

Status of HDF5 usage at ITER

Lana Abadie ¹, Rodrigo Castro ², Yury Makushok ³

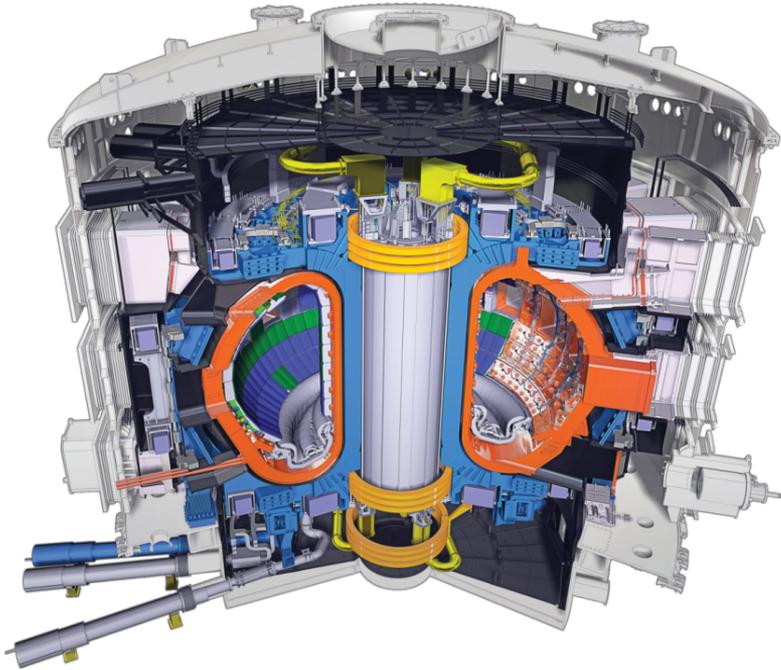
¹ ITER Organization

² CIEMAT

³ INDRA

Disclaimer: The views and opinions expressed herein do not necessarily reflect those of the ITER Organization

ITER



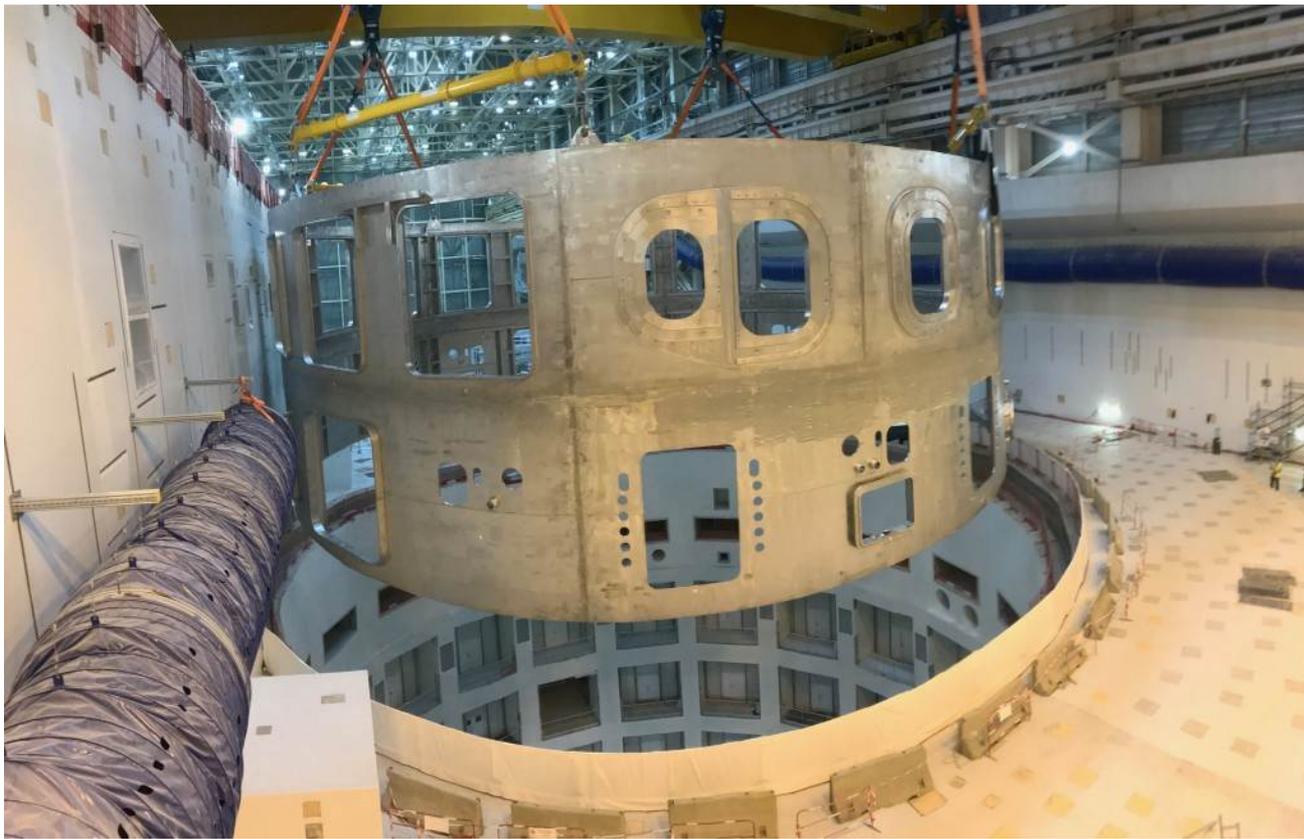
- 7 ITER members (China, E.U, India, Japan, Russia, South Korea, U.S.A) make cash in kind-contributions to the ITER project
- Demonstrate feasibility of fusion energy for peaceful purposes
- A plasma of deuterium and tritium (hydrogen isotopes) is heated to more than 150 millions °C
- The hot plasma is shaped and confined by strong magnetic fields
- Helium nuclei sustain burning plasma
- Neutron transfers energy to blanket
- Conventional steam generator, turbine and motor will transform the heat into electricity

Pictures



Lower base cylinder 1250t ,
height 6.5m, diameter 30m
manufactured by India going
down to the pit (~20m down)

Pictures



Lower cylinder
(375 tons)
manufactured by
India going down



1 TF coil (~300tons)
manufactured by Europe and
now on ITER site



1 Vacuum vessel sector (~800
tons) manufactured by Korea
and now on ITER site

Complex structure with HDF5 - requirements

- Be able to continuously archive complex structure at 10Khz for many days
- Be able to access one particular item (atomic datatype and usually corresponds to a leaf of the tree) for a given pulse (~ time window) within seconds
- Be able to access the full data structure for a given pulse (~ time window) within seconds
- Be able to access data while it is written
- Be able to retrieve the structure for a given pulse (it is constant)

Complex structure with HDF5 - Brainstorming

- Solution 1 : implement using one compound type and nested datatype
 - Bad performance in case of accessing one leaf of the tree as we need to load
 - Potential limitation of HDF5 library
- Solution 2 : use of compound type and flatten the structure
 - Good performance
 - But limited to 1024 attributes...
- Solution 3 : a mixed solution
 - Use of multiple HDF5 group and array of structures with flatten members when needed
 - Store the model as an HDF5 attribute to be self-consistent
 - Import the model in a database as Json
 - Extend our Data Access server to serve the request

Complex structure with HDF5 - requirements

```

typedef struct Data_t {
    int32_t State;
    int32_t Quality;
    float Value;
    float Error;
} __attribute__((packed)) Data_t;

typedef struct SensorInfo_t {
    Data_t Integrated;
    Data_t Proportional;
    Data_t IntegratedFiltered;
    Data_t ProportionalFiltered;
    Data_t Combined;
    Data_t CombinedIntegrated;
    Data_t Temperature;
    uint32_t ErrorFlags;
} __attribute__((packed)) SensorInfo_t;

typedef struct FPGAVoltageErr_t {
    uint32_t PLInternal;
    uint32_t PLAuxil[4];
    uint32_t PLBlockRAM;
    uint32_t PSLowPowerDomain;
    uint32_t PSAuxil[4];
} __attribute__((packed)) FPGAVoltageErr_t;

typedef struct FPGAVoltages_t {
    float PLInternal;
    float PLAuxil[4];
    float PLBlockRAM;
    float PSLowPowerDomain;
    float PSAuxil[4];
} __attribute__((packed)) FPGAVoltages_t;

typedef struct FPGAIInfo_t {
    FPGAVoltages_t Vols;
    float Temperature;
    uint32_t TemperatureAlarmCount;
    FPGAVoltageErr_t VoltAIC;
    uint32_t ClockErrorsCount;
    float ClockFrequencies[6];
    uint32_t InternalErrorsCount;
} __attribute__((packed)) FPGAIInfo_t;

typedef struct MainFPGAInfo_t {
    FPGAIInfo_t FPGAMon;
    uint32_t ConfigHash;
} __attribute__((packed)) MainFPGAInfo_t;

typedef struct main_t {
    SDNHeader_t SDNHeader;
    Time_t Time;
    SensorInfo_t Sensor[32];
    MainFPGAInfo_t FPGAIInfo;
} __attribute__((packed)) main_t;

```

Key	Value
Root	
main	
SDNHeader	
Time	uint64
Sensor[32]	
Integrated	
State	int32
Quality	int32
Value	float32
Error	float32
Proportional	
IntegratedFiltered	
ProportionalFiltered	
Combined	
CombinedIntegrated	
Temperature	
ErrorFlags	uint32
FPGAIInfo	
FPGAMon	
Vols	
PLInternal	float32
PLAuxil[4]	float32
PLBlockRAM	float32
PSLowPowerDomain	float32
PSAuxil[4]	float32
Temperature	float32
TemperatureAlarmCount	uint32
VoltAIC	
PLInternal	uint32
PLAuxil[4]	uint32
PLBlockRAM	uint32
PSLowPowerDomain	uint32
PSAuxil[4]	uint32
ClockErrorsCount	uint32
ClockFrequencies[6]	float32
InternalErrorsCount	uint32
ConfigHash	uint32

```

[abadiel@trunk-2 Downloads]$ h5dump -H -d /DataStream_2/p
HDF5 "/tmp/sdn-record/55A0FPGA0_sdn_archiver.h5" {
DATASET "/DataStream_2/payloads" {
  DATATYPE H5T_COMPOUND {
    H5T_STD_I32LE "Integrated#State";
    H5T_STD_I32LE "Integrated#Quality";
    H5T_IEEE_F32LE "Integrated#Value";
    H5T_IEEE_F32LE "Integrated#Error";
    H5T_STD_I32LE "Proportional#State";
    H5T_STD_I32LE "Proportional#Quality";
    H5T_IEEE_F32LE "Proportional#Value";
    H5T_IEEE_F32LE "Proportional#Error";
    H5T_STD_I32LE "IntegratedFiltered#State";
    H5T_STD_I32LE "IntegratedFiltered#Quality";
    H5T_IEEE_F32LE "IntegratedFiltered#Value";
    H5T_IEEE_F32LE "IntegratedFiltered#Error";
    H5T_STD_I32LE "ProportionalFiltered#State";
    H5T_STD_I32LE "ProportionalFiltered#Quality";
    H5T_IEEE_F32LE "ProportionalFiltered#Value";
    H5T_IEEE_F32LE "ProportionalFiltered#Error";
    H5T_STD_I32LE "Combined#State";
    H5T_STD_I32LE "Combined#Quality";
    H5T_IEEE_F32LE "Combined#Value";
    H5T_IEEE_F32LE "Combined#Error";
    H5T_STD_I32LE "CombinedIntegrated#State";
    H5T_STD_I32LE "CombinedIntegrated#Quality";
    H5T_IEEE_F32LE "CombinedIntegrated#Value";
    H5T_IEEE_F32LE "CombinedIntegrated#Error";
    H5T_STD_I32LE "Temperature#State";
    H5T_STD_I32LE "Temperature#Quality";
    H5T_IEEE_F32LE "Temperature#Value";
    H5T_IEEE_F32LE "Temperature#Error";
    H5T_STD_U32LE "ErrorFlags";
  }
  DATASPACE SIMPLE { ( 32000 ) / ( H5S_UNLIMITED ) }
}
}

```

Preliminary results

- Functional tests are satisfactory
- Performance tests are on-going
- Use of HDF5 SWMR, 1.12.alpha-0 and 1.10.3

Test on ramfs

sdn	8.6G	atca-7480-1	HYPERSLA, CCS6.2	1000	65.36 ms	1 sdn file + HTSCL files	ramfs	1
sdn	8.6G	atca-7480-1	HYPERSLA, CCS6.2	1000	61.76	1 sdn file + HTSCL files	ramfs	1
sdn	8.6G	atca-7480-1	HYPERSLA, CCS6.2	1000	62.26	1 sdn file + HTSCL files	ramfs	1
sdn	8.6G	atca-7480-1	HYPERSLA, CCS6.2	10000	397.61	1 sdn file + HTSCL files	ramfs	1
sdn	8.6G	atca-7480-1	HYPERSLA, CCS6.2	10000	378.24	1 sdn file + HTSCL files	ramfs	1
sdn	8.6G	atca-7480-1	HYPERSLA, CCS6.2	10000	361.72	1 sdn file + HTSCL files	ramfs	1
sdn	8.6G	atca-7480-1	HYPERSLA, CCS6.2	100000	3534.134	1 sdn file + HTSCL files	ramfs	1
sdn	8.6G	atca-7480-1	HYPERSLA, CCS6.2	100000	2351.019	1 sdn file + HTSCL files	ramfs	1
sdn	8.6G	atca-7480-1	HYPERSLA, CCS6.2	100000	2312.08	1 sdn file + HTSCL files	ramfs	1
sdn	8.6G	atca-7480-1	HYPERSLA, CCS6.2	Min,max,avg	6628.278	1 sdn file + HTSCL files	ramfs	1
sdn	8.6G	atca-7480-1	HYPERSLA, CCS6.2	Min,max,avg	5686.221	1 sdn file + HTSCL files	ramfs	2
sdn	8.6G	atca-7480-1	HYPERSLA, CCS6.2	Min,max,avg	5663.119	1 sdn file + HTSCL files	ramfs	3

Test on GPFS

sdn	8.6G	18gi00-pcf-0001	HYPERSLA, CCS6.2	uda-get-data 18gi00-pcf-0001 "variable=sdn-one-pps/identifier,startTime=0,endTime=-1,decSamples=1000"	3970.184 only one file indexed	SDN_TO	1
sdn	8.6G	18gi00-pcf-0001	HYPERSLA, CCS6.2	uda-get-data 18gi00-pcf-0001 "variable=sdn-one-pps/identifier,startTime=0,endTime=-1,decSamples=1000"	85.587 only one file indexed	SDN_TO	2
sdn	8.6G	18gi00-pcf-0001	HYPERSLA, CCS6.2	uda-get-data 18gi00-pcf-0001 "variable=sdn-one-pps/identifier,startTime=0,endTime=-1,decSamples=1000"	79.552 only one file indexed	SDN_TO	3
sdn	8.6G	18gi00-pcf-0001	HYPERSLA, CCS6.2	uda-get-data 18gi00-pcf-0001 "variable=sdn-one-pps/identifier,startTime=0,endTime=-1,decSamples=10000"	16414.74 only one file indexed	SDN_TO	1
sdn	8.6G	18gi00-pcf-0001	HYPERSLA, CCS6.2	uda-get-data 18gi00-pcf-0001 "variable=sdn-one-pps/identifier,startTime=0,endTime=-1,decSamples=10000"	7815.047 only one file indexed	SDN_TO	2
sdn	8.6G	18gi00-pcf-0001	HYPERSLA, CCS6.2	uda-get-data 18gi00-pcf-0001 "variable=sdn-one-pps/identifier,startTime=0,endTime=-1,decSamples=10000"	8625.659 only one file indexed	SDN_TO	3
sdn	8.6G	18gi00-pcf-0001	HYPERSLA, CCS6.2	uda-get-data 18gi00-pcf-0001 "variable=sdn-one-pps/identifier,startTime=0,endTime=-1,decSamples=100000"	49349 only one file indexed	SDN_TO	1
sdn	8.6G	18gi00-pcf-0001	HYPERSLA, CCS6.2	uda-get-data 18gi00-pcf-0001 "variable=sdn-one-pps/identifier,startTime=0,endTime=-1,decSamples=100000"	50817.25 only one file indexed	SDN_TO	2
sdn	8.6G	18gi00-pcf-0001	HYPERSLA, CCS6.2	uda-get-data 18gi00-pcf-0001 "variable=sdn-one-pps/identifier,startTime=0,endTime=-1,decSamples=100000"	49622.15 only one file indexed	SDN_TO	3
sdn	8.6G	18gi00-pcf-0001	HYPERSLA, CCS6.2	uda-get-data-info 18gi00-pcf-0001 "variable=sdn-one-pps/identifier,startTime=0,endTime=-1,decSamples=-1"	52286.89 only one file indexed	SDN_TO	1
sdn	8.6G	18gi00-pcf-0001	HYPERSLA, CCS6.2	uda-get-data-info 18gi00-pcf-0001 "variable=sdn-one-pps/identifier,startTime=0,endTime=-1,decSamples=-1"	53288.64 only one file indexed	SDN_TO	2
sdn	8.6G	18gi00-pcf-0001	HYPERSLA, CCS6.2	uda-get-data-info 18gi00-pcf-0001 "variable=sdn-one-pps/identifier,startTime=0,endTime=-1,decSamples=-1"	53673.57 only one file indexed	SDN_TO	3

FileSystems Test

- Challenge :
 - Archive 30GB/sec for 1 hour data from sensors (~3000 sensors at 2MHz, sample size if 4bytes), investigation if data can be well compressed
 - Dataflow is *2 one in internal network and one external network -> 2*30GB/sec on separate storage
 - Allow data access while writing
 - Plan is to have 30 writers (different machines) – 30 files, use of SWMR, in parallel, prepare a h5file to store downsampled data with min/max/avg to speed up data retrieval
- Candidate GPFS, Lustre, if possible testing of HSDS...
- Started with GPFS
 - Limited hardware : Test with one writer at 1 GB/sec and 2 writers at 500MB/sec with data read access and no loss
 - Main issue is small read IO (heavy usage of select hyperslab)

Conclusions

- Good compromise found to store HDF5 data
- Good support from the HDF group
- A lot of good new features but unfortunately lack of budget and resources.
- We hope to report on our results