

- When to use this mode?
 - FPGA-action executes a job and returns after completion
- What do you get?
 - Support for multiple processes N using multiple FPGA-actions M virtually in parallel controlled by built in job-manager
 - Generic job-execution model with request and completion queue per AFU context
 - Prefetching of memory areas, if possible
- Possible use-cases
 - FPGA acceleration within a Cloud, e.g. using Docker virtualization



#actions depend on size of logic and FPGA



```
dnut_prepare_search(&cjob, &sjob_in, &sjob_out, dbuff, dsize,
offs, items, pbuff, psize);
```

```
snprintf(device, sizeof(device)-1, "/dev/cxl/afu%d.0m", card_no);
kernel = dnut_kernel_attach_dev(device, DNUT_VENDOR_ID_IBM,
DNUT_DEVICE_ID_SEARCH, SEARCH_ACTION_TYPE);
```

```
run = 0;
gettimeofday(&stime, NULL);
do {
    rc = dnut_kernel_sync_execute_job(kernel, &cjob, timeout);
    if (cjob.retc != DNUT_RETC_SUCCESS)
        goto out_error2;

    dnut_print_search_results(&cjob, run);

    /* trigger repeat if search was not complete */
    if (sjob_out.next_input_addr != 0x0) {
        input_size -= (sjob_out.next_input_addr - (unsigned long)input_addr);
        input_addr = (uint8_t*)(unsigned long)sjob_out.next_input_addr;

        /* Fixup input address and size for next search */
        sjob_in.input.addr = (unsigned long)input_addr;
        sjob_in.input.size = input_size;
    }
    total_found += sjob_out.nb_of_occurrences;
    run++;
} while (sjob_out.next_input_addr != 0x0);
gettimeofday(&etime, NULL);
```

```
fprintf(stdout, "RETC=%x\n", cjob.retc);
fprintf(stdout, "%d patterns found.\n", total_found);
fprintf(stdout, "searching took %lld usec\n",
(long long)timediff_usec(&etime, &stime));
```

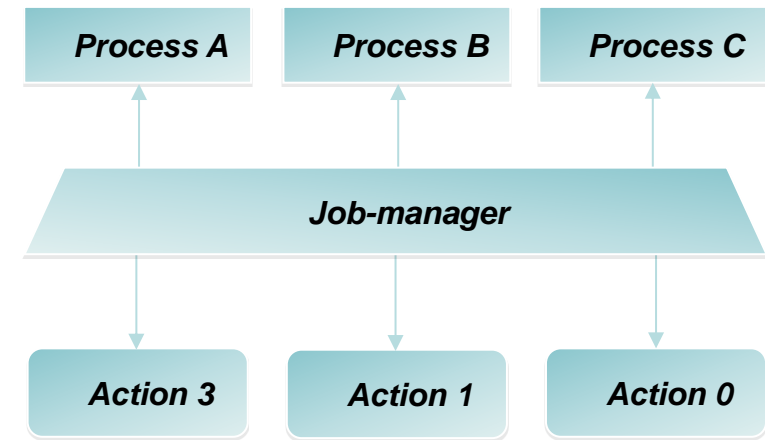
```
dnut_kernel_free(kernel);
```

INPUT

- action = 0x05 e.g. TEXT_SEARCH
- input.addr = <hostmem>, type = HOST, flags = SRC
- output.addr = <hostmem>, type = HOST, flags = DST
- pattern.addr = <hostmem>, type = HOST, flags = SRC|END

Donut Fixed-Action Mode

- When to use this mode?
 - FPGA-action is designed to permanently run
 - FPGA-action can keep state
 - Data-streaming approach with data-in and data-out queue
 - Event driven operation
- What do you get?
 - Support for N processes using N FPGA-actions in parallel
 - FPGA-action attachment to one process exclusively
 - Selected FPGA-action MMIOs are mapped into the process address space
 - Dedicated interrupt(s) per action
- Possible use-cases
 - Use-cases where FPGA-action must permanently run
 - Networking

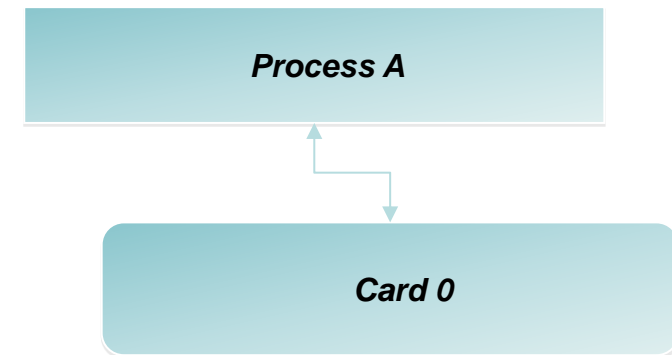


Example

Searching “**man with baguette**” in
n video streams
(n: #input_sources)

Donut Direct-Access Mode

- When to use this mode?
 - Device configuration
 - Statistic gathering
 - Debugging the FPGA
- What do you get?
 - Direct access to FPGA AFU Master or Slave MMIOs



Summary

- 3 Different modes cover variety of usage scenarios
 - Job-Queue Mode: Multiprocessing/threading support with hardware job-scheduling enables very efficient use of hardware e.g. for use in Cloud environments
 - Fixed Action Mode: Dedicated use of the hardware supporting multiple processes
 - Direct-Access Mode: Configuration management, debugging
- Open-sourced hardware and software for SNAP framework
- Question: Are there different scenarios, which we cannot cover with the suggested operation modes?
- If you like to participate, please let us know