

Virtualizing the Tolapai Accelerator

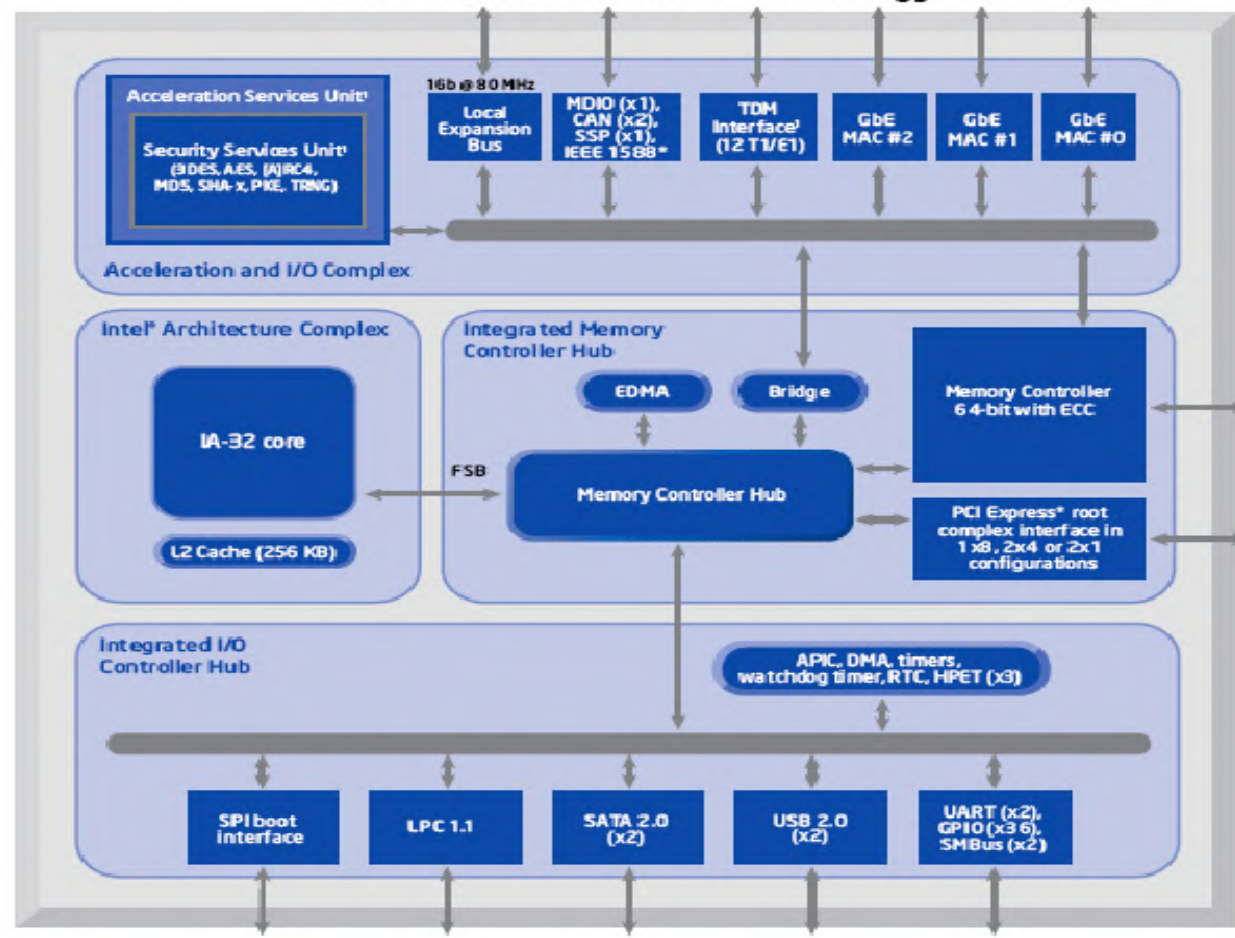
Venkat

Scope

- Tolapai Architecture
- Virtualization Approach
- Implementation Details
- Setup and Execution

Tolapai Architecture

**Block Diagram for the Intel® EP80579 Integrated Processor
with Intel® QuickAssist Technology**



Crypto Accelerator

- Hardware Support for symmetric and asymmetric cryptographic algorithms
- Crypto engine appears as a PCI device to the host CPU
- Separate non-cache coherent memory region for communication between CPU and crypto
- Crypto can also access cache coherent host memory
- Intel driver initializes the crypto and sets up shared memory rings for communication with the host
- Driver provides APIs for executing crypto requests on the device.

Crypto Request – Example

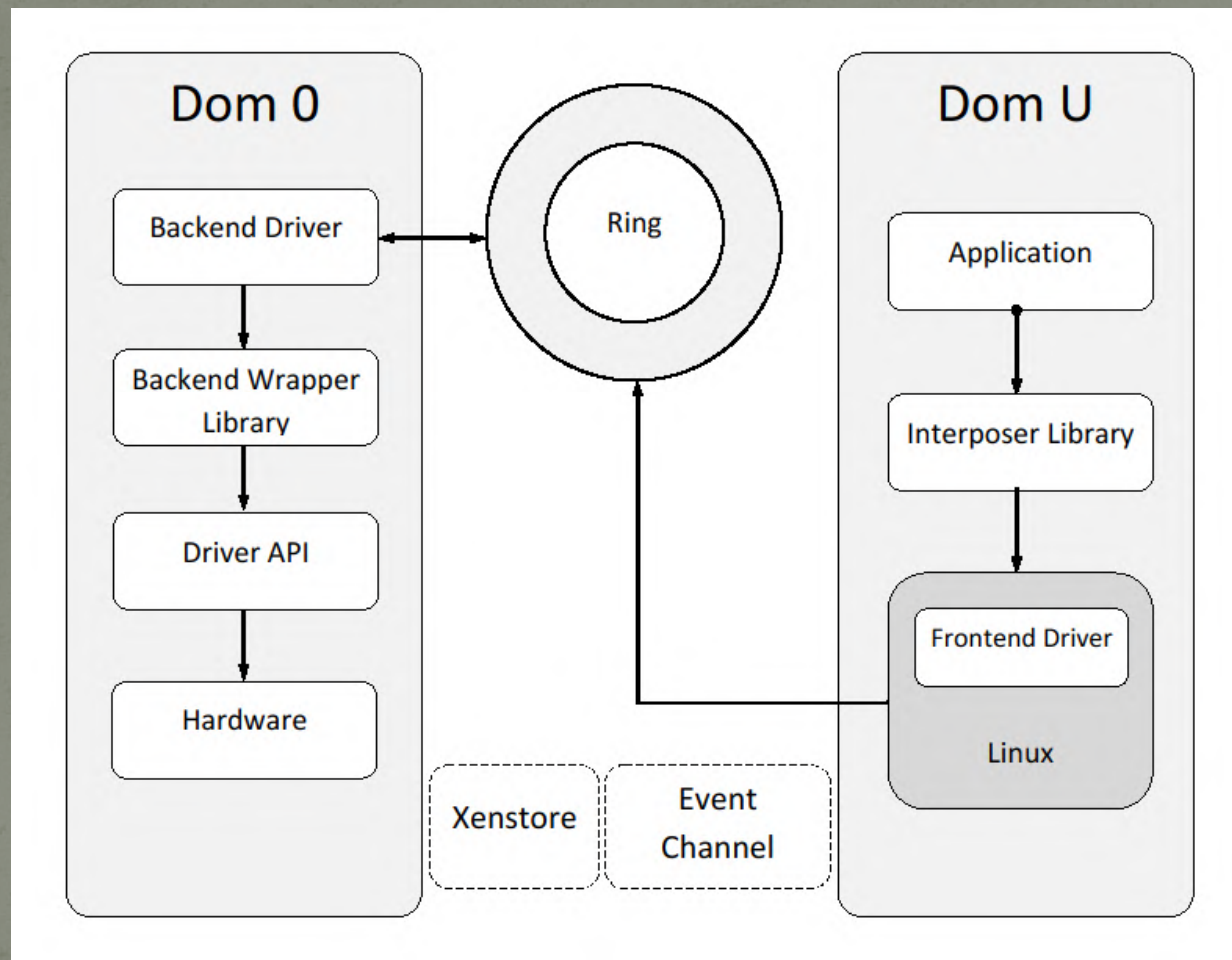
```
pOpData->pSessionCtx = pSessionCtx;  
pOpData->packetType = CPA_CY_SYM_PACKET_TYPE_FULL;  
pOpData->pIv = pIvBuffer;  
pOpData->ivLenInBytes = sizeof(sampleCipherIv);  
pOpData->cryptoStartSrcOffsetInBytes = 0;  
pOpData->messageLenToCipherInBytes = sizeof(sampleCipherSrc);
```

```
status = cpaCySymPerformOp(CPA_INSTANCE_HANDLE_SINGLE,  
    (void *)&complete, /* data sent as is to the callback function */  
    pOpData,           /* operational data struct */  
    pBufferList,       /* source buffer list */  
    pBufferList,       /* same src & dst for an in-place operation */  
    NULL);
```


Virtualization Approach

- Split driver model of virtualization
- A Frontend Wrapper Library in domU provides Intel driver API like interface to the domU applications
- The frontend driver communicates with the backend to satisfy the request
- Backend resides on privileged domain and interacts directly with the Intel driver APIs
- Xenbus state machine and XenStore used to establish connection between dom0 and domU
- Xen shared memory rings used for control/data path

Split Driver Architecture



Implementation Details

- Split Driver based on the xen block device drivers blkfront and blkback
- Backend registers device named 'ixp' with the xenstore
- To issue a request, front end allocates a page to fill the crypto parameters and places a request on the xen ring with the grant reference for the page
- Backend handles the request and places the response back on the same page
- Currently, the driver only supports DES encryption and request size cannot exceed a single page (4KB)

Control Message – Xen Ring

```
struct ixp_request {
    uint8_t    operation;          /* IXIF_OP_??? */
    uint8_t    nr_segments;        /* number of Segments */
    ixp_vdev_t  handle;            /* only for read/write requests */
    uint64_t    id;                /* private guest value, echoed in resp */
    struct ixp_request_segment {
        grant_ref_t gref;          /* reference to I/O buffer frame */
    } seg[IXPIF_MAX_SEGMENTS_PER_REQUEST];
};

struct ixp_response {
    uint64_t    id;                /* copied from request */
    uint8_t     operation;         /* copied from request */
    int16_t     status;            /* IXPIF_RSP_??? */
    uint16_t     resp_size;
};
```

Data Message – DES Request

```
struct des_request {  
    uint16_t key_size;  
    uint16_t iv_size;  
    uint16_t msg_size;  
    char *key;  
    char *iv;  
    char *msg;  
};
```


VM Migration

- Current version of the split driver supports VM migration
- Prior to migration, domU initiates a Xenbus state transition from connected to suspend state
- All pending requests are discarded and request queues are cleaned up
- After migration, domU sets up connection with the new domo
- Current version of the driver not tested with live migration

Setup and Execution

- Linux/Xen versions tested with
 - Ubuntu 9.04
 - 2.6.31.* pvops kernel (jeremy's git)
 - Xen-unstable (3.4.x)
 - Intel driver version 1.0.3-98
- Other configs known to work
 - Redhat/CentOS
 - 2.6.18 kernel-xen
 - Xen 3.4.x
 - Intel driver version 1.0.1-66
- Split driver source can be downloaded from
 - <http://code.google.com/p/tolapai>
- Intel driver source can be obtained from <http://intel.com/soc>

Patching the Intel Driver

- Following patch is required to make the intel crypto driver work with xen domo. It makes the driver machine address aware and converts certain physical address references to machine address references
- Patch works with driver version 1.0.3-98. Steps to apply the patch:

```
tar -xzvf Security.L.1.0.3-98.tar.gz .  
mv Security.L.1.0.3-98 EP805xxx_release  
cd EP805xxx_release  
patch -p1 -i tolapai-1.0.3-98.patch  
export ICP_ROOT=../EP805XXX_release  
export ICP_BUILDSYSTEM_PATH=$ICP_ROOT/build_system  
export ICP_BUILD_OUTPUT=$ICP_ROOT/StagingArea  
export KERNEL_SOURCE_ROOT=/usr/src/<kernel-dir>  
make && make install
```

Patching the Gigabit Ethernet Driver

- The gigabit ethernet driver does not work with the latest 2.6.31.* kernels
- The following patch ports the driver for the latest kernel version:

```
cd EP805xxx_release  
patch -p1 -i tolapai_gbe_1.0.3-98.patch
```


Running the Split Driver

- Extract source to /opt/src/split_driver
- Set environment variables

```
export TOLAPAI_ROOT=/opt/src/split_driver
export TOLAPAI_BUILD_OUTPUT=/opt/src/split_driver/build
export KERNEL_SOURCE_ROOT=/lib/modules/`uname -r`/build
```
- 'make' each of the above mentioned source directories
- Execute backend driver

```
insmod /opt/src/split_driver/build/backend/wrapper.ko
insmod /opt/src/split_driver/build/frontend/ixpback.ko
```

Running the Split Driver (Cont..)

- Execute frontend driver in domu
`insmod/opt/src/split_driver/build/frontend/wrapper.ko`
`insmod /opt/src/split_driver/build/frontend/ixpfront.ko`

- Initialize xenstore in domo

```
sh /opt/src/split_driver/build/backend/xenstore_script.sh ixp <domid of domu>
```

The ixp device should have now moved to connected state.

Testing with Sample Application

- The sample application simply issues the specified number of DES encryption requests from the domU. To execute the sample test,

```
cd /opt/src/split_driver/sample_app  
make  
insmod cpa_cipher_perf.ko
```

- The output can be verified from the dmesg logs

Misc Tricks / Troubleshooting

- Gigabit Ethernet Driver Crash
 - Try changing the `irq_return_t` return type in the `iegbe_intr*` functions to `int` in the file `iegbe_main.c`
 - Try loading the driver without plugging in the Ethernet cord
- Display Driver Problems – No GUI
 - Boot the kernel in single user mode by adding the 'single' parameter at the end of the boot line
 - Modify the driver in `/etc/X11/xorg.conf` to 'vesa'
 - In the root prompt, change to init level 2 by doing 'init 2'
 - Then do 'startx' to start the X-Server

SVN Repository

- All the source code, patches and documentation can be downloaded from

<http://www.code.google.com/p/tolapai>