From Embedded Systems to Scalable Platforms: Challenges in the Development of 5G baseband system on chip

Alan Gatherer PhD Senior Technical Vice President, Huawei USA
alan.gatherer@huawei.com
Huawei: a brief summary

170+ Countries
31 Joint Innovation Centers
14 R&D Centers
14 Regional HQs
45 Training Centers
180,000 Employees Worldwide

Revenue by Geography

- China: 50.5%
- EMEA: 27.1%
- Asia Pacific: 12.3%
- Americas: 12.3%

Revenue in $B USD

- 2009: 21.5
- 2010: 27.6
- 2011: 32.5
- 2012: 35.4
- 2013: 39.5
- 2014: 46.5
- 2015: 60.8
- 2016: 75.1
- 2017: 92.5

Audited by KPMG
Research Investment
Increase in Strategic Investments and Customer Focus Innovation

**Continuous Innovation Investment**


- Avg. $15M/yr
- CT Focus
- e.g. Cloud
- Avg. $75M/yr
- CT → IT
- Avg. $270M/yr
- ICT Focus

**R&D Investment in USA**

Unit: (USD in Millions)

- 2005-2007
- 2008-2010
- 2010+

<table>
<thead>
<tr>
<th>Year</th>
<th>Research Investment</th>
</tr>
</thead>
<tbody>
<tr>
<td>2008</td>
<td>1.6</td>
</tr>
<tr>
<td>2009</td>
<td>2.1</td>
</tr>
<tr>
<td>2010</td>
<td>2.7</td>
</tr>
<tr>
<td>2011</td>
<td>3.8</td>
</tr>
<tr>
<td>2012</td>
<td>4.8</td>
</tr>
<tr>
<td>2013</td>
<td>5.2</td>
</tr>
<tr>
<td>2014</td>
<td>6.6</td>
</tr>
<tr>
<td>2015</td>
<td>9.2</td>
</tr>
<tr>
<td>2016</td>
<td>11.0</td>
</tr>
<tr>
<td>2017</td>
<td>13.8</td>
</tr>
<tr>
<td>2018</td>
<td>15.2</td>
</tr>
<tr>
<td>2019</td>
<td>16.5</td>
</tr>
<tr>
<td>2020</td>
<td>17.8</td>
</tr>
<tr>
<td>2021</td>
<td>19.1</td>
</tr>
<tr>
<td>2022</td>
<td>20.4</td>
</tr>
</tbody>
</table>

- 2005-2007: Avg. $15M/yr
- 2008-2010: Avg. $75M/yr
- 2010+: Avg. $270M/yr
• Established the first US R&D office in 2001,
• HQ-ed in Santa Clara, CA
• ~900 staff in 9 R&D offices across USA

U.S. R&D Offices

Office

University
Connections

US R&D HQ
Santa Clara

San Diego
Phoenix
Denver
Chicago
Dallas
Seattle

WEST

SOUTHWEST

MIDWEST

SOUTHEAST

NORTHEAST

- UC Berkeley
- Stanford U
- UCSD
- UCLA
- UC Irvine
- UC Davis
- UC Riverside
- CalTech
- Santa Clara U
- UCSB
- UC Santa Cruz
- USC
- OSU
- WSU
- U of Washington

- Texas A&M
- UT Dallas
- UT Austin
- U of Arizona
- Arizona State U
- U of Houston
- Rice University
- Univ. North Texas
- SMU
- UT San Antonio
- UTEP
- U of Colorado

- Purdue U
- Univ. Minnesota
- Washington U
- U of Michigan
- U of Missouri
- U of Wisconsin
- UIUC
- U of Notre Dame
- Northwestern U
- Wayne State U
- Ohio State U
- Drexel U
- IIT
- U of Chicago

- NCSU
- Georgia Tech
- University of Florida
- Duke University
- UNC-Chapel Hill
- U of Virginia
- Virginia Tech
- Clemson U
- FIU

- SUNY – Bingham
- SUNY Buffalo
- U of Delaware
- U of Maryland
- Rutgers
- U of Rochester
- Manhattan College
- Tufts U

- NYU
- Princeton
- U of Penn.
- MIT
- CMU
- Cornell U
- Harvard U
- Columbia U
- Yale U
Part I: 5G the basics
5G: A Unique Time for Cellular Standards!!

Why?? We all actually want this one!!

1G
2G, cost reduction, really

2G
3G, data, but why do we need it?
Voice is good!!

3G
4G, fix 3G. Ooops..

4G
5G. Yes I want that!!
..lots of new market opportunities in V2V, IoT etc. Maybe we finally understand data? AND we have competition in a way we have never had it before

5G

HUAWEI TECHNOLOGIES CO., LTD.

Page 6
5G baseband in a nutshell

L2: optimize spectrum use

L1: air interface mod/demod

Spectrum Sharing
Different technologies co-exist in the same spectrum

Channel Sharing

Power Sharing

sub6GHz +mmW +unlicensed

L2 set up, isolation, channel management

Mostly accelerators
Antenna Space
Beam Space
Lots of FFT and RF Signal processing...

Quite soft
RU/RE space
User space
Lots of estimation

Hard
Bit space
Bit banging

Forced air cooling or convection => 10-30W/SoC

Memory dominated, Wireless channel => Firm real time

Multiple Radio Access Technologies (RATs)

100s of camped users
10s of pending users
a few active per TTI

Meaty real time DB problem
Emerging Market and Technology Drivers

Multi RAT, RAN as a Service

- A good schedule for one RAT may not make for a good schedule when another RAT is added. So we have a basic **RAT mixing problem** that has the potential to lose us significant performance if done badly (either in Normalistan or Extremeistan)

Need to start playing the statistics of RAT use to save costs

Requirement uncertainty from 5G

- leading to a softer modem and the need find strategies to mitigate the cost of softer IP. **Scheduling becomes a key technology**

New network topologies

- CRAN, MEC, Enterprise, Neutral Hosting must all be supported using a **single software architecture** strategy

Software Reuse

- General cost reduction in maintenance of baseband across multiple platforms is an issue that becomes more significant as we move to 5G and mRAT

End of Moores Law

- Performance improvement becomes unreliable. We need to find other paths to system performance improvement
Baseband: Not an Embedded System, not a Compute Server

**Normalistan**
- Traditionally we have treated it like an embedded system
  - Embedded system level of flexibility in architecture.
  - Software upgrade mainly to fix bugs

**Extremeistan**
- Compute Servers will be used in BBSoC architecture
  - close to baseband in architecture requirement?
  - addition of ML may make them more “computish”
  - the industry effort is massive!

The road from Normalistan to Extremeistan
- What issues prevent its immediate application?
- What added value do we get from it?

**Scale up**
- Embedded Systems
  - IP Performance optimized for algorithm

**Scale out**
- Compute Servers
  - Application Requirements unknown
  - Silicon optimized

**Support of heterogeneous IP and memory**
- Firm real time requirements

**Scale out, flexibility, composability, scalability, virtualization**
- General alignment with cloud architecture, intuitively a good thing..
The goals of the OTTs are different from the traditional operators:

- Find new customers to mine
- Increase marketing opportunities
- A data mining approach
- Not too fixated on cost of hardware
Part II: Finding Patterns in a Service Oriented Network
Can Cellular Infrastructure be just a big cloud app?

1. NFV: a classic example of virtualization, but there is difficulty in applying it to the RAT
   • How to manage real time? Nova resource is on the face of it not so efficient
   • Even Docker for RAT seems like overkill
   • Beginning to look like SDR!!

2. SDN for wireless: more useful sounding, app centric view of RAT. Integrate with SDN
   • How to deal with sounding, control and other common channels?
   • Benefit is to the app, so the benefit to the RAT is not so obvious (but that is OK)
   • Because that’s where the money is…but how do we do that?
   • In the future there will be a lot more rented infrastructure at multiple levels
   • Some operators want to become software houses
   • Actual infrastructure is just a burden; “shape of your money”, why take the risk?
   • Don’t let the OTT eat all the good bits
   • We cannot reach 1000X bandwidth but maybe we can reach 1000X experience!

3. Application/service driven network: customers pay for apps not bits
   • Because that’s where the money is…but how do we do that?
   • In the future there will be a lot more rented infrastructure at multiple levels
   • Some operators want to become software houses
   • Actual infrastructure is just a burden; “shape of your money”, why take the risk?
   • Don’t let the OTT eat all the good bits
   • We cannot reach 1000X bandwidth but maybe we can reach 1000X experience!

In search of pragmatism. Although active networks articulated a vision of programmable networks, the technologies did not see widespread deployment. Perhaps one of the biggest stumbling blocks was the lack of an immediately compelling problem or a clear path to deployment. A significant lesson from the active-network research effort was that killer applications for the data plane are hard to conceive. The community proffered various applications that could benefit from...
If the Infrastructure is an organism...

Recap: Sell services or apps, actual infrastructure is sooo 2000s ...
Unfortunately: Bad infrastructure will still kill you
Fortunately: Good infrastructure is still a great money maker and will be an essential part of the 1000X goal

Features are like genes

• Feature combinations may surprise us
  • MIMO for instance is not always a capacity improvement
  • The environment may change the answer for a features value
  • The application use may change the features value
  • The environment and application space are getting exponentially more complex!!
    • Now 5G wants us to support 3 distinctly different physical requirements

• Unlike evolution, we cannot afford a blind watchmaker......but adaptability is critical.
  • If the infrastructure is an organism then humans are the food!! Fortunately we know something about them.
Humans: more predictable than you might think..

Point #1: Only so many can fit in a box

- Our systems drop packets all the time it is OK to play the statistics on the hardware too.
- In fact this is one of the big arguments for CRAN

• Implication:
  - Operator sharing of resources is a guaranteed win if you are looking to reduce equipment, and yet we don’t do it. This is called capitalism…

• Caveat:
  - How many IoT devices will fit in starbucks??

• Crazy idea:
  - Shared hardware resources per unit time are bought and sold in some sort of commodity market
  - Spread the risk of hardware purchase
  - Reduce redundancy, save the planet....
  - Introduce a whole new class of traders??
Worst case loading of network hardware depends, but 25% would not surprise and it will only reduce into 5G.

Wave effect leads to strong negative correlations over the short term between close neighbors.

CQI prevents anything near “worst case” with high probability.

Real scenario (I can't show you)

**Human: more predictable than you might think.**

**Point #2: They don’t come from nowhere**

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Correlation Blue</th>
<th>Correlation Green</th>
<th>Correlation Blue+Green</th>
<th>Correlation Red</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.18</td>
<td>-0.56</td>
<td>-0.43</td>
<td>0.13</td>
</tr>
<tr>
<td>B</td>
<td>-0.25</td>
<td>-0.41</td>
<td>-0.42</td>
<td>-0.22</td>
</tr>
</tbody>
</table>

Worst case loading of network hardware depends, but 25% would not surprise and it will only reduce into 5G.
Humans: more predictable than you might think.

Point #3: They are basically insects, or wolves...

- In real data, large area average seen due long term user migration, the tidal effect
- Short term wave effect rides on top
- Wave effect is more useful
  - Short distance, short term
- Humans seem to move in well defined statistical patterns seen in bees, wolves, birds etc.

On the Levy-Walk Nature of Human Mobility

Injong Rhee, Member, IEEE, Munso Shin, Student Member, IEEE, Seongik Hong, Student Member, IEEE, Kyunghan Lee, Associate Member, IEEE, Seong-Joon Kim, Member, IEEE, and Song Chong, Member, IEEE
The future is gene therapy

• How do we spot a new genetic disease in advance? Big data of course!!
  (which is the answer to everything these days...)

<table>
<thead>
<tr>
<th>System modeling</th>
<th>Performance Analysis</th>
<th>Performance Prediction</th>
<th>Benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Analytics</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Black box approach</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scale</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Live – self learning/update</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Technology independent</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Need historical data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>What</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network-wide, automatic detection</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identify patterns (where, when, correlations)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network wide impacts:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Learn the Resource-KPI model</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Predict the KPI changes as a result of new algorithm etc.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Predictable results:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Learn and predict the network wide behavior from the data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enhanced operation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example application: Anomaly detection during Software Upgrade

Three key questions and Steps:

- Did we introduce new bugs?
- Did we fix the known bugs
- Did we achieve expected improvements?

Looks good?

Maybe not…
Part III: Services Run over the Cliff of Real Time

John Milton says I told you so…
Embedded System Philosophy Hits a Cliff

**Requirements**
- Past: Known Fixed
- Present: Known, mRAT
- Future: Evolving, mRAT

**Algorithms**
- Worst case, single scenario
- Worst case, multi scenario
- Resource aware, elastic multi scenario

**Architecture**
- Known, Worst, Use Case driven Hardwired as possible
- Worst, multi use case driven, resource sharing
- Need a “proper” scheduler

The entropy of mixing for an ideal gas

\[ n_1 V_1 + n_2 V_2 = n_{\text{mix}} V_{\text{mix}} \]
From Fixed to Opportunistic: Basic Challenges

Starting from a Classic Wireless Embedded System

Something like this is very confusing from an architecture perspective

Why? There is little apparent Opportunity to Schedule (OTS)

But... The high levels of dependency and real time constraints are apparent.

Starting from a Classic Compute Server Philosophy, NFV etc.

Ignores heterogeneity. Scheduling is spatial (NOVA) and coarse grained

Software overhead is ridiculous

Pretty much ignores scheduling data flow dependencies
From Fixed to Opportunistic: Basic Challenges

Starting from a GPU approach: DSL is great!! But what DSL?

Compute is easy. Data management is hard

Data dependency is easy. Real time constraints is hard.

Probably more than one DSL. Analysis tooling will be hard

“Innocence, Once Lost, Can Never Be Regained. Darkness, Once Gazed Upon, Can Never Be Lost.” John Milton. Same is true of real time….

Starting from a Classic Scheduling Approach

“I have a scheduling policy. Will it support the following real time problem?”

Scheduling is mostly temporal. Heterogeneity now being addressed

Schedulability continues to be mostly worst case. Leads to unacceptably loose bounds

Tend to focus on closed form solutions, but this is changing

Network Scheduling is statistical, but needs a lot of adaptation
Some Promising Approaches

Measurement based probabilistic timing analysis

MBTA to MBPTA. Data mine the crap out of it and build a model. Need enough data mined from the SoC. This is a challenge.

Bayesian Theory applied to scheduling. Need enough processors and jobs to create a statistic. Don’t schedule. Randomize and play the statistics.

Network Slicing Games: Enabling Customization in Multi-Tenant Mobile Networks

More generally connecting L2 and L1 scheduling. Consider the network statistics when scheduling L1 resources.

- OTS: You need many different ways to do the same thing
- Scheduling Granularity: Not too coarse, not too fine...
- OTS must produce more benefit than it loses due to “softening” of processing
Meanwhile…Compute Server Design Also Evolves: Machine Learn Everything

If you can automate device placement in the cloud you can automate SoC mapping and even scheduling.

If you can learn a better hash filter and reduce area by 50% you can do that for caches too.

Figure 13: Learned Bloom filter improves memory footprint at a wide range of false positive rates. (Here $W$ is the RNN width and $E$ is the embedding size for each character.)
Conclusions

- Compute Servers vs baseband: getting harder to tell the difference
  - They can learn from each other. Trick is to achieve real time and service orientation
- 5G is the most exciting standard ever
  - Service focused. Many new players and new ideas
- The service is the customer, the human is the food, we are the restaurant
  - We need to start seeing the patterns hidden in the network
  - These patterns change from location to location and service to service and over time
    - Yes of course ML will play a role....
- Real Time Support of Services is an unsolved problem
  - It will become more and more boutique in space and time.
  - How to adapt scheduling schemes? How to test??
  - Yes of course ML will play a role....
Shameless Plug for the CTN

- Alan Gatherer, Editor-in-Chief, comsoc.org/ctn
- Hottest Topics Delivered Monthly: 72K Opt-in Subscribers (as of 2016)

• Provides a quirkier, volunteer version of IEEE Spectrum, focused specifically on Communications Issues
• Tries to engage experts in hot topics and get a “read over coffee” level of article with references for the reader who is interested in further education.
• Publication via push email to the website once a month
Thanks!