Dynamic execution context management in heterogeneous computing environments, towards persistent computing environments

Serguei Boldyrev
Senior Manager
PhD, MBA

Location & Commerce BU
Common Technologies & Architecture

NVRAMOS 2011, November 9th
The nature of the consumer’s relationships with companies is changing

From a monologue... to a conversation... into continuous relationships

From a unified... to a segmented... Into a dynamic personalized offering together with ecosystem

Content and Consumer Lifetime Value are MAIN DRIVERS
Enhancing **user experience**

**past**
device only

**present**
device + service(s) + partners

**future**
Seamless User Experience across
device + services + partner ecosystem

Nokia
Lumia +
Maps +
Music

Nokia
N9+Maps
Towards inclusive and sustainable ecosystem

- Nokia targets 300M active Nokia Services users by the end of 2011
- 800M Cloud-compatible devices exist in 2012
- Persistent computing, energy and cost effective HW is on the vendor’s roadmaps, 2013 and onwards
  - ARM big.LITTLE, GPUs, SSDs NAND, PCM, MRAM
- Cloud Computing subscribers would total nearly one billion by 2014

A sustainable ecosystem where Nokia services will be only one part of the offering
Outline

• From Web of Information
• To Web of Computing
• Explore & Share
• Persistent Computing
• Opportunities & Challenges
• Data
• Computation
• Managed Performance
• Privacy
• Conclusions
From a Web of Information

- Cloud computing **today** serve email, apps, downloads and storage
  - Current "cloud apps" relying more on server-side processing
  - SaaS, PaaS, NaaS and other service models to emerge
  - Thin client, network as a computer, etc
  - Seamless information management between Client and Cloud

- Locate friends around you on a map
- Share favorite songs with friends
- Share content and micro-blog between friends
- Surface the most relevant content
Towards a Web of Computing

Between devices

Within Operators & Infrastructures

Scale up/down to/from Cloud

Seamless computing

Explore & Share

Between Devices & Infrastructure

©2011 Nokia
Because digital transformation of products and services are demanded by consumers, with increasing **complexity driven by** new customer **expectations**
Beginning: Explore & Share

Deliver services and content

Ovi at Retail concept (see in YouTube)

Share content between devices

Device to Device

Explore and Share new content in tags

RF memory tags
RF Memory tag technology

Motivation:
Evolution of non-volatile memories
=> from RFID tags to RF memory tags

Batteryless tags
Capacity ~1 Gbit

Mobile phone centric:
phone as the UI

-> 100 Mbit/s

-> Low cost
-> Low power
-> Simple
-> Minimize added HW

Applications: wireless memory in batteryless objects, ambient intelligence => Universal Local Storage
Positioning of wireless technologies

Operation range

- 100 m
- 10 m
- 1 meter
- 10 cm
- 1 cm
- 1 mm

Transfer speed

- 10kb/s
- 100kb/s
- 1Mb/s
- 10Mb/s
- 100Mb/s

EPC gen 2 operational area

NFC

UWB/RFID

WLAN, WI-FI, BTH, 3G

PASSIVE BATTERY-LESS TAGS

Physical contact required

HP Memory Spot

Sony's TransferJET

Universal Local Storage

Active technologies

60 GHz

©2011 Nokia
Explore & Share, architecture

Mobile phone

Application software

<table>
<thead>
<tr>
<th>Memory interface</th>
<th>Sensor interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network (Internet) memory stack</td>
<td>Internal memory stack</td>
</tr>
<tr>
<td>External memory stack</td>
<td>RFID stack</td>
</tr>
<tr>
<td></td>
<td>Bluetooth stack</td>
</tr>
</tbody>
</table>

Operating system

Phone hardware

Power

Data

Memory tag

RFID

Sensor tag

Cellular or WLAN

Network

UWBLEE MAC

UWBLEE PHY

I-UWB

WPT

Control block

Memory control layer

Sensor

Network

ATTACK

PHONE

UI/APP

MINAmi CONTROLLER

MEMORY SUBSYSTEM

MINAmi SUBSYSTEM

CONTROL BLOCK

MEMORY

CONTROL LAYER

NETWORK

UWBLEE MAC

UWBLEE PHY

I-UWB

WPT

PHONE RADIO SUBSYS

POWER SUBSYS

POWER MANAGEMENT

BATERY

I-UWB

WPT

MEMORY TAG

CONTROL BLOCK

MEMORY CONTROL LAYER

NETWORK

UWBLEE MAC

UWBLEE PHY

I-UWB

WPT

PCS

POWER MANAGEMENT

I-UWB

WPT
Explore & Share, functional blocks in

**ULS Architecture**

- Reader memory (PCM memory)
- ULS buffer (over the air part of the memory), memory what is coming from air (bidirectional)
- External tag
- OS (demanded from OS)
- eTag (in PCM), visible for external device

• Received Data first to ULS buffer
• Then ULS Server mark that part to eTag memory (if needed)

OS = Operating system
Etag = Embedded tag, tag in the phone
PCM = Phase Change Memory
EMC = External Memory Ctrl
SDR = Synchr. DRAM
ADmux = Addr, Data mux
LPDDR1 = Low power SDRAM Dual Data Rate Interface
ULS v. x.y = e.g. ULS IC v5, UMETAG IC version, that provides RF Frontend MAC functionality to create over the air connectivity with 1-UWB and UHF/NFC powering.

Reader/Writer

ULS Server, also called as MINAmI Server

Local reflection of the Distributed execution memory block

Distributed execution memory block

“Over the air buffer”
(Rx buffer & Tx buffer)
Explore & Share, Computation migration, sort of 😊

Mobile device, handheld form factor

- CPU
- XiP
- NVRAM
- RF

Stationary device, notebook form factor

- RAM
- CPU

Client API

Computation run-time Environment

Distribution

Agent 1

Agent 2

Client A

Client B
Step #1  Approaching
Mobile device, handheld form factor

CPU  XiP  NVRAM  RF

No link

Stationary device, notebook form factor

RF  CPU  RAM

Step #2  Suspending
Mobile device, handheld form factor

CPU  XiP  NVRAM  RF

Link is going up

Stationary device, notebook form factor

RF  CPU  RAM

Step #3  Dispersing
Mobile device, handheld form factor

CPU  XiP  NVRAM  RF

Link is going up

Stationary device, notebook form factor

RF  CPU  RAM

Execution context aggregated/running
Execution context suspended
Execution context dispersed
Step #4
Migrating

Mobile device, handheld form factor

CPU
XiP
NVRAM
RF

Stationary device, notebook form factor

CPU
RAM

Link is up

Step #5
Aggregating

Mobile device, handheld form factor

CPU
XiP
NVRAM
RF

Stationary device, notebook form factor

CPU
RAM

Link is up

Virtual run-time environment

Step #6
Resuming

Mobile device, handheld form factor

CPU
XiP
NVRAM
RF

Stationary device, notebook form factor

CPU
RAM

Link is up

Virtual run-time environment

Execution context aggregated/running
Execution context suspended
Execution context dispersed
Step #7
Suspending

Mobile device, handheld form factor

Link is going down

Stationary device, notebook form factor

Step #8
Dispersing

Mobile device, handheld form factor

Link is going down

Stationary device, notebook form factor

Step #9
Migrating

Mobile device, handheld form factor

Link is going down

Stationary device, notebook form factor
Step #10
Aggregating

Mobile device, handheld form factor

Stationary device, notebook form factor

Step #11
Resuming

Mobile device, handheld form factor

Stationary device, notebook form factor

Step #12
Leaving

Mobile device, handheld form factor

Stationary device, notebook form factor

Execution context aggregated/running

Execution context suspended

Execution context dispersed
Ideal Scenario for the Future Solution

- Mobile device is a true contributing participant, not just a depending client
- Mobile device connects to multiple data sources (possibly independent devices or clouds), forming one logical Computation/Data and Service space
- From any connected device, the user deals with the same data regardless of its physical location

- Solution provides both data and distributed computing services
- Solution access is managed by fine-grained security policies applied according to the user’s context
- From the design perspective, the Solution manages and hides complexity; does not increase it
- Solution data and services decrease development and maintenance costs
- Solution data and service pricing model creates new money flows between players
New Digital Experience

"My digital experience follows me regardless the computing environments around me"

• take advantage of finely granular and accountable processing services,

• integrate heterogeneous information sources, and

• ultimately free users of mundane challenges and details of technology usage

Greater reuse of information and computational tasks is just few steps away
Opportunity of New Digital Experience

World of fully deployed and available Computing – Persistent Platforms

The emergence of Platforms that can seamlessly span the “information spaces” of multiple
• hardware,
• software,
• information and
• infrastructure providers
Technological enablers for Persistent Computing

Utilizing the best parts of the Web development paradigm
- ...Web-based applications, not Web sites...
- and leveraging the special contextual capabilities of mobile devices (ex. WURFL databases on sensors)

Moving from a Device, Cloud and Web-centric models to a Persistent model (Persistent computing)
- Reuse (partition) information and persistent computation in the Cloud, Infrastructure and Devices
- Elimination of special-purpose software to download, install applications, Service oriented infrastructure constructed as a functional flows requested on demand
- Enabling balancing of computation and relevant data between heterogeneous Cloud Back-Ends, Infrastructures (ex. Spanner, HDFS, Web Intents etc) and devices

Fostering faster, easier, richer application innovation and deployment through SW and HW Persistent computation recycling
- AMP processors, non-volatile execution memory
- Diverse Cloud Back-Ends can all leverage the infrastructure capabilities
- Atomic computations are now deployed once to the Back-Ends and composed down to Infrastructure and clients, where a set of functional flows or description of those form the actual service
- Allowing Network Infrastructure to leverage data and computation workload, by taking care of services or provider of services capabilities, distributing computation between Back-Ends and Infrastructure
- Allowing more efficient contextual composition of services than purely device or Web-centric models
Persistent Computations

- A next generation distributed systems are dynamically spanned around virtual and physical infrastructures
- Device becomes a known and contributing node of the cloud infrastructure, service provisioning element

Examples: Services based on crowdsourced information on traffic, pollution, weather, body temperature, context intelligence, environments adapting to user context, efficient computing
Persistent computing, Requirements

The new emerging paradigm of Distributed Systems requires an explicit orchestration of computation across

- Mobile Devices (D2D connectivity)
- the Edge, (thin elements of the Cloud “skin”)  
- Cloud and
- the Core

Orchestration depends on

- components’ states (persistency!) and
- resource demand,

through the proper exploitation of granular and therefore accountable mechanisms
“Aha!” moment

Design framework should be re-defined!

Persistence is the key attribute of ALL future Data & Computation Management systems
Opportunities and Challenges

Main aspects of

• **Data and Computation** semantics,

• **Performance and Scalability** of computation and networking,

• **Role of Security and Privacy**
Data - Common Data Model (CDM) enables Shared data across application domains

Client Applications should not “own” data, instead, they should reflect the users’ desire to do something, to accomplish certain goals
- to call someone, you need Contacts,
- to know what to do next, you need Calendar, etc.

Data is still the same...
- for example, it would be helpful for Contacts to know who is the organizer of your next meeting
- EXAMPLE: Apple Siri, Google Chrome
Shared data, layers view

Layer IV: Reasoning
- self-description
- reasoning

Layer III: Metamodel
- triple store + SPARQL queries
- “generic operations” on data

Layer II: Relations
- pre-computed indices linking objects
- path queries for traversal from object to object

Layer I: Objects
- simple objects stored in a key/value store
- object access via globally unique IDs

Shared data demands
- steep learning curve (new ways to think about apps, platform, UX, …)
- represents a shift from “device-centric” and “Cloud” thinking to “Future” thinking
- risks, uncertainties, … (also big potential reward)
Computation and Data perspective

Prerequisites
Core Data assets are encapsulated into results of Computation tasks

Objectives (but not limited to listed below)
• Design and implementation of scenarios where computation can be described and represented in a system-wide understandable way
  • enabling computation to be “migrated” (transmitted) from one computational environment to another
  • such transmission would involve reconstruction of the computational task at the receiving end to match the defined semantics of task and data involved

• Construction of a system where larger computational tasks can be decomposed into smaller tasks or units of computation
  • independent of what the eventual execution environment(s) of these tasks (that is, independent of hardware platforms or operating systems)
Computation, the components

Computation closure –
- current state of execution and context
- automatically generated during the development phase
- composed out of anonymous function object

Function object – consists of:
- class template *result_of*,
- function template *mem_fn*,
- function object binders, and
- polymorphic function wrapper function
Treating Closures as Data entities, letting us compose Closures, chain them, and project computations into them.
**Computations, User space**

### Applications and Back End logic

1. **User Computations**
   
   Object \(<\text{compute}_1, \text{compute}_2, \text{compute}_3, \ldots>\)
   
2. **Selection of the context** (functional chain)
   
   \(\text{compute}_3 = \text{filter(object)}\)
   
3. **Computation distribution**
   
   \(\text{put(\text{compute}_3)}\)

### User Device

5. **Project computation to the Functional chain with User device context**
   
   \(\text{object} = \text{project(\text{compute}_3)}\)

6. **Reconstruct computation**
   
   \(\text{context\{object<\text{compute}_3>\}}\)

### Common Data Platform

- Persistent
- Granular and Reflective Process Representation
Distributed information management framework enables reflective process/context migration

- Marshalling/UnMarshalling
- Alignment and enhancement with distribution

Distributed Computation and Data model allows dynamically balanced load between concurrent execution environments taking into account the user’s current context
Persistent Computations

Distribution/sync

Front-End Side

Computation environment

URIs

0xFFFF

Legacy routines

0x0

Back-End Side

Computation environment

0xFFFF ... FFFF

URIs

Legacy routines

Computations SUPERSET

Persistent Computations based platform

Computations subset

0x0

Legacy routines
Construction of the computational ontology

- Core parts contain generic definitions for closure behavior
- Environment specific extensions as specialization of core parts
- Matching between the environment specific extensions
  - Variation in closure ontologies
  - Limited number of specializations
- Also non-functional properties
  - energy
  - performance
  - ...
- Moving closures states from e.g. Back-End Java/PHP/C++ to pure Qt/QML/Java environment and vice versa
Developer Experience – integration with SDK

Closures allow to compose function expression that can be wrapped and exposed to Developer as Data.

Closures can be passed to API, stored, shared or spread the computation.

- The best estimated result based on the first returning closure can be seen as a part of functional flow.
- When closures “return”, results can be refined (parallelism).
- Computation recycling is possible.
- Builds up a cache of results/partial results (“fast computation”).
Each branch represents Computation associated with Data, where Computational model is encapsulated into Data model.

Before Starts:

**Parameters to know beforehand (before start invoked):**

1) Capabilities
2) Functional Flow specification (of how many parts we have in the map), see legend in this slide (right).
3) Cost functions (compound function): what are cost in order to bind next Computation
4) Rules (entity that drives this connector Computation and if not then it directs elsewhere)

=> First list the 1-4 parameters then proceed with load balancing phases.
Performance/Scalability, Requirements

- Granular latency control for diverse Data and Computational load in heterogeneous environments
- Resolution of short term capacity decisions, including
  - the determination of optimal configuration,
  - the number of (live) Persistent nodes or
  - the migration of computation or data
- Resolution of long term capacity decisions
  - decisions concerning the development and
  - extension of data- and service architecture, choice of technology, etc
- Control of SLAs at different levels,
  - e.g. consumer services, platform components, etc
- Quality assessment of distributed software architecture
Test setup, SSD based

- We are running tests on MLC SSDs for the production Common Data Platform
- Initial results from performance testing
  - very promising 😊,
  - next wear testing the drives
- Performance test from 3 node cluster with 5-6 MLC SSDs on each

- 3 node cluster to over 8000TPS with constant sub 20ms latency
- safe load to achieve ~5ms
- 90% latency is around 6000TPS
- Max TPS is way higher but latency grows to 10-20ms with it (can be adjusted by playing with concurrency)
Next steps PCM/MRAM evaluation

Memory blocks modules could be packed in
1. xGB PCIx board with Memory blocks in single server (small size frontside cache)?
2. Mountable as a block device (due to SW reasons), SATA or PCI etc.?

So far, SSDs sounded like a perfect fit for reducing our latencies and cost for the Data Endpoints, but we could improve it even further and enable certain number of new business case
Motivation

Why we are doing ALL this?
User, Trust, Privacy the placement and scope

Content and Consumer Lifetime Value are MAIN DRIVERS

(note that in this diagram, by “ontology” the semantics is meant)
Conclusion - Driving agenda

• Creation of **approaches** and effective **tools** for the next developer and consumer applications, tackling
  • **Fuse & Lift of Big Data** (Bring algorithm to the data, Knowledge creation)
  • Social graph oriented patterns (social media, chat, calendar, etc)
  • Tele immersive environment (AR, 3D maps, etc)
• Persistent Computing **platform** for the Continues Operations
  o the **fine-grained** (and thus easily accountable) software **frameworks** developments
  o anything as a persistent service, e.g. sometimes giving hardware for “free”
• Mobile devices supported by Backend **infrastructure** technologies to enable qualitative leap in ICT industry offerings, from the perspectives of
  o **scalable** technology and
  o **multisided** business models with high elasticity

**Aiming and supporting future strategic growth areas and maintenance requirements**
Q&A

sergey.boldyrev@nokia.com
Thank you