High-performance computing in Java: the data processing of Gaia

X. Luri & J. Torra
ICCUB/IEEC
Outline of the talk

• The European Space Agency
• Gaia, the galaxy in 3D
• The Gaia data processing and analysis consortium
• The Gaia data processing: high-performance computing in Java
**The European Space Agency**

ESA was created in 1975 by merging two previously existing organizations: ESRO (satellites) and ELDO (launchers) with the aim of becoming Europe’s independent space agency.

It’s presently integrated by 18 member states.

Canada participates in some projects through a cooperation agreement.
# High-performance computing in Java: the data processing of Gaia

<table>
<thead>
<tr>
<th>ESA Members</th>
<th>Mandatory Contr.</th>
<th>Optional Contr.</th>
<th>Total (£ mill.)</th>
<th>Total (%)</th>
<th>National Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>France</td>
<td>15.63%</td>
<td>31.55%</td>
<td>778.8</td>
<td>26.38%</td>
<td>CNES</td>
</tr>
<tr>
<td>Germany</td>
<td>23.41%</td>
<td>21.45%</td>
<td>614.8</td>
<td>20.83%</td>
<td>DLR</td>
</tr>
<tr>
<td>Italy</td>
<td>12.88%</td>
<td>14.59%</td>
<td>397.9</td>
<td>13.48%</td>
<td>ASI</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>16.93%</td>
<td>5.91%</td>
<td>239.3</td>
<td>8.11%</td>
<td>BNSC</td>
</tr>
<tr>
<td>Spain</td>
<td>6.87%</td>
<td>5.76%</td>
<td>169.0</td>
<td>5.72%</td>
<td>INTA</td>
</tr>
<tr>
<td>Belgium</td>
<td>2.83%</td>
<td>7.37%</td>
<td>167.4</td>
<td>5.67%</td>
<td>BIRA</td>
</tr>
<tr>
<td>Switzerland</td>
<td>3.40%</td>
<td>3.49%</td>
<td>97.3</td>
<td>3.30%</td>
<td>SSO</td>
</tr>
<tr>
<td>Netherlands</td>
<td>4.43%</td>
<td>2.67%</td>
<td>90.9</td>
<td>3.08%</td>
<td>SRON</td>
</tr>
<tr>
<td>Sweden</td>
<td>2.61%</td>
<td>2.11%</td>
<td>62.5</td>
<td>2.11%</td>
<td>SNSB</td>
</tr>
<tr>
<td>Austria</td>
<td>2.26%</td>
<td>0.87%</td>
<td>33.7</td>
<td>1.14%</td>
<td>ASA</td>
</tr>
<tr>
<td>Norway</td>
<td>1.70%</td>
<td>1.02%</td>
<td>33.2</td>
<td>1.12%</td>
<td>NSC</td>
</tr>
<tr>
<td>Denmark</td>
<td>1.82%</td>
<td>0.78%</td>
<td>28.8</td>
<td>0.98%</td>
<td>DNSC</td>
</tr>
<tr>
<td>Finland</td>
<td>1.37%</td>
<td>0.54%</td>
<td>20.7</td>
<td>0.70%</td>
<td></td>
</tr>
<tr>
<td>Ireland</td>
<td>0.95%</td>
<td>0.30%</td>
<td>12.8</td>
<td>0.43%</td>
<td></td>
</tr>
<tr>
<td>Portugal</td>
<td>1.40%</td>
<td>0.21%</td>
<td>12.7</td>
<td>0.43%</td>
<td></td>
</tr>
<tr>
<td>Greece</td>
<td>1.50%</td>
<td>0.12%</td>
<td>12.5</td>
<td>0.42%</td>
<td>ISARS</td>
</tr>
<tr>
<td>Luxembourg</td>
<td>0.21%</td>
<td>0.13%</td>
<td>4.2</td>
<td>0.14%</td>
<td></td>
</tr>
<tr>
<td>Czech Republic</td>
<td>???%</td>
<td>???%</td>
<td>5.1</td>
<td>0.17%</td>
<td>CSO</td>
</tr>
<tr>
<td>Total Member states</td>
<td>100%</td>
<td>99.06%</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Total ESA</td>
<td>-</td>
<td>100%</td>
<td>2951.8</td>
<td>100%</td>
<td></td>
</tr>
</tbody>
</table>
High-performance computing in Java: the data processing of Gaia

**ESA’s space science**

- The space science projects have proven in the last 34 years the scientific benefits of the multinational cooperation

- ESA’s areas of work:
  - Earth’s space environment
  - Sun-Earth interaction
  - Interplanetary medium
  - The Moon and the planets
  - The stars and the universe
Gaia, the galaxy in 3D
**Gaia history**

- Gaia is the successor of the Hipparcos satellite, the first space astrometry mission. The Hipparcos catalogue is today an essential reference in astronomy and has led to more than 1600 refereed publications since 1996

  http://www.rssd.esa.int/index.php?project=HIPPARCOS&page=science_results

- Gaia is the *Cornerstone 6* in the frame of ESA’s “Horizon 2000+” program. It was approved in 2001 and its launch is scheduled for 2012.
Gaia: an astrometric mission

Will provide the most complete 3D survey of objects in our Galaxy (and beyond)

- >10^9 objects (~1% Milky Way)
- Complete up to 20th magnitude
- Positions, velocities and parallaxes
  - Nominal precision (15th mag): ~25μas
- Spectrophotometry
- Spectroscopy and radial velocities (G<16)
- No input catalogue → unbiased survey
Nominal precision in parallax ~25 μas and proper motions 25 μas/yr

25 μas → measurement of a 4cm object on the Moon as seen from Earth

25 μas/yr → measurement of the nail growth of an astronaut on the Moon as seen from Earth
Spacecraft & payload

Two SiC primary mirrors
1.45 × 0.50 m² at 106.5°

Rotation axis (6 h)

Basic angle
monitoring system

SiC toroidal
structure
(optical bench)

Superposition of
two Fields of View
(FoV)

Combined
focal plane
(CCDs)
Focal plane

106 CCDs, 938 million pixels, 2800 cm$^2$

Sky Mapper CCDs

Astrometric Field CCDs

Image motion
Gaia’s main aim: unravel the formation, composition, and evolution of the Galaxy.

Key: stars, through their motions, contain a fossil record of the Galaxy’s past evolution.
**Main scientific goals**

- Structure and kinematic of our galaxy
- Stellar populations
- Tests of the galactic formation

⇒ Origin, Formation and evolution of the galaxy

**Additional goals: stellar astrophysics**

- Stellar astrophysics
- Multiple stellar systems
- Solar System objects
- Extrasolar planets
- General relativity
- Galaxies & QSOs
The Gaia Data Processing & Analysis consortium

Gaia

Data Processing & Analysis Consortium

Science Operations Centre (and DPC)
Mission Operations Centre
Data Processing Centre

Institute of Astronomy, Cambridge
ESOC, Darmstadt
Geneva
CNES, Toulouse
IUOH Observatories
ESAC, Madrid
Barcelona
**Data Processing and Analysis Consortium**

- Formed to answer the Announcement of Opportunity (AO) for Gaia data processing

- Involves large number of European institutes and observatories (>300 people)

- The science community must fund the majority of the Gaia processing (not ESA)
DPCs underpin and support the processing
– Software support and production
– Operation of processing system(s)

- ESAC (CU1,3) Madrid
- BPC (CU2,3) Barcelona
- CNES (CU4,6,8) Toulouse
- ISDC (CU7) Geneva
- IoA (CU5) Cambridge
- OATO (CU3) Torino
Gaia data processing in a nutshell

• Complex algorithms
• Distributed processing
  – Six European wide DPCs
  – Local algorithms must be distributed
  – Mostly embarrassingly parallel
• Large quantity of data
  – All data accessed repeatedly
  – Heavy data exchanges between DPCs
• No users – no security needed
• Naïve approaches have proved impossibly slow
• This requires Thought and Work.
High-performance computing in Java: the data processing of Gaia

[U1] IDT/FL inputs to MCS – Planning/Calibration
[U2] MOC satellite uplink, TC, Calibrations
[1.1] HK from Gaia to Mission Control System (MCS)
[1.2] TM delivered to Initial Data Treatment and First Look at ESAC directly from ground station
[2] HK data delivered to DDS.
[3.1] IDT/FL Orbit and Aux Data from DDS
[3.2] IDT/FL inputs from Main DB (Obs, Calibration)
[4] Daily observational files created
[5.1] Daily data ingested to Main Database
[5.2] Raw Data Archived in Raw Database
[5.3] Daily data delivered to Subsystems as required.
[6.1] Main Database version delivered to Subsystems
[6.2] Raw Database input to Intermediate Update system
[8] Main Database delivered for Catalogue production
[S1] Possibility to produce observational simulation
[S2] Possibility of TM simulation
[S3] Special Training Set Simulations for CU8
The Gaia data reduction system: HPC in Java
Very early on in the preparation of the Gaia data reduction the issue of the programming language to use to develop the system was raised.

The decision process involved scientists and software engineers; it was focused on the needs of a long-term project, with stringent requirements regarding the software validation and quality and large CPU and data handling needs ($10^{21}$ flops, 1PB).
FORTRAN was somewhat favoured by the scientific community but was quickly discarded; the type of system to develop would have been unmaintainable, and even not feasible in some cases.

For this purpose the choice of an object-oriented approach was deemed advisable. The choice was narrowed to C++ and Java.
The C++ versus Java debate lasted longer.

“Orthodox” thinking stated that C++ should be used for High Performance Computing for performance reasons.

“Heterodox” thinking suggested that the disadvantage of Java in performance was outweighed by faster development and higher code reliability.
However, when JIT Java VMs were released we did some benchmarks to compare C++ vs Java performances (linear algebra, FFTs, etc.).

The results showed that the Java performance had become quite reasonable, even comparable to C++ code (and likely to improve!).

Additionally, Java offered 100% portability and I/O was likely to be the main limiting factor rather than raw computation performance.
Java was finally chosen as the development language for DPAC. Since then hundreds of thousands of code lines have been written for the reduction system.

We are happy with the decision made and haven’t (yet) faced any major drawback due to the choice of language.
A practical example: relativity corrections

A key piece of the Gaia astrometry is the calculation of the relativity effects on the apparent position of the objects in the sky: aberration, light bending, etc.

This is a complex calculation taking into account the ephemeris of the major solar system bodies and requiring, for a $\mu$as accuracy, to reach the limit of the numerical precision of double variables.
An initial (legacy) implementation was available from S. Klioner in C. Used in the simulator code until 2008 through JNI calls.

The same author recently developed for DPAC a new implementation using Java. Both implementations have been thoroughly compared and results agree at sub-μas level.
However, computation times differ substantially …

In Mare Nostrum the Java version runs about four times faster than the C version (and it’s not due to the JNI overhead).

Obviously, an optimisation of the C code should make it much more efficient, to at least the level of the Java code. However, this shows how the same developer did a quicker and better job in Java (a language that, unlike C, he was unfamiliar with).
A working example: the Gaia simulator

The Gaia Simulator code amounts today to more than 100,000 lines of code and has produced several terabytes of simulated data in the last years that have been used for mission design and development and testing of the initial versions of some reduction algorithms.

First fully functional system of DPAC, in production since 2006.
The simulator is run at the **Mare Nostrum supercomputer** using the Grid Superscalar framework.

Latest run:
- 600,000 CPU hours
- 12TB of simulated data
We routinely deploy it with **very small overheads** in several environments: e.g. developers desktop computers, small department cluster for testing and validation and Mare Nostrum for production. Also in all OS flavours: Linux/Unix, Windows, Mac.
Obviously, some caveats

• We have no numerical libraries of the quality and sophistication of those available in C or Fortran (but this seems to be improving)

• Other types of libraries seem also more limited (e.g. MPI libraries)

• Support for Java development in HPC platforms is scarce. Experience and advice also. We are breaking the ice.
• Occasionally, we find some subtle but annoying differences between JVMs (but not often and never critical for the moment).

• We might not be able to take advantage of future advanced processors, like Cell.

• We do not have much control of garbage collection, and sometimes this may be a problem (but not having to worry about memory leaks is a bless).
Java can be used (is being used!) for HPC, but would need more support from the HPC community.