BBU POOLING FRAMEWORK

NPG, 26th, Oct 2018.

Legal Disclaimer

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: http://www.intel.com/design/literature.htm%20 Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, visit Intel Performance Benchmark Limitations

All products, computer systems, dates and figures specified are preliminary based on current expectations, and are subject to change without notice.

Celeron, Intel, Intel logo, Intel Core, Intel Inside, Intel Inside logo, Intel. Leap ahead., Intel. Leap ahead. logo, Intel NetBurst, Intel SpeedStep, Intel XScale, Itanium, Pentium, Pentium Inside, VTune, Xeon, and Xeon Inside are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Intel® Active Management Technology requires the platform to have an Intel® AMT-enabled chipset, network hardware and software, as well as connection with a power source and a corporate network connection. With regard to notebooks, Intel AMT may not be available or certain capabilities may be limited over a host OS-based VPN or when connecting wirelessly, on battery power, sleeping, hibernating or powered off. For more information, see http://www.intel.com/technology/iamt.

64-bit computing on Intel architecture requires a computer system with a processor, chipset, BIOS, operating system, device drivers and applications enabled for Intel® 64 architecture. Performance will vary depending on your hardware and software configurations. Consult with your system vendor for more information.

No computer system can provide absolute security under all conditions. Intel® Trusted Execution Technology is a security technology under development by Intel and requires for operation a computer system with Intel® Virtualization Technology, an Intel Trusted Execution Technology-enabled processor, chipset, BIOS, Authenticated Code Modules, and an Intel or other compatible measured virtual machine monitor. In addition, Intel Trusted Execution Technology requires the system to contain a TPMv1.2 as defined by the Trusted Computing Group and specific software for some uses. See http://www.intel.com/technology/security/ for more information.

†Hyper-Threading Technology (HT Technology) requires a computer system with an Intel® Pentium® 4 Processor supporting HT Technology and an HT Technology-enabled chipset, BIOS, and operating system. Performance will vary depending on the specific hardware and software you use. See www.intel.com/products/ht/hyperthreading_more.htm for more information including details on which processors support HT Technology.

Intel® Virtualization Technology requires a computer system with an enabled Intel® processor, BIOS, virtual machine monitor (VMM) and, for some uses, certain platform software enabled for it. Functionality, performance or other benefits will vary depending on hardware and software configurations and may require a BIOS update. Software applications may not be compatible with all operating systems. Please check with your application vendor.

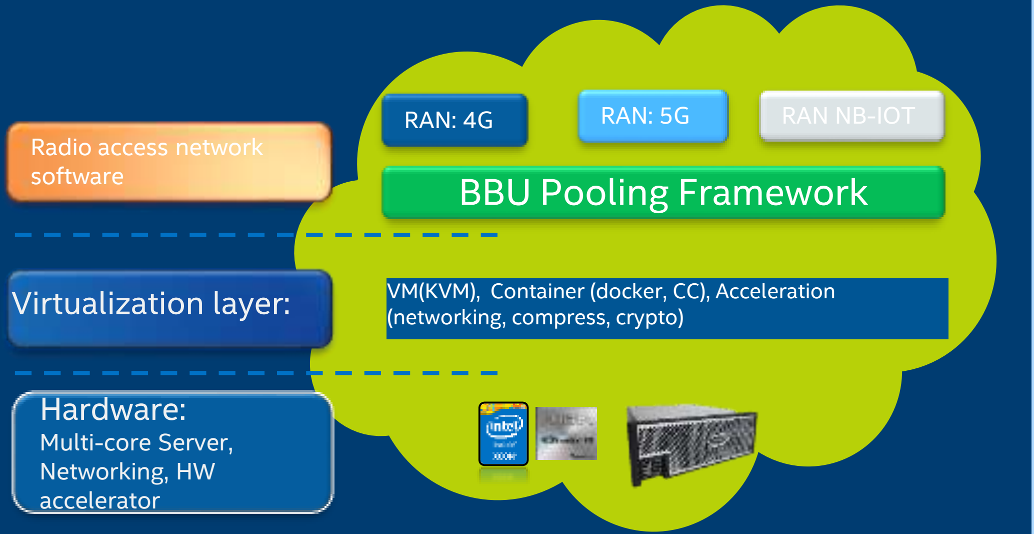
* Other names and brands may be claimed as the property of others.

Other vendors are listed by Intel as a convenience to Intel's general customer base, but Intel does not make any representations or warranties whatsoever regarding quality, reliability, functionality, or compatibility of these devices. This list and/or these devices may be subject to change without notice.

Agenda

- Overview
- Key Functionality
- Working Mode
- An Example

Overview



Radio access network software

RAN: 4G

RAN: 5G

RAN NB-IOT

BBU Pooling Framework

Virtualization layer:

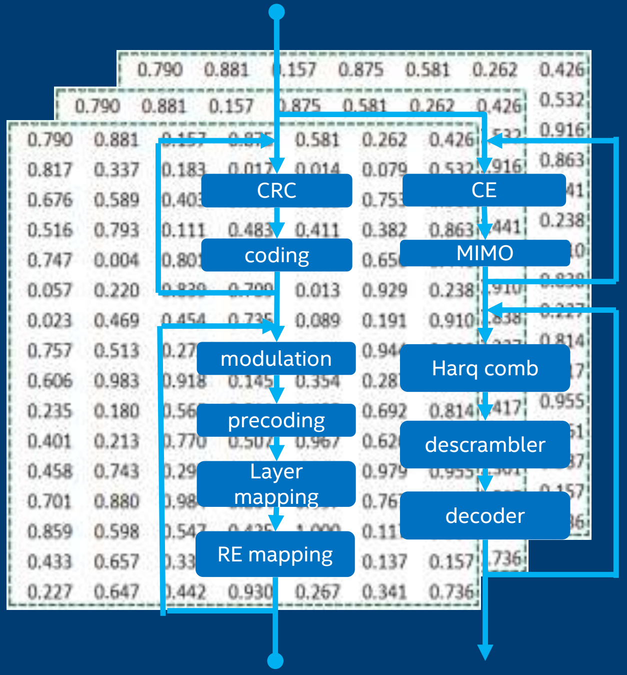
VM(KVM), Container (docker, CC), Acceleration (networking, compress, crypto)

Hardware:
Multi-core Server, Networking, HW accelerator

1. Focus on the **radio access network** (RAN)scenario.
2. Help customer leverage IA Computing resource to design high **efficient** software.
3. Help customer leverage IA general purpose processer character and virtualization technic to design high **flexible** software.

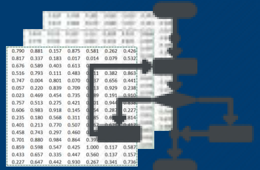
- main functionality is **task dispatching** and **task management** .
- provide core management, help application achieve **intelligent hardware resource management**.
- Pure library, portable and composable.

Tasks



Example of physical layer signal processing, Modules are computing intensive algorithms.

Breaking down the system into reasonable size of tasks that can be executed parallel.

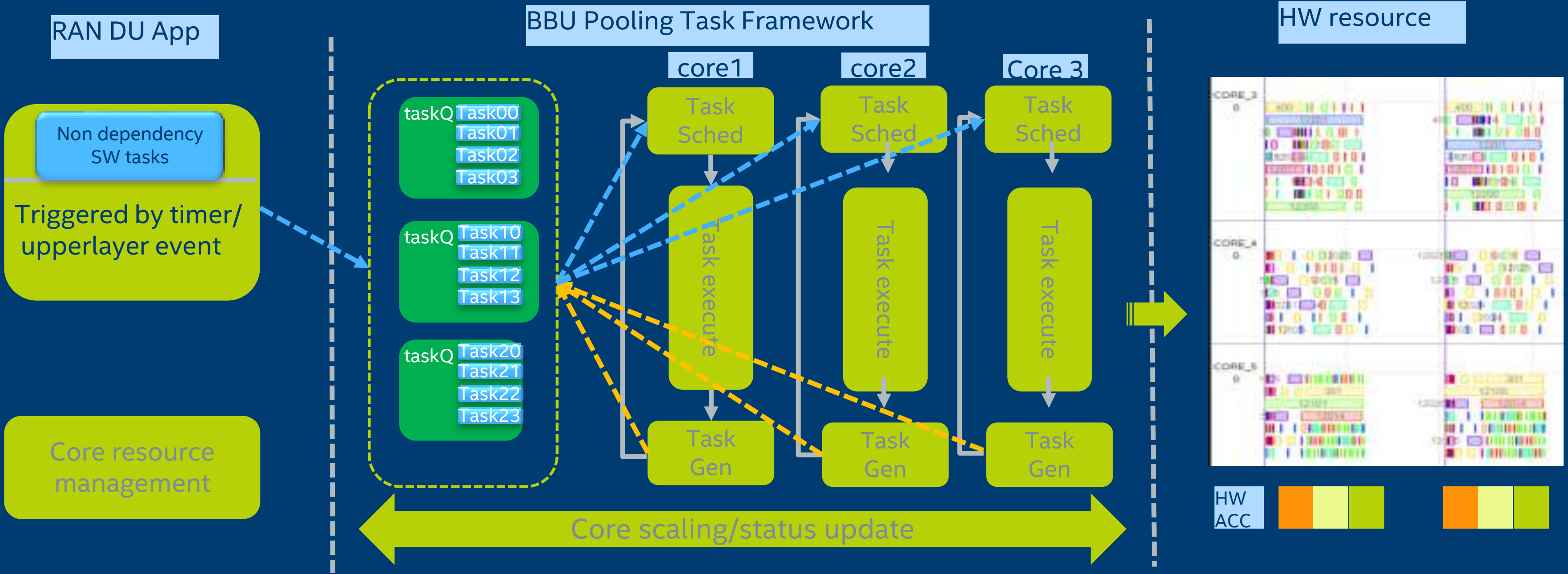


A task consists of the algorithms which is executed against the data.

A task might depends on another task, and the whole system consists of a chained tasks.

Tasks are prioritized, and time sensitive, it may or may not update dynamically.

Architecture



RAN DU App

Non dependency SW tasks

Triggered by timer/ upperlayer event

Core resource management

BBU Pooling Task Framework

taskQ Task00 Task01 Task02 Task03

taskQ Task10 Task11 Task12 Task13

taskQ Task20 Task21 Task22 Task23

core1

core2

Core 3

Task Sched

Task execute

Task Gen

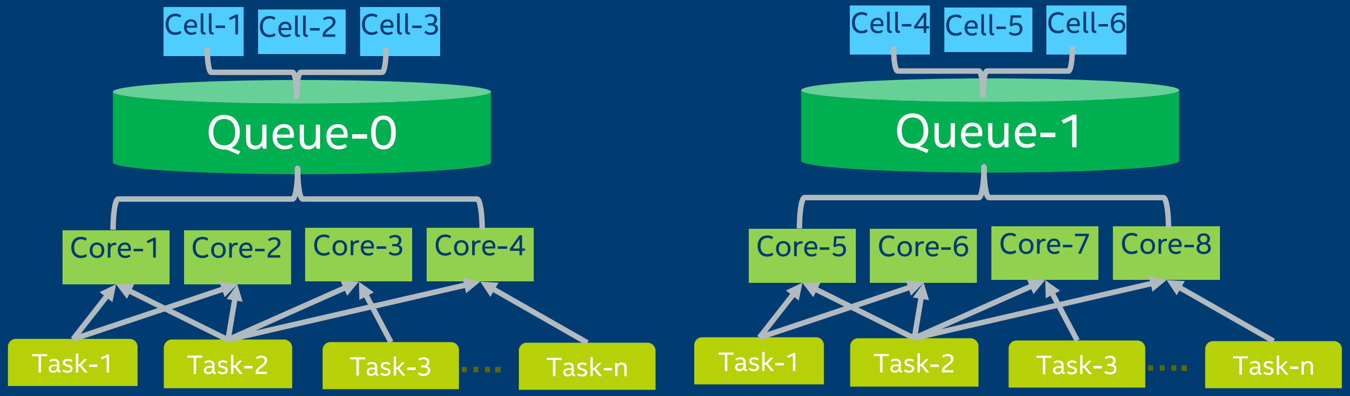
Core scaling/status update

HW resource

HW ACC

1. This architecture was proved the most efficient way to help RAN application to do parallel and pooling.
2. Did Enhancement from the latency, overhead, flexibility and power consumption perspective.
3. Moving forward to 5G requirement.

Hierarchical



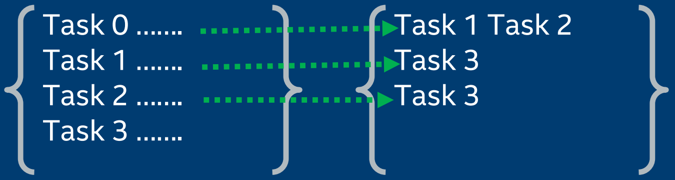
Flexible configure to support different requirement:

- Cell &Core group into one Queue, different QOS scenarios can be supported within different Queues separately;
- The number of queues can be configured;
- The relationship between queues can be configured;

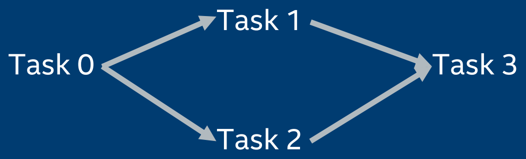
Functionality - Task Dependency

Task static info

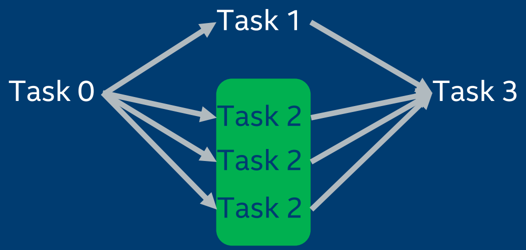
Next task info



Task static flow graph



Task dynamic split(loop parallel)



✓Task static information and next tasks information was registered into framework.

✓ Framework maintain the flow graph and put the tasks into task queues according the relationship automatically.

✓ support dynamic loop parallel computing through task split.

Functionality - Task Scheduler

Low overhead

Low latency

Proper priority

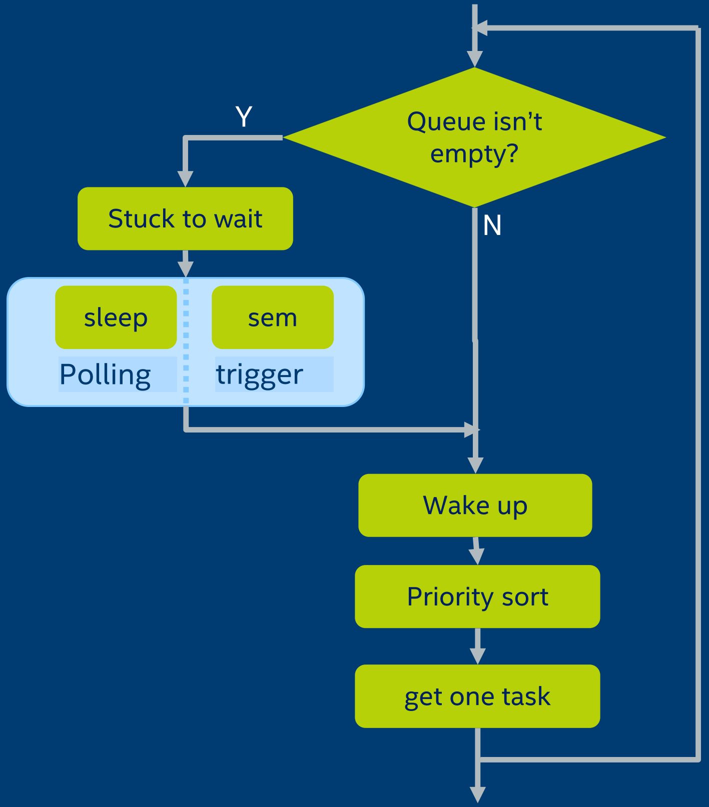
Parallelism

- task parallel execution on multi-core
- Built-in dependency management
- Non-block, no-long monolithic processing

Load Balance

- Automatic load balance between cores
- Integrated with IA power saving feature

Functionality - Task Scheduler

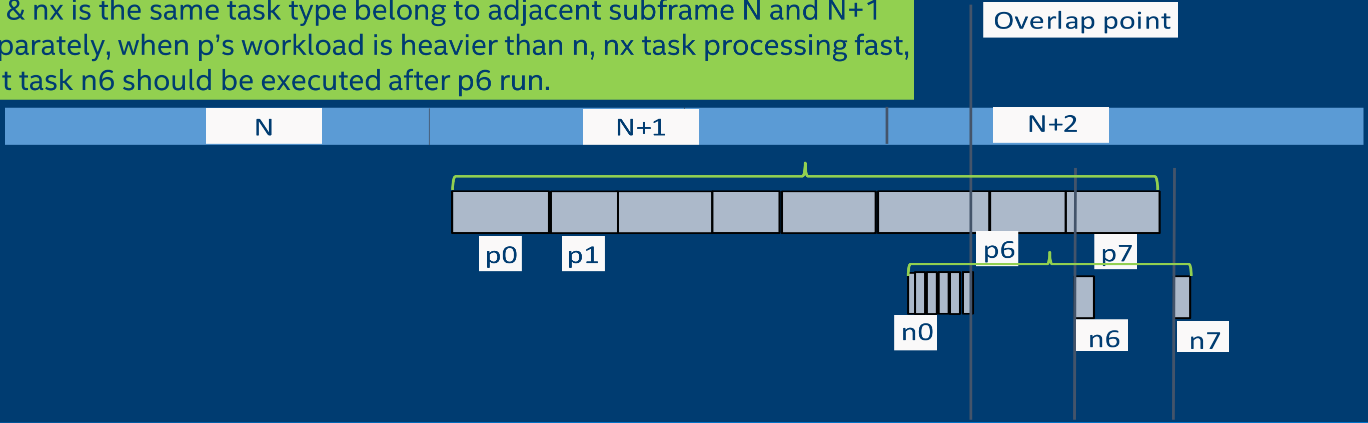


- ✓ Distributed task scheduler, each core run the same task scheduler procedure. Best way to achieve core load balance and core usage fairly.
- ✓ Change polling mode to trigger mode to reduce the power consumption and scheduler latency.
- ✓ Optimize Priority sort algorithm to reduce the scheduler overhead.
- ✓ Specify core waking up to reduce Lock overhead.
- ✓ Almost the highest priory tasks, try to find one task belong to the same cell with previous task
- ✓ Task Scheduler using two dimension priority
  - Dynamic priority: time related priority(early deadline first).
  - Static priority: tasks relative priority(highest priority task first).

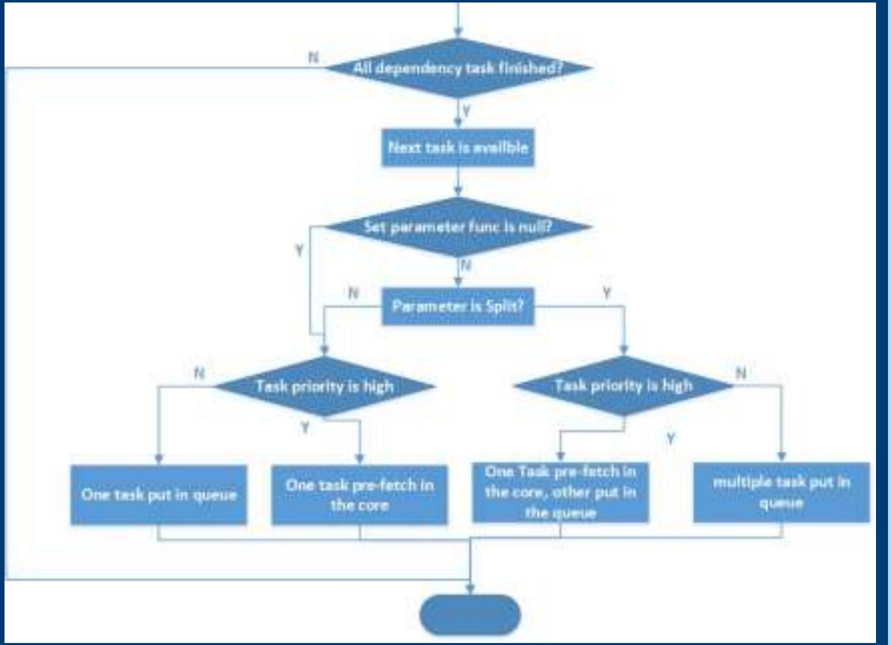
Functionality - Task Execution

- **Real-time system –** support deadline check for each task, bypass the missed task and following tasks .
- **Ordered processing –** provide radio based synchronization control for the task execution.
- **Recovery processing –** when task bypass or execution error, trigger recover scenarios if the task has provided recover function.

Example of synchronization control:
px & nx is the same task type belong to adjacent subframe N and N+1 separately, when p's workload is heavier than n, nx task processing fast, but task n6 should be executed after p6 run.

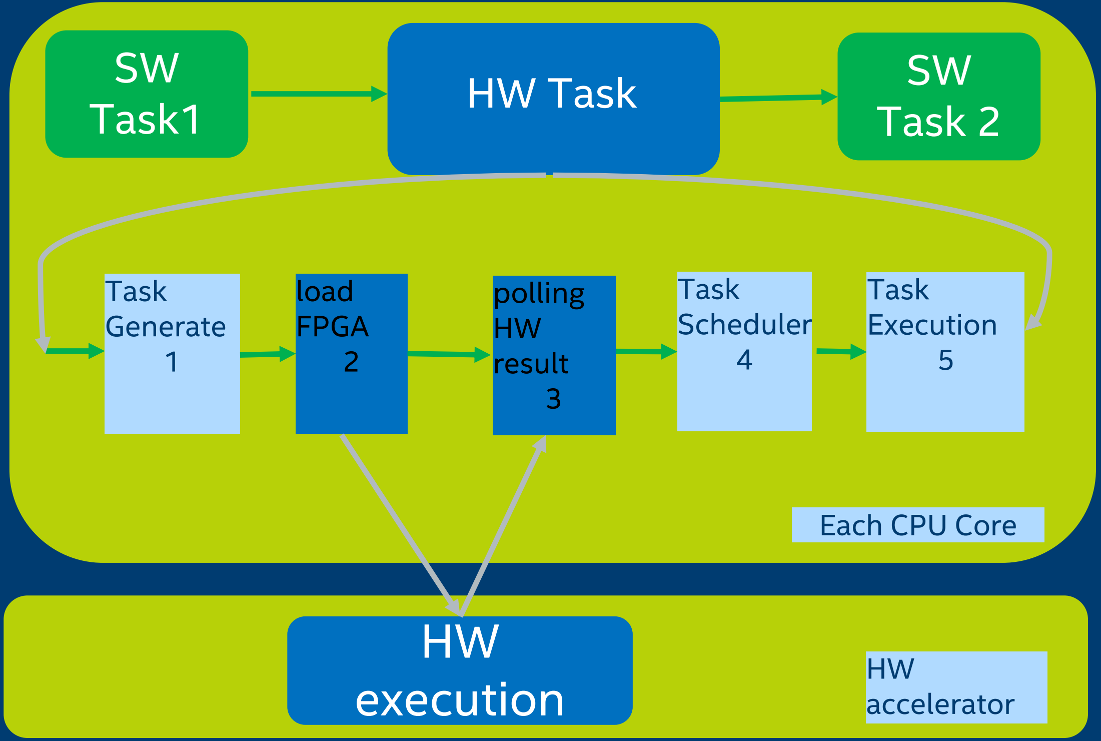


Functionality - Task generation


All dependency task finished?
N
Y
Next task is availble
Set parameter func is null?
Y
N
Parameter is Split?
N
Y
Task priority is high
N
Y
Task priority is high
N
Y
One task put in queue
One task pre-fetch in the core
One Task pre-fetch in the core, other put in the queue
multiple task put in queue

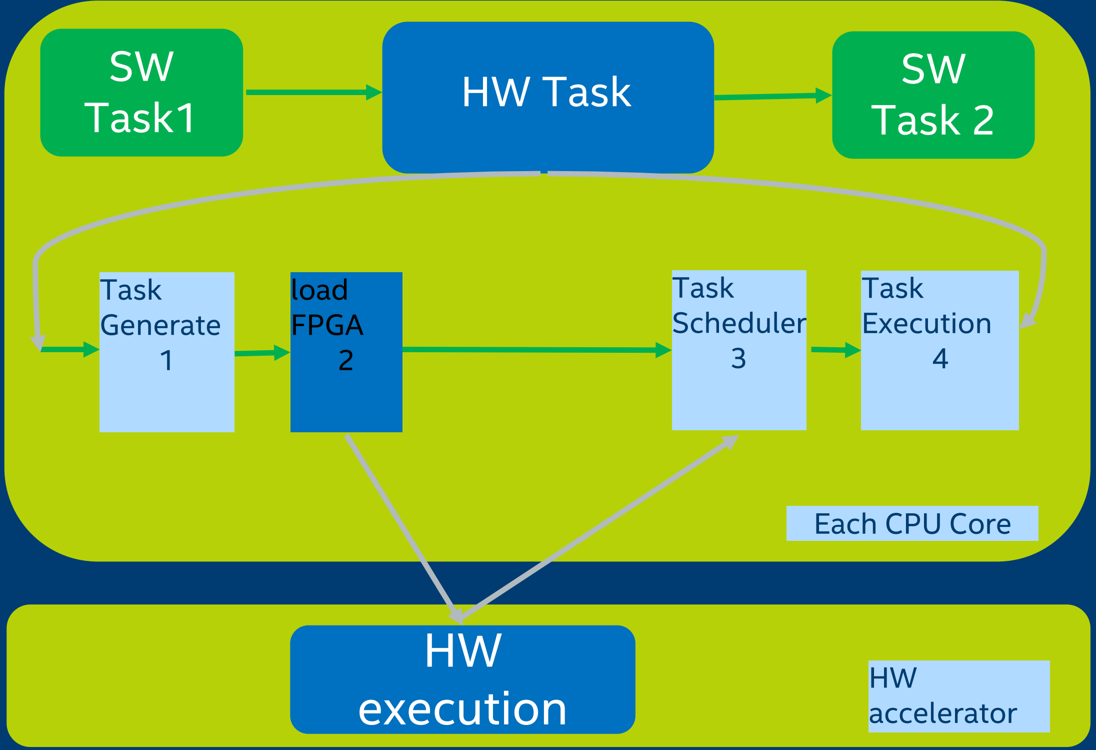
- ✓ Application Trigger: first task of a task chain or if it is not dependent on other tasks, it can be triggered only through the FlexRAN BBU pooling API by application.
- ✓ Auto-Generation: a task is dependent on other tasks and the dependency is registered, then this task will be generated, scheduled, and run automatically by framework.
- ✓ Task pre-fetch: In order to reduce cache miss and scheduler overhead, one task will be pre-fetch in the core and run immediately if the task's priority is high.
- ✓ Task Split: split tasks dynamically.

Functionality - HW task polling



1. HW acceleration or FPGA registered as HW tasks.
2. The relationship and priority with other tasks will still be maintain by framework.
3. send description(load FPGA) and polling function should registered into framework.
4. Task scheduler do polling when core is available(distributed polling).
5. Software task go through step 1,4,5 in the Framework, while HW task go through 1,2,3,4,5 in the framework

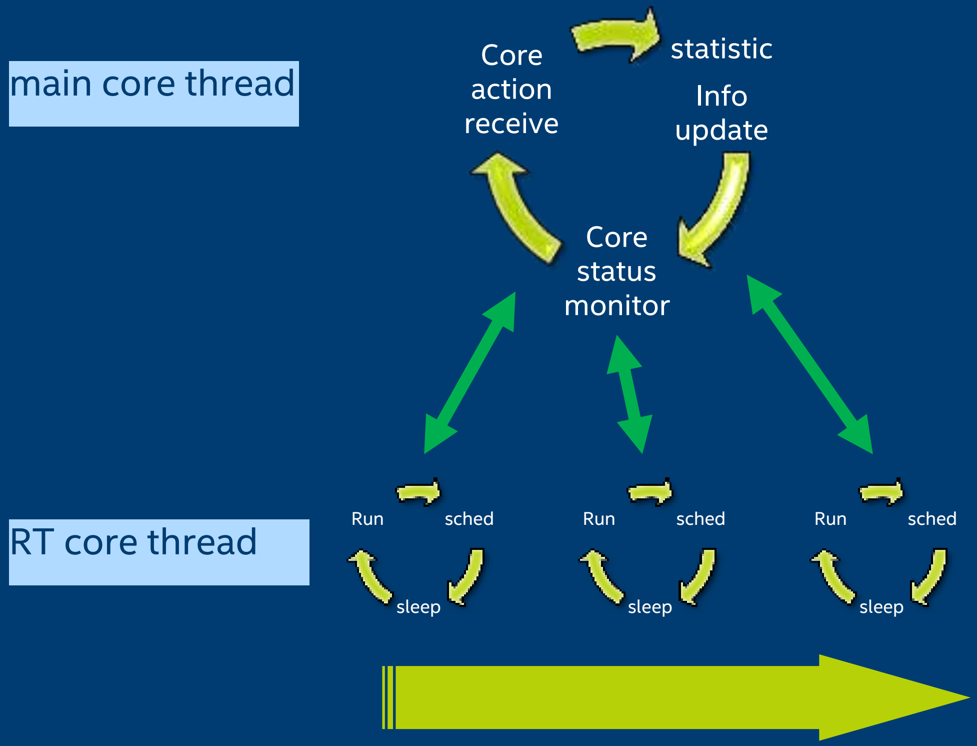
Functionality - HW task trigger mode



The same with Polling mode, except for:

1. Application do polling outside bbupooling or use interrupt mode to inform framework through calling callback function provided by framework after HW execution ready.

Functionality - Core Scaling & status



main core thread

Core action receive

statistic Info update

Core status monitor

RT core thread

Run

sched

sleep

Run

sched

sleep

Run

sched

sleep

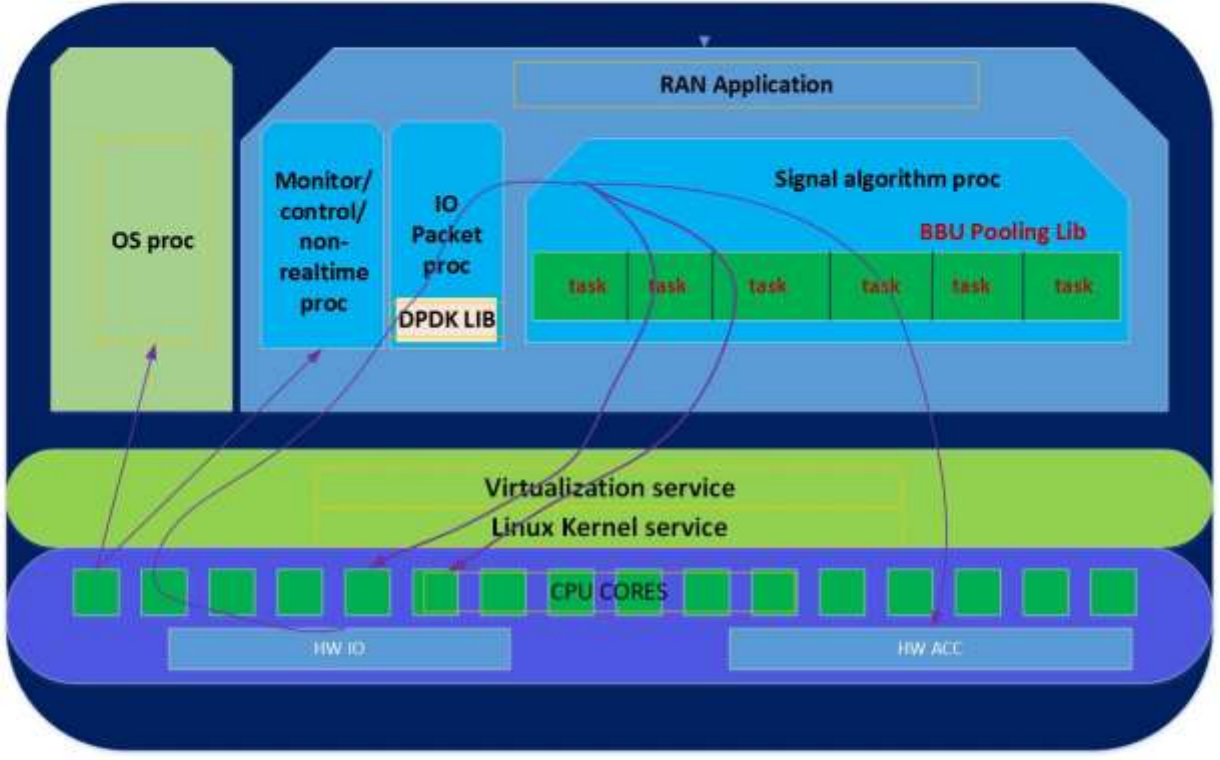
- supported core action:
  - ✓ Suspending
  - ✓ Add
  - ✓ Remove
  - ✓ Resume
- Reported core statistic information:
  - ✓ CPU load per core
  - ✓ Task execution time
  - ✓ Task latency time
- Core recovery:
  - ✓ Detect dead cores

Work Mode

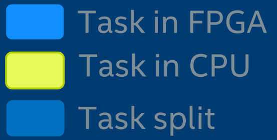
1. BBU Pooling Lib help to break down the computing intensive signal algorithm processing into CPU cores or HW ACC.

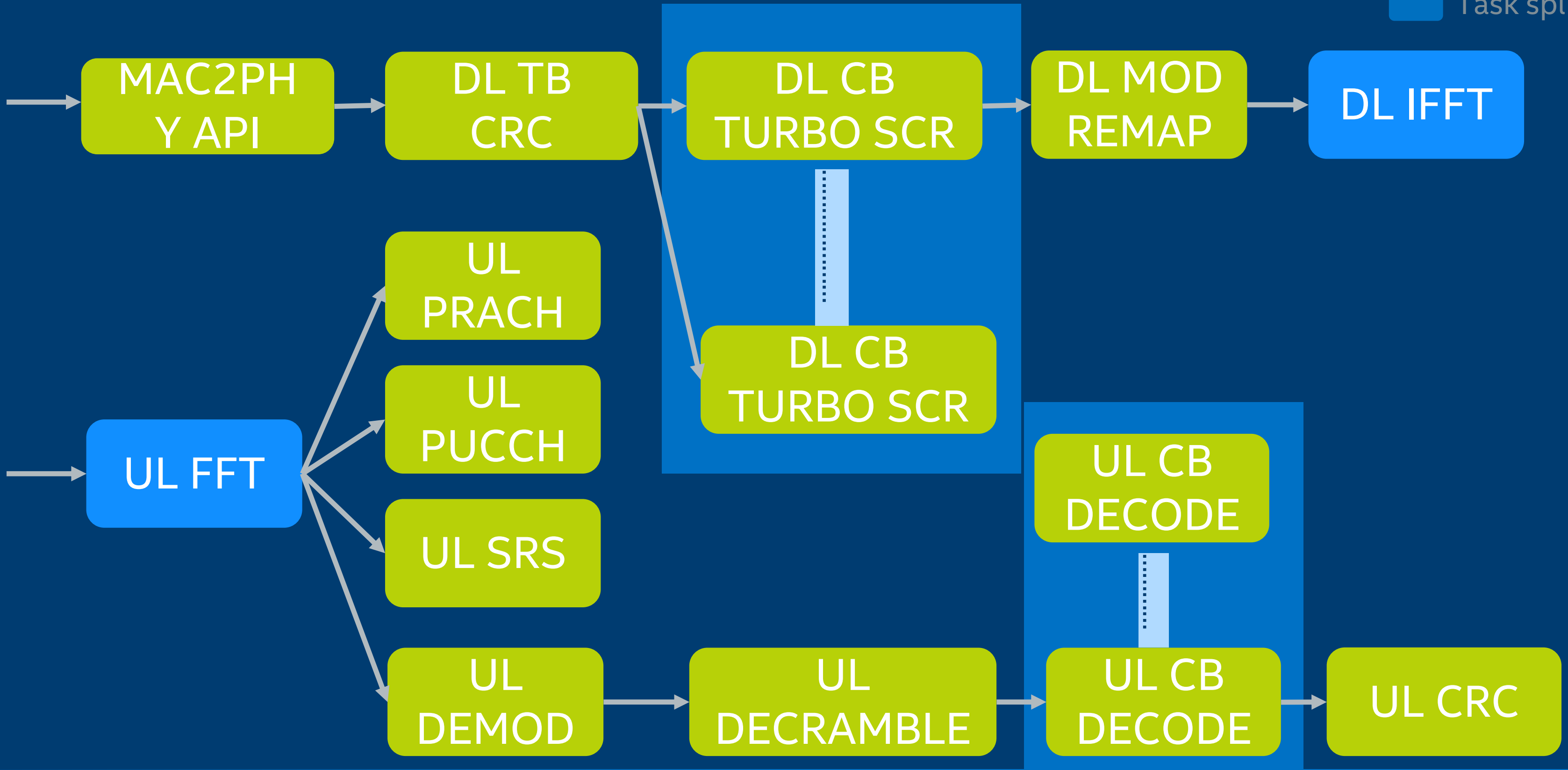
2. BBU Pooling lib help RAN application to be adoptable, compatible to different IA platform

3. BBU Pooling lib help RAN application manager the CPU resource dynamically, help application to use lowest clock on lowest number of cores at real-time to achieve power saving.



FlexRAN 4G Task Flow Graph

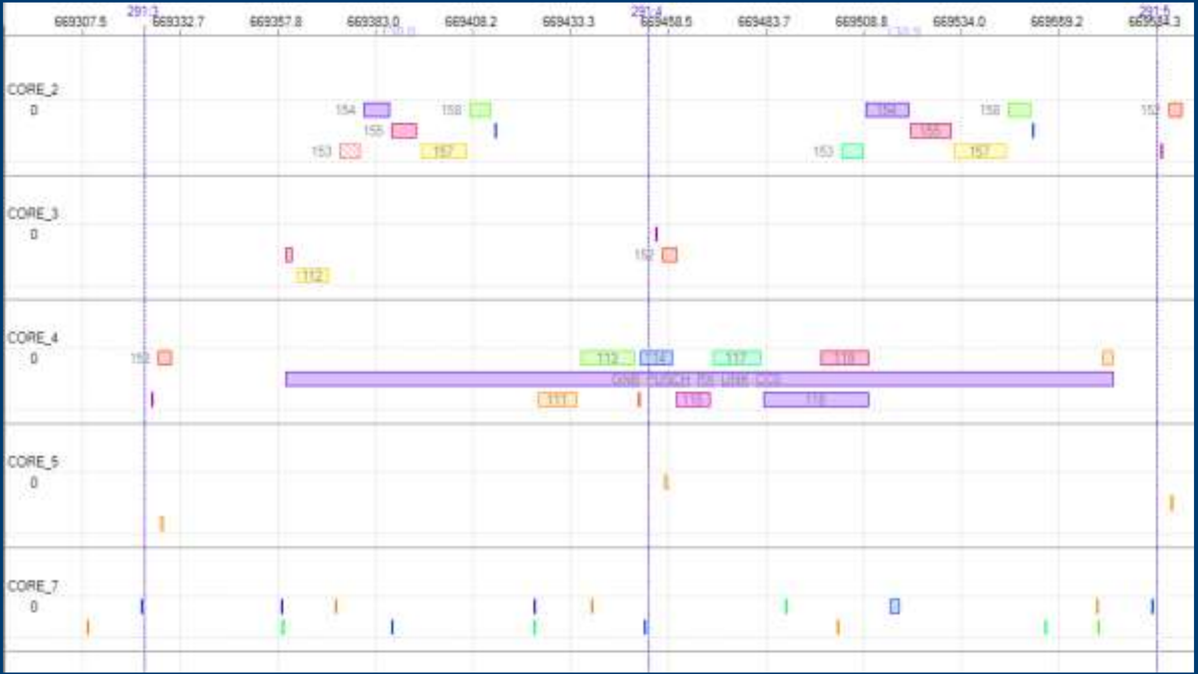




BBU Pooling Framework Usage

- The usage of the BBU Pooling framework as follow:
  - ✓ Define tasks that need to be run as part of the BBU Pool framework
    1. Must have unique task id (enum)
    2. priority
    3. call back function
    4. call back parameters
  - ✓ Define a task list (list of all tasks that have been written for parallelization)
  - ✓ Define the task dependencies.
- Initialize the BBU Pooling framework with the following information
  - ✓ Task list per cell and their dependencies
  - ✓ Number of cores used in BBU Pool for the processing.
  - ✓ Also set the BBU Pool thread priority and policy (if core is being shared with some other threads. This is not recommended of course)
- The BBU Pool framework also has a feature where a single task can be split into further smaller tasks (with same taskname and callback function but different callback parameters) so that we can further parallelize the tasks without having the need to create a new task and add it to the list.
- Define Number of cells being run in the system
- At TTI boundaries / other user defined locations, start the task list (starts the first few tasks that have no dependencies)

Debug Tool



Mlog give detailed profiling data as well as visualizations to show what function is running under what core and when in time. It is a powerful utility that is used for debugging real time issues and code optimization efforts.

Thank you!