Data Representation and Code Interoperability in Quantum Chemistry:

the Q5COST approach

Stefano Evangelisti

University of Toulouse

Plan of the talk

- Introduction
- Use a XML data format?
- Q5Cost data format
- Q5cost library
- Q5cost coding examples
- Applications
- Future Developments and Conclusions

Introduction

The context

- Activity carried on within Cost in Chemistry D23-D37
- Codes produced by the involved parties are complementary and often need to be interfaced
- Final goal: To build a grid based distributed laboratory
- Facilitate communication between different QC codes
- First problem to face: Each code works with its own data format

Involved Parties and Codes

- CINECA (Italy) coordinator
- University of Bologna (Italy) a FCI package, with calculation of energy an d first and second order properties;
- University of Budapest (Hungary) participation to the COLUMBUS project: a general purpose abinitio chain (SCF, CASSCF,CI); implementation of a direct MRCC algorithm;
- University of Ferrara (Italy) NEVPT a MR perturbative algorithm;
- University of Toulouse (France) CASDI a MR CI algorithm;
- University of Lille (France) EPCISO a spin orbit code;
- University of Valencia (Spain) PROP evaluation of molecular properties;
- ETH of Zurich (Switzerland) GAMESS US a general ab initio packag e and Gemstone a grid architecture environment for QC;
- University of Tromsø (Norway): participation to the DALTON project;

A Common Format for QC codes

- Our decision:
  - To build a Common Format for QC problems [a]
  - To write a converter wrapper for each code in the set
  - Directly add the support to the format inside the original codes

- Common Format should be:
  - as general and complete as possible
  - flexible enough to be interfaced with codes under constant development
  - platform independent
  - easy to use for chemical users

[a] Angeli *et al. Int. Jour. Quant. Chem. v. 107, p. 2082 (2007)*

QC Data

- We can identify two types of QC data:
  - Small data quantities (mainly ASCII coded)
    - Geometry, Symmetry, Atomic basis set, etc...
  - Large datasets (mainly binary)
    - AO or MO integrals, MO coefficients, Wavefunction
- We devised Q5Cost an HDF5 based data format
- Small data can be retrieved from the Q5Cost and coded in XML (Qcml)

Qcml data format

Qcml Data Format

- Xml based format
- Deals with Quantum Chemistry Concepts
  - Basis set
  - Geometry
  - etc...
- Xml Schema and Documentation at the url: http://abigrid.cineca.it
- Planning a better integration with Cml and Cml-Comp

Qcml Data Format

- First section (Base Facts)

```
 <file address URL/>
<molecule nElectrons charge spinMult spaceSymmetry>
<symmetry ... />
<geometry ... />
<basis    ... />
</molecule>
```

Qcml Data Format

- Second Section (Derived Facts)

```
<computedData>
 <energy unit levelOfTheory quality value>
    <state spaceSymmetry spinMultiplicity excLevel />
 </energy>
 <property unit levelOfTheory quality value>
    <state "bra" spaceSymmetry spinMultiplicity excLevel />
    <state "ket" spaceSymmetry spinMultiplicity excLevel />
    <operator order name/>
 </property>
</computedData>
```

- Third Section Workflow (Not Defined yet)

f90/f77 xml library

- At the time we started no Fortran/Xml library was available
- We wrote a f90/f77 xml library
- General library (Works with Xml objects not about QC ones)
- Built on top of gdome2 C/xml library
- We plan to write a high level Fortran library based on Chemical concepts

Q5Cost data format

What is HDF5?

- HDF5 [a] Format and software for scientific data produced by NCSA/University of Illinois
- Supports any kind of data for digital storage regardless of their origin and size
- Stores data in a highly organised and hierarchical format
- High efficient chunked I/O
- High Efficiency compression using zlib
- Allows inclusion of metadata (attributes)
- Platform independent file format
- Widely used in scientific or visualisation codes

[a] HDF5 a general purpose library and file format for storing scientific data. http://hdf.ncsa.uiuc.edu/HDF5/

HDF5 Data Model

- Datasets
  - Multidimensional arrays of elements together with supporting metadata (attributes)
- Groups
  - Directory like structures containing, datasets, attributes, other groups

HDF5 Hierarchy

● Plan of the talk

Introduction

Qcml data format

Q5Cost data format

● What is HDF5?

● HDF5 Data Model

● HDF5 Hierarchy

● Q5Cost file

● Q5Cost file hierarchy

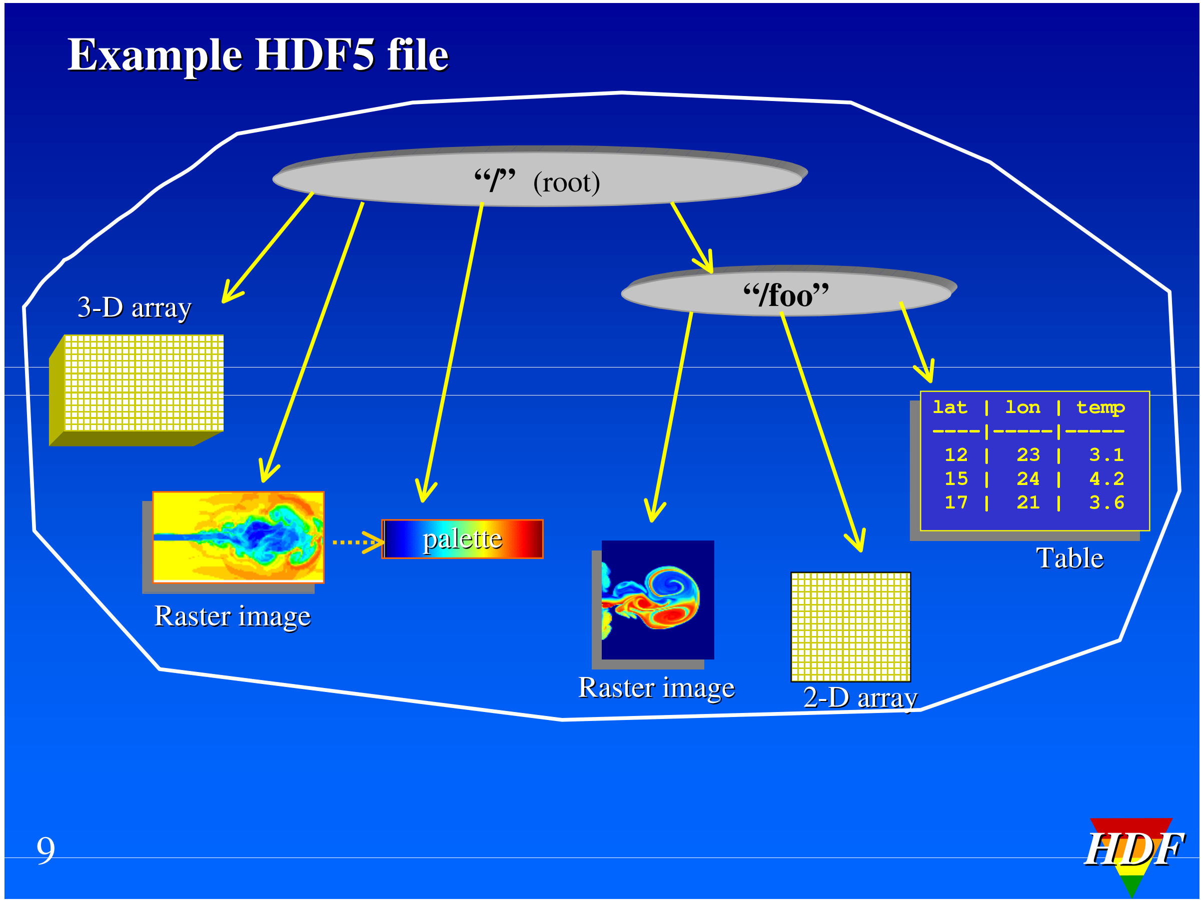
● Q5Cost file Conclusion

Q5cost library

Coding Examples

Applications

Future Developments and Conclusions



Example HDF5 file

"/" (root)

3-D array

"/foo"

Raster image

palette

Raster image

2-D array

```
lat | lon | temp
----|-----|-----
 12 |  23 |  3.1
 15 |  24 |  4.2
 17 |  21 |  3.6
```

Table

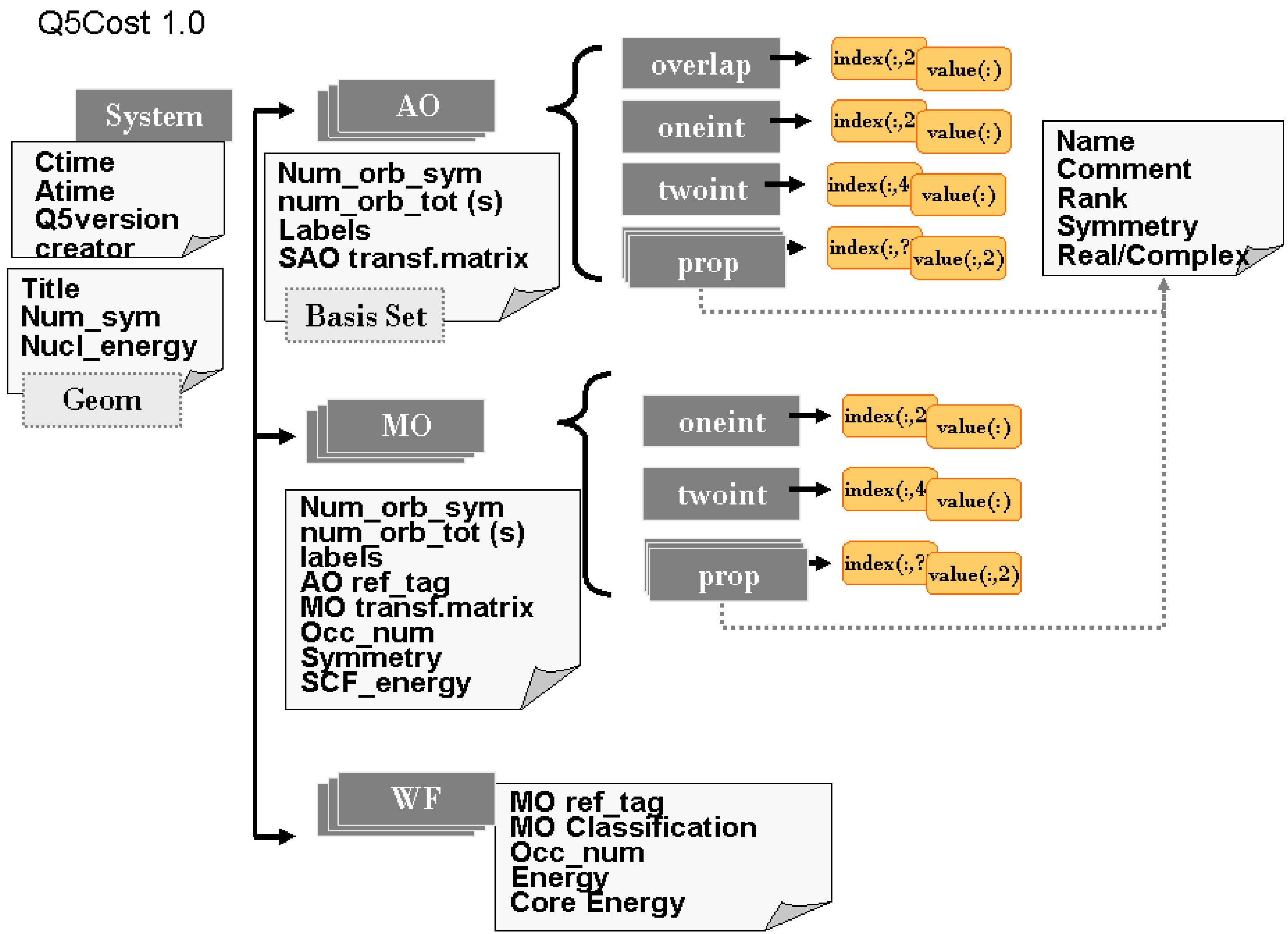
9

HDF

Q5Cost file

- Q5Cost file stores:
  - large sparse matrices with arbitrary number of indeces (AO and MO integrals related to a generic One or Two particles operator), defined as Generic Properties.
  - large matrices to represent CI type Wave-functions
  - small data (scalar and arrays), called metadata (nuclear energy, geometry, orbitals label, MO coefficients, etc..)
- File has a hierarchical structure
  - A first root container (**System**) represents the molecular system
  - A System can contain several Domains, grouping together Properties whose indeces conceptually refers to the same kind of functions
    - **AO Domain**
    - **MO Domain**
    - **WF Domain**

Q5Cost file hierarchy



Q5Cost 1.0

System

Ctime
Atime
Q5version
creator

Title
Num_sym
Nucl_energy

Geom

AO

Num_orb_sym
num_orb_tot (s)
Labels
SAO transf.matrix

Basis Set

overlap → index(:,2 value(:)

oneint → index(:,2 value(:)

twoint → index(:,4 value(:)

prop → index(:,? value(:,2)

Name
Comment
Rank
Symmetry
Real/Complex

MO

Num_orb_sym
num_orb_tot (s)
labels
AO ref_tag
MO transf.matrix
Occ_num
Symmetry
SCF_energy

oneint → index(:,2 value(:)

twoint → index(:,4 value(:)

prop → index(:,? value(:,2)

WF

MO ref_tag
MO Classification
Occ_num
Energy
Core Energy

Q5Cost file Conclusion

- Q5Cost file contains all information one needs to perform a QC computation.
- Q5Cost file stores geometry and symmetry data, and basis set specification.
- Q5Cost file stores Atomic and/or Molecular Integrals and MO Coefficients.
- Q5Cost file stores CI/SCF type Wave-Function determinants and coefficients
- If some information is missing or still not produced it can be added to the file later.
- We can define a proper hierarchy, and store in a simple accessible way metadata.
- Different AO, MO or WF are separate by the use of the identifier tag_ as different objects of a given domain
- Due to HDF5 features Q5Cost files are platform independent.

Q5cost library

Q5cost library

- Basing on HDF5 API we wrote a FORTRAN95 high level library [a]
- Provides read and write access to Q5cost files
- API is based on well known Chemical entities, rather than HDF5 objects
- Provides a high level access for quantum chemistry codes developers

[a] Borini *et al J. Chme Inf. and Model. v. 47, p.1271 (2007)*

Q5cost, where can I found it?

● Plan of the talk

Introduction

Qcml data format

Q5Cost data format

Q5cost library

● Q5cost library

● Q5cost, where can I found it?

● Library structure

● Q5Cost module

● The Q5Cost API

● The Q5cost package

● Performance Test 1

● Performance Test 2

Coding Examples

Applications

Future Developments and Conclusions

- Present version 1.0 by CNRS and CINECA
  - released at ICCSA 2008 conference in Perugia [a]
- The library is free and licensed as LGPL
- Developed in a collaborative environment using CVS
- It can be downloaded from the net: http://abigrid.cineca.it
- It has been tested on various Unix/Linux architecture, and with different Fortran compilers

[a] Scemama et al. Lect. Not. Comp. Sc.

Library structure

● Plan of the talk

Introduction

Qcml data format

Q5Cost data format

Q5cost library

● Q5cost library

● Q5cost, where can I found it?

● Library structure

● Q5Cost module

● The Q5Cost API

● The Q5cost package

● Performance Test 1

● Performance Test 2

Coding Examples

Applications

Future Developments and Conclusions

- The library consists of several modules. The most important ones:
  - **Q5Cost**: Defines the high level API to be used by the final programmer
  - **Q5Core**: provides a wrapping facilities for HDF5 routines
  - **Q5Error**: provides error management. Useful for debugging of library or application codes

Q5Cost module

● Plan of the talk

Introduction

Qcml data format

Q5Cost data format

Q5cost library

● Q5cost library

● Q5cost, where can I found it?

● Library structure

● Q5Cost module

● The Q5Cost API

● The Q5cost package

● Performance Test 1

● Performance Test 2

Coding Examples

Applications

Future Developments and Conclusions

5 Main groups:

- System (the molecular system)
  - molecular geometry, symmetry
  - nuclear repulsion energy, number of α and β electrons
- Basis (the basis set information)
  - Coordinate system (spherical/cartesian)
  - Gaussian contractions (exponents, coefficients,. . .)
- AO (the atomic orbitals information)
  - Symmetry-adatped LCAO on which the MOs are expressed
  - 1- and 2-electron integrals, overlap matrix
- MO (the molecular orbitals information)
  - Orbital energies, occupation numbers, symmetry, ...
  - Classification (frozen, active, virtual, alpha, beta)
  - MO coefficients
- WF (the wave function information)
  - Determinants and coefficients

The Q5Cost API

- A set of fortran routines which encapsualte the HDF5 library calls → The users don't need to know HDF5
- All routine names can be calculated. Example:

```
 Q5Cost_System_get_num_alpha
&    (file_id,num_alpha,error)
```

  - 1) all routine names start with "Q5Cost"
  - 2) the group which contains the data
  - 3) set/get (append/read) the data
  - 4) the name of the data to reach
  - 5) the ID of the file to use
  - 6) the variable in which to put the data (or the variable to write)
  - 7) an error code which is 0 upon success

The Q5cost package

● Plan of the talk

Introduction

Qcml data format

Q5Cost data format

Q5cost library

● Q5cost library
● Q5cost, where can I found it?
● Library structure
● Q5Cost module
● The Q5Cost API
● The Q5cost package
● Performance Test 1
● Performance Test 2

Coding Examples

Applications

Future Developments and Conclusions

- ./configure
- library and include files
- tests
- documentation (file format and API)
- auto-generated F77, C++ and Python bindings
- q5edit (interactive)
- q5dump

Performance Test 1

● Plan of the talk
Introduction
Qcml data format
Q5Cost data format
Q5cost library
● Q5cost library
● Q5cost, where can I found it?
● Library structure
● Q5Cost module
● The Q5Cost API
● The Q5cost package
● Performance Test 1
● Performance Test 2
Coding Examples
Applications
Future Developments and Conclusions

- First test: writing time versus Buffer size for Q5Cost and binary file

Buffer size	Time Binary (s.)	Time Q5Cost (s.)
1024	265.23	226.62
2048	121.13	114.53
4096	62.38	59.02
8192	34.39	31.46
16384	18.86	17.04
32768	8.56	6.09
131072	6.19	4.86
262144	5.84	4.08

Number of integrals: 15000064, binary file size: 343 Mb, Q5Cost file size: 346 Mb

Performance Test 2

● Plan of the talk

Introduction

Qcml data format

Q5Cost data format

Q5cost library
● Q5cost library
● Q5cost, where can I found it?
● Library structure
● Q5Cost module
● The Q5Cost API
● The Q5cost package
● Performance Test 1
○ Performance Test 2

Coding Examples

Applications

Future Developments and Conclusions

- Disk occupation and writing time versus number of integrals for Q5Cost and binary file. (Fixed chunk 16384 integrals)

Integrals	Q5Cost size	Wrt Q5Cost (s)	Binary size	Wrt binary (s)
16384	397 Kb	$5.00 \cdot 10^{-2}$	384 Kb	$5.00 \cdot 10^{-}2$
65536	1.5 Mb	$1.00 \cdot 10^{-1}$	1.5 Mb	$1.00 \cdot 10^{-1}$
114688	2.7 Mb	0.15	2.6 Mb	0.17
507904	12 Mb	0.62	12 Mb	0.68
1015808	23 Mb	1.21	23 Mb	1.37
5013504	115 Mb	5.88	115 Mb	6.41
10010624	231 Mb	11.11	229 Mb	12.12
50003968	1.1 Gb	56.19	1.1 Gb	64.21
100007936	2.3 Gb	125.32	2.2 Gb	148.53

Coding Examples

AO MO create example

● Plan of the talk

Introduction

Qcml data format

Q5Cost data format

Q5cost library

Coding Examples

○ AO MO create example

● MO writing file example

● MO reading file example

Applications

Future Developments and Conclusions

```
PROGRAM aomocreate
      use q5cost
      IMPLICIT NONE
      integer :: error,num_orb_sym(4)=(/8,4,2,0/),num_sym=4
      integer(HID_T) :: file_id !file identification parameter
      character(LEN=10) :: filename="file.h5",
    $    ao_ref_tag="",title="test"
      call Q5Cost_init(error)
      call Q5Cost_file_create(filename,file_id,error)
      call Q5Cost_system_create(file_id,num_sym,title,error)
      call Q5Cost_AO_create(file_id,num_orb_sym,error)
      call Q5Cost_MO_create(file_id,num_orb_sym,error)
      call Q5Cost_file_close(file_id,error)
      call Q5Cost_deinit(error)
END
```

MO writing file example

```
KOUNT_MONO=0
 KOUNT_BI=0
 DO
 DO II = 1,100
    READ(10,*,iostat=error) int,i,j,k,l
    IF (error .lt.  0) EXIT
    IF (k .eq. 0.) EXIT
          kount_bi=kount_bi+1
          value_bi(II)=int
          idx_bi(II,1)=i
          idx_bi(II,2)=j
          idx_bi(II,3)=k
          idx_bi(II,4)=l
 ENDDO
 call Q5Cost_MOTwoInt_append(file_id,idx_bi,$
        value_bi,ii-1,error)
    IF (k .eq. 0) EXIT
 ENDDO
```

MO reading file example

- Plan of the talk

Introduction

Qcml data format

Q5Cost data format

Q5cost library

Coding Examples

- AO MO create example
- MO writing file example
- MO reading file example

Applications

Future Developments and Conclusions

```
offset=0
howmany=chunk
howmany_fixed=chunk
DO
  call Q5Cost_MOOneInt_read(file_id,offset,howmany $
                     ,idx_mono,value_mono,error)
  offset=offset+howmany
  KOUNT_MONO=KOUNT_MONO+howmany
  DO II = 1,howmany
    WRITE(10,’(1X,D20.13,4I4)’) value_mono(II), $
       idx_mono(II,1),idx_mono(II,2),0,0
  ENDDO
  IF (howmany .lt. howmany_fixed) EXIT
  howmany=howmany_fixed
ENDDO
```

● Plan of the talk

Introduction

Qcml data format

Q5Cost data format

Q5cost library

Coding Examples

Applications

● First Application: Interfaces

● First Applications: Quantum Chemistry

Future Developments and Conclusions

Applications

First Application: Interfaces

- Interface from MolCas files to Q5Cost file
  - A project is going on to include Q5Cost in Molcas releases
- Inclusion of Q5Cost into Dalton
- Interface from Columbus files to Q5Cost file
- Interface from Q5Cost file to MolCost files (Toulouse format)
- Bologna FCI code reads data directly from Q5Cost file
- Next targets:
  - Gamess US
  - Molekel via OpenBabel

First Applications: Quantum Chemistry

- A Study on the Dispersion Interaction in Neon dimer performed with Q5Cost format [a]
  - FCI space of 1 billion determinants
  - Toulouse CI code CASDI interfaced to Dalton via Q5Cost
- A Study on the Dispersion Coefficients in BH dimer
  - FCI space of about 300 milion determinants
- FCI study of Mott transition in Li$_n$ chains
  - FCI space up to 1 billion determinants

[a] Monari *et al, Journ. Chem. Theor. Comp.*, ***3****, 477-485, 2007*

Future Developments and Conclusions

Future Developments

● Plan of the talk

Introduction

Qcml data format

Q5Cost data format

Q5cost library

Coding Examples

Applications

Future Developments and Conclusions

○ Future Developments

● Conclusions

● Aknowledgments

- Complete the Qcml definition and integration with Cml
- Write a high-level library
- Write the extractor of "small data" from the Q5Cost
- Introduce routines to obtain the indeces when integrals are stored with a defined order and/or to order integrals given the indeces
- Adding Q5Cost support to more QC codes
- Move towards Grid: Workflow, Web-interfaces, Visualization, etc...

Conclusions

- An efficient binary, platform independent file format for QC data has been presented
- An easy to use Fortran library has been written to access the Q5Cost file format
- Preliminary performance tests show library efficiency regarding disk occupation and writing/reading time
- Applications have been written
- First actual computations have been performed.

Aknowledgments

- The Q5Cost team:
  - E. Rossi (CINECA)
  - S. Borini (Zurich)
  - A. Monari (Bologna)
  - A. Scemama (Toulouse)
  - S. Evangelisti (Toulouse)

- Founded by European Community under the project: Cost in Chemistry D37