A Model for Capacity Planning and Performance Estimations of Data Storage Array

D.Gueorguiev, 9/24/2013

Abbreviations

Components
Memory – (M)
CPU – (C)
Channel – (P)
Board – (B)
Fabric – (F)
Bus – (S)

Elementary operations
Direct Storage Accesses – (DSA)
Direct Memory Accesses – (DMA)
CPU Accesses – (CPU)
Hardware locks – (HL)
Transfers Throughput – (TR)
Memory Transfers Throughput – (MTR)

Workload
Related to Standard devices (non-RDF, non-BCV, non-TDEV) – (STD)
Related to VDEV devices – (VD)
Related to TDEV devices – (TD)
Related to RDF R1 devices – (R1)
Related to RDF R2 devices – (R2)

IO Types
Read hit – (RH)
Read miss – (RM)
Sequential Read – (SR)
Write Hit – (WH)
Write Miss – (WM)
Sequential Write – (SW)
Copy Read – (CR)
Copy Write – (CW)
RDF Copy Disk R1 – (CD)
RDF Copy WP R1 – (CWP)

What is given:
Device workload in terms of io rates and sizes for the following io types:
Standard devices (Non-RDF, Non-VDEV, Non-TDEV)
read hit, read miss, seq read miss, write hit, write miss, seq write, copy read, copy write
BCV VDEV devices
VDEV read hit, VDEV read miss, VDEV seq read miss, VDEV write hit, VDEV write miss, VDEV seq write,
VP TDEV devices
TDEV read hit, TDEV read miss, TDEV seq read miss, TDEV write hit, TDEV write miss, TDEV seq write,
SRDF devices (R1 and R2)
RDF write hit J0 R1, RDF write miss J0 R1, RDF seq write J0 R1, RDF write hit R2, RDF write miss R2, RDF copy disk R1, RDF copy WP R1

Steps to calculate component utilizations:

1. Obtain component internal workload
   - 1.1 Device workload processing:
     a. Push device workload to disks
     b. Push device workload to FE channels
        Options:
        - Use FE channel BTP stats to determine the device load going to each FE channel
        - Distribute the device load evenly among the fe channels
     c. Push device workload to SRDF groups
     d. Push device workload to storage groups
   - 1.2 Disk workload processing
     a. Push disk workload to BE channels
   - 1.3 FE CPUs workload processing
     a. Calculate FE Channel utilizations
     b. Collect internal workload from FE channels
     c. Calculate internal workload due to polling
   - 1.4 BE CPUs workload processing
     a. Calculate BE Channel utilizations
     b. Collect internal workload from BE channels
     c. Calculate internal workload due to polling
   - 1.5 RDF CPUs workload processing
     a. Calculate BE Channel utilizations
     b. Collect internal workload from BE channels
     c. Calculate internal workload due to polling
   - 1.6 Board workload processing
     Collect internal workload from relevant CPUs – this is the internal workload due to IO activity and due to polling
   - 1.7 Fabric workload processing / for VMAX-based arrays /
     Collect internal workload from boards – this is the internal workload due to IO activity and due to polling
   - 1.8 Bus workload processing /this applies only for pre-VMAX arrays/
     Collect internal workload from relevant boards – this is the internal workload due to IO activity and due to polling
   - 1.9 Memory board workload processing /this applies only for pre-VMAX arrays/
     Collect internal workload from relevant buses - this is the internal workload due to IO activity and due to polling

2. internal workload based on elementary operations
   Let us consider a workload represented by the following IO quantities :

   Standard io rates IO and sizes L
   read hit IO_{RH}, L_{RH} , read miss IO_{RM}, L_{RM} , sequential read IO_{SR}, L_{SR} , write hit IO_{WH}, L_{WH} , write miss IO_{WM}, L_{WM} , sequential write IO_{SW}, L_{SW} , copy read IO_{CR}, L_{CR} , copy write IO_{CW}, L_{CW}

   VDEV io rates IO and sizes L
   VDEV read hit $IO_{RH,VD}, L_{RH,VD}$, VDEV read miss $IO_{RM,VD}, L_{RM,VD}$, VDEV sequential read $IO_{SR,VD}, L_{SR,VD}$, VDEV write hit $IO_{WH,VD}$, VDEV write miss $IO_{WM,VD}$, VDEV sequential write $IO_{SW,VD}$,

TDEV io rates IO and sizes L

TDEV read hit $IO_{RH,TD}$, TDEV read miss $IO_{RM,TD}$, TDEV sequential read $IO_{SR,TD}$, TDEV write hit $IO_{WH,TD}$, TDEV write miss $IO_{WM,TD}$, TDEV sequential write $IO_{SW,TD}$,

RDF io rates IO and sizes L

RDF write hit J0 R1 $IO_{WH,R1}$, RDF write miss J0 R1 $IO_{WM,R1}$, RDF seq write J0 R1 $IO_{SW,R1}$, RDF write miss R2 $IO_{WM,R2}$, RDF copy disk R1 $IO_{CD,R1}$, RDF copy WP R1 $IO_{CWP,R1}$

Let us start our discussion with the conversion of the workload to Direct Storage Accesses – the expressions which are obtained are similar to those used for obtaining Direct Memory Accesses, CPU Accesses and Hardware Locks. Depending on the Symmetrix model the following quantities, one for each io type listed above, are measured and are Symmetrix model dependent:

N_{DSA}^{RH} – number of DSA ops due to 1 read hit

⋮

$N_{DSA}^{CWP,R1}$ – number of DSA ops due to RDF copy WP R1

In total there are 27 IO types in the workload so there will be 27 N_{DSA} quantities. Complete list of the quantities denoting the number of DSAs, DMAs, CPU Accesses and HL per io type and all Symmetrix models is shown in the Appendix.

Then the total rate of DSA operations for a workload with the given io rates is obtained as:

$$N_{DSA} = N_{DSA}^{RH} \times IO_{RH} + \cdots + N_{DSA}^{CWP,R1} \times IO_{CWP,R1} \qquad (2.1)$$

where there are 27 terms , one for each IO type listed above.

Similar expressions hold for the DMAs, CPU Accesses and Hardware Locks.

For Transfers Throughput we have the following equation:

$$P_{TR} = L_{RM} \times F_{TR}^{RM} \times IO_{RM} + \cdots + L_{CWP,R1} \times F_{TR}^{CWP,R1} \times IO_{CWP,R1} \qquad (2.2)$$

where the same number of terms , each corresponding to an IO type, as in the expression above contribute to the total transfer throughput. Here L is the IO size and F_{TR} is the Transfer Factor which is io type, channel emulation type and model-dependent. For example the Transfer Factor for some of the io types with Fibre emulation in VMAX-based models is shown below

- Read hits: $F_{TR}^{RH} = m \times N_{node} + n$ where N_{node} is the number of system nodes , $n = 1.404,\ m = 0.145$
- Write hits: $F_{TR}^{WH} = m \times N_{node} + n$ where N_{node} is the number of system nodes , $n = 2.458,\ m = 0.242$
- Read Miss: $F_{TR}^{RM} = 1$
- Write Miss: $F_{TR}^{WM} = 2$
- Copy Read: $F_{TR}^{CR} = 2$
- Copy Write: $F_{TR}^{CW} = 1$

Complete list of the transfer factors for all io types and all Symmetrix models is given in the Appendix.
For sequential reads and writes the Transfer Factor is determined in the following way: those IOs which are smaller than the system track size are treated as sequential hits and the corresponding hit Transfer Factor is used. The ios which are larger than the system track size are treated as sequential miss and the corresponding miss Transfer Factor is used. For all RDF write types the Transfer is set to 1.

Similar equation holds for Memory Transfer Throughputs:

$$P_{MTR} = L_{RM} \times F_{MTR}^{RM} \times IO_{RM} + \cdots + L_{CWP,R1} \times F_{MTR}^{CWP,R1} \times IO_{CWP,R1} \qquad (2.3)$$

where the summation is performed over 27 terms, one for each io type, contributing to the total memory transfers throughput.

Then the internal workload is fully characterized by the following 6 quantities: N_{DSA} , N_{DMA} , N_{HL} , N_{CPU} , P_{TR}, and P_{MTR} . Note that the number of CPU accesses does not impact the internal workload utilization and is used only in order to estimate the impact of the CPU polling and decide if convergence in the iterative utilization computation has been reached (see 9.).

The internal workload of a component can be converted into component utilization contribution. Here we will discuss how the Direct Storage Accesses (DSA) and Transfer Throughput (TR) are converted noting that Direct Memory Accesses (DMA) and Hardware Locks (HL) have the same equations as DSAs while Memory Transfer Throughput (MTR) is similar to TRs. For the utilization due to DSAs the following relationship holds:

$$U_{DSA} = \frac{N_{DSA} \times T_{DSA}}{10^9} \qquad (2.4)$$

where U_{DSA} is the utilization due to DSAs . Here the quantity T_{DSA} is the time in nanoseconds the component is busy due to 1 DSA operation and is calculated as

$$T_{DSA} = a_{DSA} + b_{DSA} \times R(U) \qquad (2.5)$$

where the coefficients a_{DSA} and b_{DSA} are measured in nanoseconds; they depend on the Symmetrix model and on the component to which the internal workload belong to . The function $R(U)$ is the *utilization impact function* defined as

$$\mathrm{R(U)} = \begin{cases} \frac{\mathrm{U}}{1-\mathrm{U}} \text{ iff } \mathrm{U} \neq 1 \\ 0 \text{ iff } \mathrm{U} = 1 \end{cases} \qquad (2.6)$$

The argument of the utilization impact function depends on the component to which the internal workload belongs to. For example for CPU internal workload of VMAX-based model, $a_{DSA} = 2250\ ns$, $b_{DSA} = 200\ ns$.
For DMAs and HLs the utilization is calculated using similar to (2.5) and (2.6) expressions.
Note: CPU accesses do not contribute to the internal workload utilizations.
For Transfer and Memory Transfer Throughputs the expressions are

$$U_{TR} = \frac{P_{TR} \times T_{TR}}{10^9},\ U_{MTR} = \frac{P_{MTR} \times T_{MTR}}{10^9} \qquad (2.7)$$

where P_{TR} is the transfer throughput given with (2.2) and T_{TR} is the time the component is busy due to one transfer operation. The measurement unit of T_{TR} is $\frac{nanosec}{MB}$. Similarly to (2.5) The expression for T_{TR} and T_{MTR} are

$$T_{TR} = a_{TR} + b_{TR} \times R(U),\ T_{MTR} = a_{MTR} + b_{MTR} \times R(U) \qquad (2.8)$$

where $R(U)$ is the utilization impact function defined in (2.6). Note that the measurement unit for a_{TR} , b_{TR} , a_{MTR} , and b_{MTR} is nanoseconds per MB.

3. Account for polling

3.1. CPU polling

The CPU polling is accounted by adding an additional internal workload to each CPU (both front-end and back-end).

$$N_{DSA} = P_{DSA} \times C_u\ ;\ N_{DMA} = P_{DMA} \times C_u\ ;\ N_{HL} = P_{HL} \times C_u \qquad (3.1)$$

where P_{DSA} , P_{DMA} and P_{HL} are the CPU polling coefficient for DSAs , DMAs and HLs respectively. Those coefficients depend on the CPU type and the Symmetrix model. Detailed data for them is provided in the Appendix.
The coefficient C_u is the utilization coefficient given with the equation

$$C_u = 1 - max\left(U^C, U^{B(C)}\right) \qquad (3.2)$$

where U^C is the current CPU utilization without polling contribution and $U^{B(C)}$ is the board utilization for the board to which the CPU belongs.

3.2 Board, Bus and Fabric Polling

From CPUs the internal workload due to polling is aggregated in the boards and for VMAX-based arrays from the boards it is aggregated into Fabric polling workload. For DMX-based systems the the CPU polling workload is propagated to the relevant buses following the Bus-CPU connectivity graph.
The additional internal workloads due to polling are added to the total CPU, board and fabric workloads and participate in the CPU, board and fabric utilization calculation.

4. Channel Utilizations

The channel utilization is simply the ratio of two numbers – the total channel workload throughput divided by the max throughput for this channel type. Hence

$$U^P = \frac{T_{TR}^P}{T_{max}^P} \qquad (4.1)$$

where T_{max}^P is the maximum throughput which channel *P* can sustain. The maximum throughput depends on the channel technology and is listed in the Appendix.

5. Fabric Busy Times and Utilizations

Fabric utilizations are calculated after the entire internal workload from boards has been aggregated – the two components of the internal fabric workload are the workload due to device ios and the workload due to cpu polling.

5.1. Busy Times

The busy times due to elementary operations are computed following the discussion on the conversion of internal to utilization in 2.
The utilization component in the utilization impact function is defined as the lesser of the sum of all board utilizations and 0.75. The equations for the busy times due to elementary ops have the same form as (2.4) and (2.7).

$$T_{DSA}^F = a_{DSA}^F + b_{DSA}^F \times R\big(\min\left(\textstyle\sum_B U^B, 0.75\right)\big) \qquad (5.1)$$

$$T_{DMA}^F = a_{DMA}^F + b_{DMA}^F \times R\big(\min\left(\textstyle\sum_B U^B, 0.75\right)\big) \qquad (5.2)$$

$$T_{HL}^F = a_{HL}^F + b_{HL}^F \times R\big(\min\left(\textstyle\sum_B U^B, 0.75\right)\big) \qquad (5.3)$$

$$T_{TR}^F = a_{TR}^F + b_{TR}^F \times R\big(\min\left(\textstyle\sum_B U^B, 0.75\right)\big) \qquad (5.4)$$

$$T_{MTR}^F = a_{MTR}^F + b_{MTR}^F \times R\big(\min\left(\textstyle\sum_B U^B, 0.75\right)\big) \qquad (5.5)$$

Note that the Fabric in VMAX is modeled as a bus connected to all boards. Therefore the time coefficients a^F and b^F for the Fabric are obtained by using the bus coefficients divided by an appropriate factor:

$a_{DSA}^F = \frac{a_{DSA}^S}{2\times {N_{brd}}^2}$, $b_{DSA}^F = \frac{b_{DSA}^S}{2\times {N_{brd}}^2}$
$a_{DMA}^F = \frac{a_{DMA}^S}{2\times {N_{brd}}^2}$, $b_{DMA}^F = \frac{b_{DMA}^S}{2\times {N_{brd}}^2}$
$a_{HL}^F = \frac{a_{HL}^S}{2\times {N_{brd}}^2}$, $b_{HL}^F = \frac{b_{HL}^S}{2\times {N_{brd}}^2}$
$a_{TR}^F = \frac{a_{TR}^S}{2\times {N_{brd}}^2}$, $b_{TR}^F = \frac{b_{TR}^S}{2\times {N_{brd}}^2}$
$a_{MTR}^F = \frac{a_{MTR}^S}{2\times {N_{brd}}^2}$, $b_{MTR}^F = \frac{b_{MTR}^S}{2\times {N_{brd}}^2}$

5.2. Fabric Utilization

Following the discussion in 2. the fabric utilization is given with equations similar to (2.4) and (2.7)

$$U^F = U_{DSA}^F + U_{DMA}^F + U_{HL}^F + U_{TR}^F + U_{MTR}^F \qquad (5.6)$$

where the utilization components due to DSAs, DMAs, HL and transfers are given as:

$$U^F_{DSA} = \frac{N^F_{DSA} \times T^F_{DSA}}{10^9},\ U^F_{DMA} = \frac{N^F_{DMA} \times T^F_{DMA}}{10^9},\ U^F_{HL} = \frac{N^F_{HL} \times T^F_{HL}}{10^9},\ U^F_{TR} = \frac{P^F_{TR} \times T^F_{TR}}{10^9},\ U^F_{MTR} = \frac{P^F_{MTR} \times T^F_{MTR}}{10^9} \quad (5.7)$$

6. Board Busy Times and Utilizations

Board utilizations are calculated after the entire internal workload from relevant CPUs has been aggregated – the two components of the internal fabric workload are the workload due to device ios and the workload due to cpu polling.

6.1. Busy Times

The board's busy times due to elementary operations are computed following the discussion on the conversion of internal to utilization in 2.

The utilization component in the utilization impact function is defined as the fabric utilization for VMAX-based systems. Then the equations for the busy times due to elementary ops for VMAX-based systems become:

$$T^B_{DSA} = a^B_{DSA} + b^B_{DSA} \times R(U^F) \quad (6.1)$$
$$T^B_{DMA} = a^B_{DMA} + b^B_{DMA} \times R(U^F) \quad (6.2)$$
$$T^B_{HL} = a^B_{HL} + b^B_{HL} \times R(U^F) \quad (6.3)$$
$$T^B_{TR} = a^B_{TR} + b^B_{TR} \times R(U^F) \quad (6.4)$$
$$T^B_{MTR} = a^B_{MTR} + b^B_{MTR} \times R(U^F) \quad (6.5)$$

For DMX-based systems the utilization impact function is defined in terms of bus utilization and the busy times are given as :

$$T^S_{DSA} = a^S_{DSA} + b^S_{DSA} \times R(U^S) \quad (6.6)$$
$$T^S_{DMA} = a^S_{DMA} + b^S_{DMA} \times R(U^S) \quad (6.7)$$
$$T^S_{HL} = a^S_{HL} + b^S_{HL} \times R(U^S) \quad (6.8)$$
$$T^S_{TR} = a^S_{TR} + b^S_{TR} \times R(U^S) \quad (6.9)$$
$$T^S_{MTR} = a^S_{MTR} + b^S_{MTR} \times R(U^S) \quad (6.10)$$

6.2. Board Utilization

Following the discussion in 2. The board utilization is given with equations similar to (2.4) and (2.7)

$$U^B = U^B_{DSA} + U^B_{DMA} + U^B_{HL} + U^B_{TR} + U^B_{MTR} \quad (6.11)$$

where the utilization components due to DSAs, DMAs, HL and transfers are given as:

$$U^B_{DSA} = \frac{N^B_{DSA} \times T^B_{DSA}}{10^9},\ U^B_{DMA} = \frac{N^B_{DMA} \times T^B_{DMA}}{10^9},\ U^B_{HL} = \frac{N^B_{HL} \times T^B_{HL}}{10^9},\ U^B_{TR} = \frac{P^B_{TR} \times T^B_{TR}}{10^9},\ U^B_{MTR} = \frac{P^B_{MTR} \times T^B_{MTR}}{10^9} \quad (6.12)$$

7. CPU Busy Times and Utilizations

7.1 CPU busy times

T^C_{DSA} - the time in which CPU is busy due to 1 DSA operation

T^C_{DMA} - the time in which CPU is busy due to 1 DMA operation

T^C_{HL} - the time in which CPU is busy due to 1 Hardware Lock operation

T^C_{TR} - the time in which CPU is busy due to 1 Transfer operation

T^C_{MTR} - the time in which CPU is busy due to 1 Memory Transfer operation

For models before VMAX:

$$T_{DSA}^{C} = a_{DSA}^{C} + b_{DSA}^{C} \times R\left(\max\left(\min\left(\overline{U}^{S(C)}, \frac{8}{9}\right), \max\left(U^{B(C)}, 0.75\right)\right)\right) \quad (7.1)$$

$$T_{DMA}^{C} = a_{DMA}^{C} + b_{DMA}^{C} \times R\left(\max\left(\min\left(\overline{U}^{S(C)}, \frac{8}{9}\right), \min\left(U^{B(C)}, 0.75\right)\right)\right) \quad (7.2)$$

$$T_{HL}^{C} = a_{HL}^{C} + b_{HL}^{C} \times R\left(\max\left(\min\left(\overline{U}^{S(C)}, \frac{8}{9}\right), min\left(U^{B(C)}, 0.75\right)\right)\right) \quad (7.3)$$

$$T_{TR}^{C} = a_{TR}^{C} + b_{TR}^{C} \times R\left(\max\left(\min\left(\overline{U}^{S(C)}, \frac{8}{9}\right), min\left(U^{B(C)}, 0.75\right)\right)\right) \quad (7.4)$$

$$T_{MTR}^{C} = a_{MTR}^{C} + b_{MTR}^{C} \times R\left(\max\left(\min\left(\overline{U}^{S(C)}, \frac{8}{9}\right), min\left(U^{B(C)}, 0.75\right)\right)\right) \quad (7.5)$$

where $R(U)$ is the utilization impact function defined in the internal workload discussion. The time coefficients a^C and b^C were mentioned in the discussion of converting the internal workload to component utilization . They are given in nanoseconds and a complete list for all components and io type can be found in the Appendix.

With $S(C)$ it is denoted the set of all buses connected to CPU C and with $B(C)$ it is denoted the board to which CPU C belongs. Then the quantity $\overline{U}^{S(C)}$ represents the average utilization of all buses connected to CPU C and $U^{B(C)}$ is the board utilization to which CPU C belongs.

For VMAX-based models $R(U) = 0$. Then eq. (7.1)-(7.5) become

$$T_{DSA}^{C} = a_{DSA}^{C} \quad (7.6)$$
$$T_{DMA}^{C} = a_{DMA}^{C} \quad (7.7)$$
$$T_{HL}^{C} = a_{HL}^{C} \quad (7.8)$$
$$T_{TR}^{C} = a_{TR}^{C} \quad (7.9)$$
$$T_{MTR}^{C} = a_{MTR}^{C} \quad (7.10)$$

7.2 CPU utilization due to internal workload

The CPU utilization by converting the CPU internal workload to utilization discussed in 2. is given with equations similar to (2.4) and (2.7)

$$U^{C} = U_{DSA}^{C} + U_{DMA}^{C} + U_{HL}^{C} + U_{TR}^{C} + U_{MTR}^{C} + \frac{N_{CPU}^{C}}{10^9} \quad (7.11)$$

where the utilization components due to DSAs, DMAs, HL and transfers are given as:

$$U_{DSA}^{C} = \frac{N_{DSA}^{C} \times T_{DSA}^{C}}{10^9},\ U_{DMA}^{C} = \frac{N_{DMA}^{C} \times T_{DMA}^{C}}{10^9},\ U_{HL}^{C} = \frac{N_{HL}^{C} \times T_{HL}^{C}}{10^9},\ U_{TR}^{C} = \frac{P_{TR}^{C} \times T_{TR}^{C}}{10^9},\ U_{MTR}^{C} = \frac{P_{MTR}^{C} \times T_{MTR}^{C}}{10^9} \quad (7.12)$$

8. Components Utilization Calculation

8.1. Procedure For VMAX-based systems

The steps to calculate the component utilizations for VMAX-based system are as follows:

Components Utilization Calculation VMAX

Calculate Fabric Utilization U^F

Set IterCount = 1

Set ExtraUtilOld = 0
Set ExtraUtilNew = 0
Set ExtraUtilDiff = 0
Set StopDueToAccuracy = False
Iterate while (IterCount < IterMax And StopDueToAccuracy == False)
{
Calculate Board Utilizations U^B
Calculate average boards utilization $\overline{U}^B = \frac{\sum_B U^B}{N_{brd}}$
Set ExtraUtilNew = $\overline{U}^B \times 0.12$
Set ExtraUtilDiff = Abs (ExtraUtilNew – ExtraUtilOld)
If ExtraUtilDiff < 0.001
Set StopDueToAccuracy = True
Set ExtraUtilOld = ExtraUtilNew
$U^F = U^F +$ ***ExtraUtilNew***
}
Calculate CPU Utilizations U^C

8.2. Procedure For DMX-based systems

The steps to calculate the component utilizations for DMX-based system are essentially the same but fabric is replaced by buses

Components Utilization Calculation DMX

Calculate Fabric Utilization U^S
Set IterCount = 1
Set ExtraUtilOld = 0
Set ExtraUtilNew = 0
Set ExtraUtilDiff = 0
Set StopDueToAccuracy = False
Iterate while (IterCount < IterMax And StopDueToAccuracy == False)
{
Calculate Board Utilizations U^B
Calculate average boards utilization $\overline{U}^B = \frac{\sum_B U^B}{N_{brd}}$
Set ExtraUtilNew = $\overline{U}^B \times 0.12$
Set ExtraUtilDiff = Abs (ExtraUtilNew – ExtraUtilOld)
If ExtraUtilDiff < 0.001
Set StopDueToAccuracy = True
Set ExtraUtilOld = ExtraUtilNew
$U^S = U^S +$ ***ExtraUtilNew***
}
Calculate CPU Utilizations U^C

Here the extra utilization due to board activity is distributed evenly the buses that is every bus in the array gets the same extra utilization value.

9. Components Polling Calculation

After the procedure *Component Utilization Calculation* is performed once the procedure *Components Polling Calculation* is invoked which in turn invokes *Component Utilization Calculation* until convergence for the additional

CPU polling workload is reached. The calculation of CPU polling and its impact on the components utilization is coupled with the components utilization calculation discussed in 8.

Components Polling Calculation

```
Set CurrentIter = 0
 Continue = Calculate Polling Workload (250)
While (CurrentIter < 10 And Continue == True)
 {
   Distribute Polling Workload To Components
  Components Utilization Calculation
  Continue = Calculate Polling Workload (250)
  CurrentIter = CurrentIter + 1
}
```

In the procedure above Calculate Polling Workload is performed as shown below:

Bool Calculate Polling Workload(Limit)

```
Bool Continue = False
For Every CPU
{
  Calculate internal Polling workload using (3.1) and (3.2)
  Calculate total number of elementary ops: N_EO = N_DSA + N_DMA + N_HL + N_CPU
  If N_EO > Limit
      Continue = True
}

Return Continue
```

Appendix:

A.1. Quantities denoting the number of Direct Storage Accesses N_{DSA}, Direct Memory Accesses N_{DMA}, Hardware Locks N_{HL} as well as Transfer Factors $F_{TR}$and Memory Transfer Factors F_{MTR}:

N_{DSA}^{RH}, N_{DMA}^{RH} , N_{HL}^{RH} , F_{TR}^{RH} , F_{MTR}^{RH} – number of DSA, DMA, HL ops and Transfer Factors due to 1 read hit
For VMAX-based models:
Fibre: $N_{DSA}^{RH} = 6$, $N_{DMA}^{RH} = 0$, $N_{HL}^{RH} = 2$,
$F_{RH}^{TR} = m \times N_{node} + n$ where N_{node} is the number of system nodes with $n = 1.404; m = 0.145$

N_{DSA}^{RM}, N_{DMA}^{RM} , N_{HL}^{RM} , F_{TR}^{RM}, F_{MTR}^{RM} – number of DSA, DMA, HL ops and Transfer Factors due to 1 read miss
Fibre: $N_{DSA}^{RH} = 11$, $N_{DMA}^{RH} = 0$, $N_{HL}^{RH} = 2$, $F_{RH}^{TR} = 1$

N_{DSA}^{SR}, N_{DMA}^{SR} , N_{HL}^{SR} , F_{TR}^{SR}, F_{MTR}^{SR} – number DSA, DMA, HL ops due to 1 sequential read
N_{DSA}^{WH}, N_{DMA}^{WH} , N_{HL}^{WH} , F_{TR}^{WH}, F_{MTR}^{WH} – number of DSA, DMA, HL ops due to 1 write hit
For VMAX-based models:
Fibre: $N_{DSA}^{WH} = 12$, $N_{DMA}^{WH} = 0$, $N_{HL}^{WH} = 10$,
$F_{WH}^{TR} = m \times N_{node} + n$ where N_{node} is the number of system nodes with $n = 2.458; m = 0.242$

N_{DSA}^{WM}, N_{DMA}^{WM} , N_{HL}^{WM} , F_{TR}^{WM}, F_{MTR}^{WM} – number of DSA, DMA, HL ops due to 1 write miss
N_{DSA}^{SW}, N_{DMA}^{SW} , N_{HL}^{SW} , F_{TR}^{SW}, F_{MTR}^{SW} – number of DSA, DMA, HL ops due to 1 sequential write
N_{DSA}^{CR}, N_{DMA}^{CR} , N_{HL}^{CR}, F_{TR}^{CR}, F_{MTR}^{CR} – number of DSA, DMA, HL ops due to 1 copy read
N_{DSA}^{CW}, N_{DMA}^{CW} , N_{HL}^{CW} , F_{TR}^{CW}, F_{MTR}^{CW} – number of DSA, DMA, HL ops due to 1 copy write

$N_{DSA}^{RH,VD}$, $N_{DMA}^{RH,VD}$, $N_{HL}^{RH,VD}$, $F_{TR}^{RH,VD}$, $F_{MTR}^{RH,VD}$ – number of DSA, DMA, HL ops due to 1 VDEV read hit
$N_{DSA}^{RM,VD}$, $N_{DMA}^{RM,VD}$, $N_{HL}^{RM,VD}$, $F_{TR}^{RM,VD}$, $F_{MTR}^{RM,VD}$ – number of DSA, DMA, HL ops due to 1 VDEV read miss
$N_{DSA}^{SR,VD}$, $N_{DMA}^{SR,VD}$, $N_{HL}^{SR,VD}$, $F_{TR}^{SR,VD}$, $F_{MTR}^{SR,VD}$ – number of DSA, DMA, HL ops due to 1 VDEV sequential read miss
$N_{DSA}^{WH,VD}$, $N_{DMA}^{WH,VD}$, $N_{HL}^{WH,VD}$, $F_{TR}^{WH,VD}$, $F_{MTR}^{WH,VD}$ – number of DSA, DMA, HL ops due to 1 VDEV write hit
$N_{DSA}^{WM,VD}$, $N_{DMA}^{WM,VD}$, $N_{HL}^{WM,VD}$, $F_{TR}^{WM,VD}$, $F_{MTR}^{WM,VD}$ – number of DSA, DMA, HL ops due to 1 VDEV write miss
$N_{DSA}^{SW,VD}$, $N_{DMA}^{SW,VD}$, $N_{HL}^{SW,VD}$, $F_{TR}^{SW,VD}$, $F_{MTR}^{SW,VD}$ – number of DSA, DMA, HL ops due to 1 VDEV sequential write

$N_{DSA}^{RH,TD}$, $N_{DMA}^{RH,TD}$, $N_{HL}^{RH,TD}$, $F_{TR}^{RH,TD}$, $F_{MTR}^{RH,TD}$ – number of DSA, DMA, HL ops due to 1 TDEV read hit
$N_{DSA}^{RM,TD}$, $N_{DMA}^{RM,TD}$, $N_{HL}^{RM,TD}$, $F_{TR}^{RM,TD}$, $F_{MTR}^{RM,TD}$ – number of DSA, DMA, HL ops due to 1 TDEV read miss
$N_{DSA}^{SR,TD}$, $N_{DMA}^{SR,TD}$, $N_{HL}^{SR,TD}$, $F_{TR}^{SR,TD}$, $F_{MTR}^{SR,TD}$ – number of DSA, DMA, HL ops due to 1 TDEV sequential read miss
$N_{DSA}^{WH,TD}$, $N_{DMA}^{WH,TD}$, $N_{HL}^{WH,TD}$, $F_{TR}^{WH,TD}$, $F_{MTR}^{WH,TD}$ – number of DSA, DMA, HL ops due to 1 TDEV write hit
$N_{DSA}^{WM,TD}$, $N_{DMA}^{WM,TD}$, $N_{HL}^{WM,TD}$, $F_{TR}^{WM,TD}$, $F_{MTR}^{WM,TD}$ – number of DSA, DMA, HL ops due to 1 TDEV write miss
$N_{DSA}^{SW,TD}$, $N_{DMA}^{SW,TD}$, $N_{HL}^{SW,TD}$, $F_{TR}^{SW,TD}$, $F_{MTR}^{SW,TD}$ – number of DSA, DMA, HL ops due to 1 TDEV sequential write

$N_{DSA}^{WH,R1}$, $N_{DMA}^{WH,R1}$, $N_{HL}^{WH,R1}$, $F_{TR}^{WH,R1}$, $F_{MTR}^{WH,R1}$ – number of DSA, DMA, HL ops due to 1 RDF write hit J0 R1
$N_{DSA}^{WM,R1}$, $N_{DMA}^{WM,R1}$, $N_{HL}^{WM,R1}$, $F_{TR}^{WM,R1}$, $F_{MTR}^{WM,R1}$ – number of DSA, DMA, HL ops due to 1 RDF write miss J0 R1
$N_{DSA}^{SW,R1}$, $N_{DMA}^{SW,R1}$, $N_{HL}^{SW,R1}$, $F_{TR}^{SW,R1}$, $F_{MTR}^{SW,R1}$ – number of DSA, DMA, HL ops due to 1 RDF seq write J0 R1
$N_{DSA}^{WM,R2}$, $N_{DMA}^{WM,R2}$, $N_{HL}^{WM,R2}$, $F_{TR}^{WM,R2}$, $F_{MTR}^{WM,R2}$ – number of DSA, DMA, HL ops due to 1 RDF write miss R2
$N_{DSA}^{CD,R1}$, $N_{DMA}^{CD,R1}$, $N_{HL}^{CD,R1}$, $F_{TR}^{CD,R1}$, $F_{MTR}^{CD,R1}$ – number of DSA, DMA, HL ops due to 1 RDF copy disk R1

$N_{DSA}^{CWP,R1}, N_{DMA}^{CWP,R1}, N_{HL}^{CWP,R1}, F_{TR}^{CWP,R1}, F_{MTR}^{CWP,R1}$ – number of DSA ops due to RDF copy WP R1

For VMAX-based models the Transfer Factor is given as:
Read hits:
Read Miss: $F_{RM}^{TR} = 1$
Write Miss: $F_{WM}^{TR} = 2$
Seq Read: $F_{SR}^{TR} = 1$
Copy Read: $F_{WM}^{TR} = 2$
Copy Write: $F_{WM}^{TR} = 1$

RDF write hit J0 R1: $F_{RM}^{TR} = 1$
RDF write miss J0 R1: $F_{RM}^{TR} = 1$
RDF write hit R2: $F_{RM}^{TR} = 1$
RDF write miss R2: $F_{RM}^{TR} = 1$

A.2. Time coefficients for calculating the component busy time contribution due to various io types

The busy time of component $comp$ due to io of type $iotype$ is given with

$$T_{iotype}^{comp} = a_{iotype}^{comp} + b_{iotype}^{comp} \times R(U)$$

where a_{iotype}^{comp} and b_{iotype}^{comp} are the time coefficients measured in nanoseconds and $R(U)$ is the utilization impact function.

A.3 CPU polling coefficients

P_{DSA} , P_{DMA} and P_{HL} are the CPU polling coefficients and they depend on the CPU type (back-end, front-end, RDF) and Symmetrix model.

A.4 Maximum throughput which a channel can sustain

T_{max}^{P} =