Analysis of properties of designs of the ATLAS LVL2 system with Microsoft Excel

Jos Vermeulen
NIKHEF
June 25, 1999
(for version 3.00 of spreadsheet)

Abstract

For judging the basic properties of different designs for the Atlas LVL2 system and DAQ-1 Read Out Crates a “paper model” has been implemented in Microsoft Excel. This note describes aspects of the model and of the spreadsheet implementation for the LVL2 system. The Read Out Crates are dealt with in a separate note.

1. Introduction

For analysis of the influence of decisions on the architecture of the Atlas LVL2 system on requirements on number of messages per second and link bandwidth at various places in the system, on number of processors of different type, etc. “back of the envelope” calculations give a first approximation. However, although the type of calculations is simple, the amount of numbers to deal with makes some form of bookkeeping necessary. For this type of task spreadsheet programs are well suited, as was recognized by the RHUL group.

In this note a “paper model”, implemented in Microsoft Excel, is described. The spreadsheet can be used for studying in first-order approximation the properties of different architectures and of different trigger strategies. Two trigger menus are included, a high-luminosity menu and a low-luminosity menu. It is possible to choose between parallel and sequential trigger strategies. Paper300.xls contains the pilot project LVL2 model for low luminosity, parameters, trigger menus and selection strategies are as documented in the slides of the presentation made in the pilot project modelling meeting of June 2, 1999, see the pilot project modelling web page. For obtaining the high luminosity results change cell D1 on the "Abstract" sheet from "Low" to "High".

The spreadsheet model uses only worksheet functions on 18 worksheets. The worksheet functions all refer to relative cell numbers. Hence input and output of the calculations still come from and go to the correct cells after insertion of new columns or rows. The first worksheet (labeled with “Abstract”) provides a short overview of the results and a number of graphs, on the second (labeled with “Main sheet”) all calculations are done, the third (labeled with “Overview”) provides an overview of the most important parameters and results, while the fourth and fifth sheets contain the low- and high-luminosity trigger menus. At the bottom of the worksheets with the menus graphs with the total RoI request rates for the different types of RoIs and the different subdetectors can be found. The sixth and seventh sheet are used for the model of the DAQ-1 Read Out Crates and are documented in a separate note. The next two sheets contain tables in the same format as used for DAQ note 113 describing the results of paper modelling with the parameters documented in DAQ note 70. The sheet labeled with "RobsPerRoI av." contains a table with the average number of ROBOuts of the different subdetectors receiving a RoI request for em, muon and jet RoIs with entries for 1 - 128 ROBIns per ROBOut and a related table. The sheet labeled with "min-max rates" contains a table with a lower limit of and a table with an upper limit of the RoI request rates per ROBOut. The remaining sheets contain tables and graphs with for each ROB of each subdetector the probability for a hit by a RoI request for the different possible RoI types given a RoI request with arbitrary eta and phi. The RoI request rate for a certain ROB for a certain RoI type can be determined by multiplying the probability with the total RoI request rate for the subdetector and RoI type considered.

When using the “A format” option for referring to cells, by clicking a cell one can see immediately in the formula bar which cells contain input data for the calculation the outcome of which is stored in the cell clicked (A graphical indication by means of arrows pointing to these cells is also possible with the auditing tools of Excel). A number of cells contain input data or select different options. For example it is possible to make a choice between a “push” or a “pull” architecture. Most parameters can be changed on the second worksheet, but the most important parameters need to be changed on the “Abstract” worksheet (indicated with blue cells with a bold red font). Changing input data almost immediately produces new output results. Text placed on the worksheet describes the type of data in each cell.

2. Basic design of the spreadsheet model

2.1 Rates in stead of individual trigger items

A possible approach consists of computing for each individual trigger item quantities of interest like e.g. the input rate to all feature extractor processors. For each quantity the contributions of the various trigger items are then summed. Only for computing latencies this approach is necessary, as the composition of each trigger item, in combination with the properties of the LVL2 system, determines what the average latency will be. However, for computing rates and data volumes only the sums of the rates and the data volumes associated with the different trigger objects do matter (recall that a trigger item is composed of one or more trigger objects). Only the rates depend directly on the trigger menu, the data volumes depend in turn on the rates for different trigger objects. The conclusion is that the analysis can be based on the rates for muon RoIs, e.m./photon RoIs, jet RoIs, TRT scan RoIs and missing energy RoIs. A sequential model can be studied by adapting the rates for the different RoI types and by making the rates for each RoI type different for different detectors, if required. For example in a non-sequential trigger the muon RoI rate may cause an identical RoI rate for the e.m. calorimeter, the hadron calorimeter, the SCT and the TRT. A model in which all muon RoIs are analyzed for the muon detector, but where only a certain fraction of the muon RoIs will be analyzed for the other detectors, can be studied by giving the RoI rates for the other detectors a value equal to that fraction of the RoI rate for the muon detector.

The RoI rates are taken from one of the trigger menu’s. In the case of a sequential model, the rates are reduced according to a table found at the right (columns AA and AB for the low luminosity menu, columns Y and Z for the high luminosity menu) of the worksheets with the trigger menus. This table contains reduction factors.

2.2 Regions of Interest

2.2.1 RoI types

In the spreadsheet model all calculations are done for 4 different RoI types : muon, e.m., jet and “LVL2 RoIs”. Missing energy, the scan for the TRT and the RoIs produced by the scan for the SCT and pixels are “LVL2 RoIs”.

2.2.2 Distribution of RoIs in eta-phi space

It is assumed that all RoIs are independent of each other and that all possible eta and phi values of the RoI center have an equal probability to occur.

2.3 Detector types

In the spreadsheet 7 different subdetectors are used : Pixels, SCT, TRT, em calorimeter, hadron calorimeter, muon precision and muon trigger detector. The e.m.

and hadron calorimeter are handled as different subdetectors. For feature extraction for the calorimeter is done using the data of both calorimeters. Also for feature extraction for the muon detector the data of both the muon precision chambers and the muon trigger detectors is used. Feature extraction for the pixel detector and the SCT is also assumed to be done for both detectors together.

2.4 LVL2 system model

2.4.1 LVL2 system elements

The following LVL2 system elements are implemented in the spreadsheet model :

- a supervisor receiving RoIs either from the LVL1 trigger or from LVL2 global processing and distributing these RoIs either to the ROBs ("push" model) or to the LVL2 processors, responsible for feature extraction ("pull" model). The supervisor also receives final LVL2 decisions from the LVL2 processors responsible for global processing and broadcasts these decisions to the ROBs as blocks containing a number of decisions,
- for the "push model" : distributors of RoI requests to the ROBOuts (see below). The input of the distributors is coming from the supervisor; each subdetector has its own distributor,
- ROBOuts receive RoI requests and pass these to the ROBIn or ROBIns connected to the ROBOut and receive RoI fragments from the ROBIns, merge these in a single fragment per event and transmit these fragments via data links connecting to the LVL2 system. In the "pull" model the RoI requests are received via the same data links; in the "push" model the RoI requests are received from the distributors,
- ROBIns taking care of buffering of data, responding to RoI requests and to LVL2 decisions by sending data to the LVL2 system or, for accepted events, to the LVL3 system, respectively,
- feature extraction : results are calculated for (i) processors individually connected to the networks via which RoI fragments are input and results are output to the global processors, as well as for (ii) processors arranged in subfarms, which are assigned to subdetectors, and for (iii) processors arranged in subfarms that can receive data from all subdetectors and also take care of global processing,
- global processing : results are calculated for (i) processors individually connected to the networks via which feature extraction results are input and decisions are output to the supervisor, as well as for (ii) processors arranged in subfarms, and for (iii) processors arranged in subfarms that also take care of feature extraction.

2.4.2 RoI request distributors and ROBOuts

For the RoI request distributors and for the handling of RoI requests by the ROBOuts a choice can be made between different operational modes :

- routing of incoming messages,
- broadcasting of incoming messages on all outputs,
- intelligent distribution of incoming messages to only those ROBIns that need to receive the message. For this option there is always only one input message per RoI, the distributor or ROBOut can derive request messages for all the ROBIns or ROBOuts in the region of interest that are connected to the ROBOut or distributor respectively.

A detailed knowledge of how RoI requests map onto the ROBIns and the ROBOuts has been obtained with a C++ program, the "ROBsPerRoI program", which is included in the simdaq distribution. This program determines for all possible RoI positions which ROBs and ROBOuts in each subdetector need to send data to the LVL2 decision. Results were obtained for 1 - 128 ROBIns connected to a single ROBOut. The results

were incorporated in tables in the spreadsheet. The sheet labeled with "RobsPerRoI av." contains a table with the average number of ROBOuts of the different subdetectors receiving a RoI request for em, muon and jet RoIs with entries for 1 - 128 ROBIns per ROBOut. The table was calculated with the "ROBsPerRoI program. The same sheet also contains a table with the reduction factors, defined as : (average number of RoI requests per ROBOut) / ((average number of RoI requests per ROBIn) * (number of ROBIns connected to a ROB out)). The sheet labeled with "min-max rates" contains a table with a lower limit of and a table with an upper limit of the RoI request rates per ROBOut, again calculated with the "ROBsPerRoI program". The rates are relative rates and should be multiplied with the total RoI request rate for the RoI type and subdetector considered to determine the absolute rate.

2.4.3 Subfarms of processors

The input and output data rates are equal to the sums of the rates for the individual processors.

2.5 Important input parameters

Important input parameters are indicated with light violet cells, that contain dark blue or red bold text. Light violet cells with "<<<<" in bold red indicate that the value immediately left is copied into the entries left to it.

The most important parameters are found at the top of the first worksheet labeled with "Abstract". They specify :

- the type of luminosity ("High" or "Low"),
- the processing model (specify "Yes" or "No" after "Sequential processing" and "Push" or "Pull" after "Push or Pull"),
- whether there is a missing energy LVL2 trigger ("Yes" or "No"),
- whether traffic to the event builder is taken into account or not can be specified,
- scaling to 100 kHz LVL1 rate ("Yes" or "No"),
- whether model C is used ("Yes" or "No", has consequences for the processing resources needed for the supervisor),
- the parallelism factor of the TRT scan and for the missing energy trigger,
- the communication overhead that is everywhere used (in ms),

On the first worksheet also the number of ROBs connected to a single ROBOut can be set for each subdetector. See also section 2.4.2. All other parameters can be set on the second worksheet (labeled with "main sheet").

Before changing a number in a cell, always verify in the formula bar that the number is not the output of a formula. If it is, do change the cell contents only when you know what you are doing !!

On the "main sheet", cell I42, there is an option to scale (reduce) the number of ROBs with a factor specified in cell I44. The event fragment sizes and the number of ROBs hit per RoI are adjusted proportionally. Note that for this option the results from the program referred to in section 2.4.2 are not used. Hence this option should only be used to obtain a rough idea on the effect of reducing the number of ROBs (by mapping larger parts of the detector to single ROBs).

3. Results produced by the spreadsheet model

Table 3 provides an overview of the quantities calculated. The numbers in the table are also used in the spreadsheet, so that the table can be used to locate results in the spreadsheet.

5. Average number of ROBIns hit per RoI, per type
6. Fraction of ROBIns hit per RoI
7. Total RoI rate per detector (kHz)
8. RoI request rate per ROBIn (kHz)
9. Data volume requests per ROBIn (kByte/s)
10. RoI fragm. volume -> feature extr. (kByte)
11. RoI fragm. data volume / s output per ROBIn (kByte/s)
12. Processing time per ROBIn
14. RoI fragment rate / volume in per ROBOut (kHz / kByte/s)
15. ROBOut occupancy (%)
16. RoI fragment rate out per ROBOut (kHz)
18. RoI req rate in per distributor and per ROBOut (kHz)
19. RoI req data volume in per distributor and per ROBOut (kByte/s)
20. RoI req rate out per distributor and per ROBOut (kHz)
21. RoI req data volume out per distributor and per ROBOut (kByte/s)
22. RoI fragment data volume -> feature extractors (MByte/s)
23. RoI fragment rate -> feature extractors (kHz)
24. Average RoI fragment size (kByte)
25. Supervisor request rate -> feature extractors (kHz)
26. Supervisor data volume -> feature extractors
27. Feature extractors -> RoI request output rate (kHz)
28. FEX -> RoI request output volume (kByte/s)
29. Feature extraction (ms)
29.1 Number of processors
29.2 Numer of processors when SFI is separate
29.3 Minimal number of SFI processors
29.4 Total in per processor (kByte/s)
29.5 Total out per processor (kByte / s)
29.6 Total in per processor (kHz)
29.7 Total out per processor (kHz)
30. Subfarms per detector for parallel FEX
30.1 Number of subfarms
30.2 Total in per subfarm (MByte/s)
30.3 Total out per subfarm (MByte/s)
30.4 Total rate in per subfarm (kHz)
30.5 Total rate out per subfarm (kHz)
30.6 Total processing time per interface (s/s)
31. Model C Only, FEX and global processing combined
31.1 Number of FEX + global processors, SFIs are separate
31.2 Number of SFIs
31.3 Number of FEX-global processors + number of SFIs
31.4 Number of processors without separate SFIs
32. Data volume / s -> global processing (MByte/s)

33. Global processing
33.1 Number of processors
33.2 Message rate in per processor (kHz)
33.3 Data volume in per processor (MByte/s)
33.4 Message rate out per processor (kHz)
33.5 Data volume out per processor (kByte/s)
33.6 Proc. time per event (ms)
34. Supervisor
34.1 Total input rate (kHz)
34.2 Total input data volume NOT from LVL1 (kByte/s)
34.3 Total processing time (s / s)
34.4 Supervisor outppput rate -> distributors (kHz)
34.5 Out data volume -> distributors (kByte/s)
34.6 Supervisor out rate -> FEX / Global (kHz)
34.7 Out data volume -> FEX / Global (kByte/s)
34.8 Decision out data rate -> ROBOuts (kHz)
34.9 Decision out data volume -> ROBOuts (kByte/s)

Table 1. Overview of the quantities calculated.

4. Specification of calculations on the Main sheet

Conventions :

Square brackets [ ] indicate summing or averaging
Curly brakets { } indicate an “If” “else” condition

Numbers refer to the item numbers on the main sheet

```
21. Quantity =
  A (17) +
  If Condition
  {
    C (input) +
  }
  D (14)
```

means : if in item 21. the condition is valid then Quantity = A + C + D, else Quantity = A + D. A is computed in item 17, D is computed in item (14) and C is an input for this item.

Specification of the calculations :

```
1.  Input

2.  LVL1 rate : calculated from trigger menu chosen

3.  RoI rates : calculated from trigger menu chosen

4.  Number of ROBs per detector : input, but scaled with factor in cell I44

5.  Average number of ROBs hit per RoI : input (calculated with ROBsperRoI program,
    but scaled with factor in cell I23

6.  Fraction of ROBs hit per RoI =
      Average number of ROBs hit per RoI (5) / Number of ROBs per detector (4)

7.  Total RoI rate per detector =
      sums over different categories of RoI rates (3)

8.  RoI request rate per ROB, intelligent distribution (8.1) =
      RoI rate (3) * Fraction of ROBs hit per RoI (6)

    RoI request rate per ROB (8.2) =
      RoI request rate out per ROBOut (20.2) / Number of ROBs per ROBOut (13)

9.  Data volume requests per ROB (kByte/s) =
      RoI request rate per ROB (8.2) * request size (input for 9.)

10. RoI fragment volume -> feature extractors =
      (Overhead per ROB (kByte) (input for 10.) +
      ROB data volume (input for 10.) ) *
      Average number of ROBs hit per RoI (5)

11. RoI fragment data volume / s output per ROB =
      Sum over all possible RoIs [ (RoI rate (3) *
      Fraction of ROBs hit per RoI (6) ) *
      (Overhead per ROB (kByte) (input for 10.) +
       ROB data volume (input for 10.) ) ]

12. Processing time per ROB =
      LVL1 rate * overhead per decision +
       Sum over all possible RoIs [RoI request rate per ROB (8) *
       (pre-processing time + overhead for pre-processing +
       overhead per RoI request + overhead for output) ]

13. ROBOut function : inputs taken from abstract sheet

14. RoI fragment rate in per ROBOut =
      Number of ROBs per ROBOut (13) *
      RoI request rate per ROB, intelligent distribution (8.1)
      NOTE : Only ROBs that have RoI data send this to the ROBOut, for the
      calculation we need therefore to use the "RoI request rate per ROB for
      intelligent distribution"

    RoI volume in per ROBOut =
      Number of ROBs per ROBOut (13) *
      RoI fragment data volume / s output per ROB (11)
```

```
15. ROBOut occupancy =
       RoI fragment rate in per ROBOut (14) * overhead per RoI fragment in +
       RoI fragment rate out per ROBOut (16) * overhead per RoI fragment out +
       RoI volume in per ROBOut (14) / Merging speed (input for 15) +
       RoI request rate in per ROBOut (18.2) * overhead per RoI request rate in +
       RoI request rate out per ROBOut (20.2) * overhead per RoI request rate out +
       If Model C
       {
         Decision rate per ROB (cell I48) * (Number of ROBs per ROBOut (13) *
         I/O overhead for output +
         I/O overhead for input)
       }

16. RoI fragment rate out per ROBOut =
       If mixing in ROBOut of RoI fragments (13)
       {
         Reduction factor * Number of ROBIns per ROBOUT (13) * RoI rate (3) *
         Fraction of ROBs hit per RoI (6)
       }
       else
       {
         RoI fragment rate in per ROBOut (14)
       }

17. Number of ROBOuts (that receive RoI requests from distributor) per distributor,
    push only =
       Number of ROBs per detector (4) / Number of ROBs per ROBOut (13)

18. RoI req rate in per distributor =
        If Push model
        {
           If Intelligent distribution (incl. replication) of RoI requests (13)
          {
            Total RoI rate per detector (7)
          }
          else
          {
             If Broadcasting for "dumb" distribution of RoI requests (13)
            {
              Total RoI rate per detector (7)
            }
            else
            {
              Total RoI rate per detector (7) *
              Average number of ROBs hit per RoI (5)
            }
          }
        }
        else
        {
           0
        }

    RoI request rate in per ROBOut (source : distributor (push) or
    FEX / global processor (pull) ) =
         If Intelligent distribution (incl. replication) of RoI requests (13)
        {
           Number of ROBs per ROBOut (13) *
           RoI request rate per ROB, intelligent distribution (8) *
           Reduction factor (16)
        }
        else
        {
             If Broadcasting for "dumb" distribution of RoI requests (13)
            {
              If Push model
              {
                 RoI rate for detector (3)
              }
              else
              {
                 Number of ROBs per ROBOut (13) *
                 RoI request rate per ROB, intelligent distribution (8.1) *
                 Reduction factor (16)
                 (NOTE : it is assumed here that only in the ROBOUT there is
                 broadcasting of RoI requests, so the ROBOUT itself
                 only receives RoI requests that have to go to at least one of the
                 ROBs connected to it)
              }
            }
            else
            {
               Number of ROBs per ROBOut (13) *
               RoI request rate per ROB, intelligent distribution (8)
               (NOTE : in this case the ROBOut only passes the requests to the ROBs
               addressed)
            }
```

```
  }

 Decision rate in per ROBOut =
  If Model C
  {
     Decision rate per ROB (cell I49)
  }
     else
  {
     0
  }

 Total rate in per ROBOut =
  RoI request rate in per ROBOut + Decision rate in per ROBOut
```

19. RoI request data volume in per distributor =

```
    Request size RoIR record (input) * RoI request rate in per distributor (18)

 RoI request data volume in per ROBOut =
    Request size RoIR record (input) * RoI request rate in per ROBOut (18)

 Decision data volume in per ROBOut =
  If Model C
  {
     Decision rate per ROB (cell I49) *
     (4 * number of decisions per block (cell I46) + 8)
  }
  else
  {
     0
   }

  Total data volume in per ROBOut =
    RoI request data volume in per ROBOut + Decision data volume in per ROBOUT
```

20. RoI request rate out per distributor =

```
 If Push model
 {
    If Intelligent distribution (incl. replication) of RoI requests (13)
   {
      Number of ROBs per detector (4) *
      RoI request rate per ROB, intelligent distribution (8) *
      reduction factor (16)
   }
   else
   {
      If Broadcasting for "dumb" distribution of RoI requests (13)
     {
        Total RoI rate per detector (7) *
        Number of ROBs per detector (4) / Number of ROBs per ROBOut (13)
        (NOTE : broadcasting is assumed to be done in 2 phases : by distributor
         to ROBOuts, and next by ROBOuts to ROBs)
     }
     else
     {
        Average number of ROBs hit per RoI * Total RoI rate per detector (7)
     }
   }
 }
 else
 {
   0
 }

RoI request rate out per ROBOut =
  If Intelligent distribution (incl. replication) of RoI requests (13)
 {
    Number of ROBs per ROBOut (13) *
    RoI request rate per ROB, intelligent distribution (8)
 }
 else
 {
    If Broadcasting for "dumb" distribution of RoI requests (13)
   {
      Number of ROBs per ROBOut (13) * RoI request rate in per ROBOut (18.2)
   }
   else
   {
      RoI request rate in per ROBOut (18.2)
      (NOTE : in this case the ROBOut only passes the requests to the ROBs
       addressed)
   }
 }

 Decision rate out per ROBOut =
  If Model C
  {
```

```
      Decision rate per ROB (cell I49) * Number of ROBs per ROBOut (13)
   }
      else
   {
      0
   }

   Total rate out per ROBOut =
     RoI request rate out per ROBOut + Decision rate out per ROBOut
```

21. RoI request data volume out per distributor =
```
   Request size RoIR record (input 19) * RoI req rate out per distributor (20)

 RoI request data volume out per ROBOut =
   Request size RoIR record (input 19) * RoI req rate out per ROBOut (20)

  Decision data volume out per ROBOut =
    If Model C
    {
       Number of ROBs per ROBOut (13) * Decision rate per ROB (cell I48) *
       (4 * number of decisions per block (cell I45) + 8)
    }
    else
    {
      0
    }

    Total data volume in per ROBOut =
      RoI request data volume out per ROBOut +
      Decision data volume out per ROBOut
```

22. RoI fragment data volume -> feature extractors =
```
   Sum over all possible RoIs [RoI rate (3) * Data volume RoI (10) ]
```

23. RoI fragment rate -> feature extractors =
```
   RoI fragment rate out per ROBOut (16) * Number of ROBs per detector (4) /
   Number of ROBs per ROBOut (13)
```

24. Average RoI fragment size =
```
   Sum over all possible RoIs [RoI rate (3) *
   Average number of ROBs hit per RoI (5) *
   Data volume RoI (including overhead) (10) /
   RoI fragment rate -> feature extractors (23) ]
```

25. Supervisor request rate -> feature extractors =
```
   If Pull model
   {
     Total RoI rate per detector (7)
   }
   else
   {
     0
   }
```

26. Supervisor data volume -> feature extractors =
```
   Supervisor message size (input) *
   Supervisor request rate -> feature extractors (25)
```

27. Feature extractors -> RoI fragment request output rate =
```
   If Pull model
   {
      If (Intelligent distribution (incl. replication) of RoI requests (13) OR
           Broadcasting for "dumb" distribution of RoI requests (13) )
     {
       Number of ROBs per detector (4) *
       RoI request rate per ROB, intelligent distribution (8) *
       reduction factor (16)
     }
     else
     {
       Average number of ROBs hit per RoI (5) *
       Total RoI rate per detector (7)
     }
   }
   else
   {
     0
   }
```

28. FEX -> RoI fragment request output volume =
```
   RoI fragment request size (input) *
   Feature extractors -> RoI fragment request output rate (27)
```

29. Feature extraction

```
 Total FEX processing time =
   Sum over all possible RoIs [ FEX time (input) * (1 + overhead factor) +
```

```
  Overhead for receiving RoI fragments +
  Merging time + Overhead per output +
  If Pull mode
  {
     Overhead for sending RoI requests + Overhead per input +
     RoI request formulation time
     (NOTE : 1 message from supervisor contains all LVL1 trigger RoI
     information)
  }
  ) ]

Overhead for receiving RoI fragments =
  Overhead per fragment received  * Average number of ROBs hit per RoI (5) *
  RoI fragment rate out per ROBOut (16) / RoI fragment rate in per ROBOut (14)

Overhead for sending RoI requests =
  Overhead per RoI request  * Average number of ROBs hit per RoI (5) *
  RoI request rate in per ROBOut (18) / RoI request rate out per ROBOut (20)

Merging time =
  ( RoI fragment data volume / s output per ROB (11) *
    Number of ROBs per detector (4) ) /
  ( Total RoI rate per detector (7) * Merging speed (input) )

RoI request formulation time =
  Formulate RoIR time (input) * (1 + overhead fraction) * total RoI rate (7)

Number of FEX processors = Rounded total processing time in seconds / second

"SFI" processing time =
  Sum over all possible RoIs [Total RoI rate per detector (7) *
  (Overhead for receiving RoI fragments + Merging time
  + Overhead per output +
  If Pull mode
  {
    Overhead for sending RoI requests +
    Overhead per input + RoI request formulation time +
  }
  ) ]

 Data volume in per processor =
   ( RoI fragment data volume -> feature extractors (22)  +
   Supervisor data volume -> feature extractors (26) ) /
   Number of FEX processors (29)

 Data volume out per processor =
   ( FEX -> RoI fragment request output volume (28) + Data volume / s ->
   global processing (32) ) /
   Number of FEX processors (29)

 Message rate in per processor =
   ( RoI fragment rate -> feature extractors (23) +
   Supervisor request rate -> feature extractors (25) ) /
   Number of FEX processors (29)

 Message rate out per processor =
   ( Feature extractors -> RoI fragment request output rate (27) +
   Total RoI rate per detector (7) ) /
   Number of FEX processors (29)

 Number of processors when SFI is separate =
   Rounded [total processing time in seconds / second -
   "SFI" processing time in seconds / second +
   If SFI->FEX overhead
   {
     Overhead for receiving RoI data by FEX-global processor
   } ]

 Minimal number of SFI processors =
   Rounded ["SFI" processing time in seconds / second +
   If SFI->FEX overhead
   {
     Overhead for sending RoI data to FEX-global processor
   }
```

30. Subfarms per detector for parallel FEX

```
 Number of subfarms =
   Rounded (Number of FEX processors (29) /
   Number of processors per subfarm (input) )

 Total data volume in per subfarm =
   Number of processors per subfarm (input) * Data volume in per processor (29)

 Total data volume out per subfarm =
   Number of processors per subfarm (input) *
   Data volume out per processor (29)
```

```
Total rate in per subfarm =
  Number of processors per subfarm (input) *
  Message rate in per processor (29)

Total rate out per subfarm =
  Number of processors per subfarm (input) *
  Message rate out per processor (29)

Total processing time per interface (only I/O overhead taken into account) =
  ( Total rate in per subfarm + Total rate out per subfarm ) *
  Processing overhead per message (29)
```

31. Model C Only, FEX and global processing combined

```
If Model C
{
  Total number of SFI processors =
    Sum over all possible RoIs and over all detectors
    [ Total RoI rate per detector (7) *
      (Overhead for receiving RoI fragments (29) + Merging time (29) +
      If SFI->FEX overhead
      {
        Overhead per output (for sending RoI data to FEX step) (29) +
      }
      Overhead for sending RoI requests (29) + Overhead per input (29) +
      RoI request formulation time) (29)
    ] +
    If SFI->FEX overhead
    {
      LVL1 rate * Overhead per input of T2DR message (input) +
    }
    LVL1 rate * Overhead per output of TD2R message (input)

  NOTE : the assumption here is that the global processing step receives a
  RoIRSF message for an event and then sends per subdetector a message for
  each RoI to be analysed to the SFI. The SFI next formulates the individual
  RoI requests and sends these to the ROBs. The global processing step
  produces also decisions which are passed through the SFI via the network to
  the supervisor.

  Total FEX processing requirement (without SFI function) (s/s) =
    Total processing time in seconds / second (29) -
    “SFI” processing time in seconds / second (29) -
    I/O overhead for output to global in seconds / second (29) +
    If SFI->FEX overhead
    {
      Overhead for receiving RoI data by FEX-global processor (29)
    }

  Total global processing requirement (without SFI function) (s/s) =
    Total global algorithm processing time in seconds / second (33) +
    If SFI->FEX overhead
    {
      LVL1 rate (2) *
      Overhead for receiving RoIRSF message by FEX-global processor (input) +
      Sum over all dettectors [Total RoI rate per detector (7) *
      Overhead for sending RoI request messages to SFI (input) ] +
      LVL1 rate (2) * Overhead for sending T2DR messages to SFI (input)
    }
}
```

32. Data volume / s -> global processing =
Average for single feature (input) * Total RoI rate per detector (7)

33. Global processing

```
Number of processors =
  Rounded (For one second :
  Time needed for global step +
  Time needed for RoI formulation +
  Overhead for sending decisions)

  Time needed for global step =
        LVL1 rate * (time for event topo decision)

  Time needed for RoI formulation =
        LVL1 rate * Average number of features * (
        scheduling overhead + RoI formulation process time)

  Overhead for sending decisions =
        LVL1 rate * (I/O overhead + average number of steps - 1) *
        If Model C
        {
          0
        }
```

```
          else
          {
            I/O overhead
          } )
```

34. Supervisor

```
    Total rate in =
      If Model C
      {
        2 * LVL1 rate (2)
        (NOTE : LVL1 trigger out + LVL2 decisions from global processors)
      }
      else
      {
        2 * LVL1 rate (2) +
        sequential RoI rates (rate at which new RoI requests are sent to
        supervisor)
      }

   Total input data volume NOT from LVL1 =
      If Model C
      {
        Decision data for single event from global processing *
        LVL1 rate (2)
      }
      else
      {
        Decision / RoI request data for single event from global processing *
        (LVL1 rate (2) + sequential RoI rates - TRT scan rate) +
        TRT scan rate * RoI request data for single event for TRT scan
      }

   Total processing time =
     LVL1 rate (2) *
     ( I/O overhead  for receiving data from LVL1 system (input) +
       Time for allocating LVL2 processors and routing ROIs +
       If Not Model C
       {
          Time for formulating RoIRs +
       }
       I/O overhead for receiving decisions +
       Time for processing LVL2 decisions +
       Time per event for preparing decision list +
       Time for monitoring LVL2 resources
     ) +
     RoI request rate from global processing *
     ( Processing time per sequential ROI request +
       I/O overhead for sequential ROI request
     ) +
     Total Supervisor Output rate * I/O overhead for output

     (NOTE 1 : RoI request rate from global processing = 0 for model C)

     (NOTE 2 : Processing time per sequential ROI request =
      Time for formulating RoIRs +
      Time for processing message itself
      (DAQ note 70, page 5, 4.1, item 3) )

    Total supervisor Output rate =
      If Model C
      {
        Total number of ROBOuts *
        LVL1 rate / Number of decisions per decision block (cell I91) +
        LVL1 rate                   // ROIRSFs are sent at LVL1 rate for Model C
      }
      else
      {
        If Distribution in Supervisor
        {
           RoI request rate summed over all distributor outputs (20.1) +
        }
        else
        {
           RoI request rate summed over all distributor inputs (18.1) +
        }
        LVL1 rate (2) / Number of decisions per decision block (cell I91) +
        RoI request rate to FEX/Global processors
      }

     (NOTE : RoI request rate at distributors = 0 for Pull mode)

    Supervisor Output data volume =
      If Model C
      {
```

```
   Total number of ROBOuts * LVL1 rate (2) *
   ( 8 + 4 * Number of decisions per decision block (cell I46) ) /
   Number of decisions per decision block (cell I46) +
   ROIRSF size * LVL1 rate
}
else
{
   If Distribution in Supervisor
  {
     RoI request rate summed over all distributor outputs (20.1) *
     Size of RoIR record +
  }
  else
  {
     RoI request rate summed over all distributor inputs (18.1) *
     Size of RoIR record +
  }
  LVL1 rate (2) *
   ( 8 + 4 * Number of decisions per decision block (cell I46) ) /
   Number of decisions per decision block (cell I46)
}
```