ORCHIDEE-MICT

Profiling

Table of Contents

Overview

This document tries to understand Orchidee MICT computing time behavior. In the latest version 6.5 it takes a lot of time to compute. Around 8h in 0.5 degrees for 1 year. So it is necessary to understand why It happens. Once the problems are identified it is possible to apply different solutions for each issue.

In order to make such thing possible the code is profiled. Different tools are used. They provide an easy way to identify basic hotspots in the code.

History file size

Overview

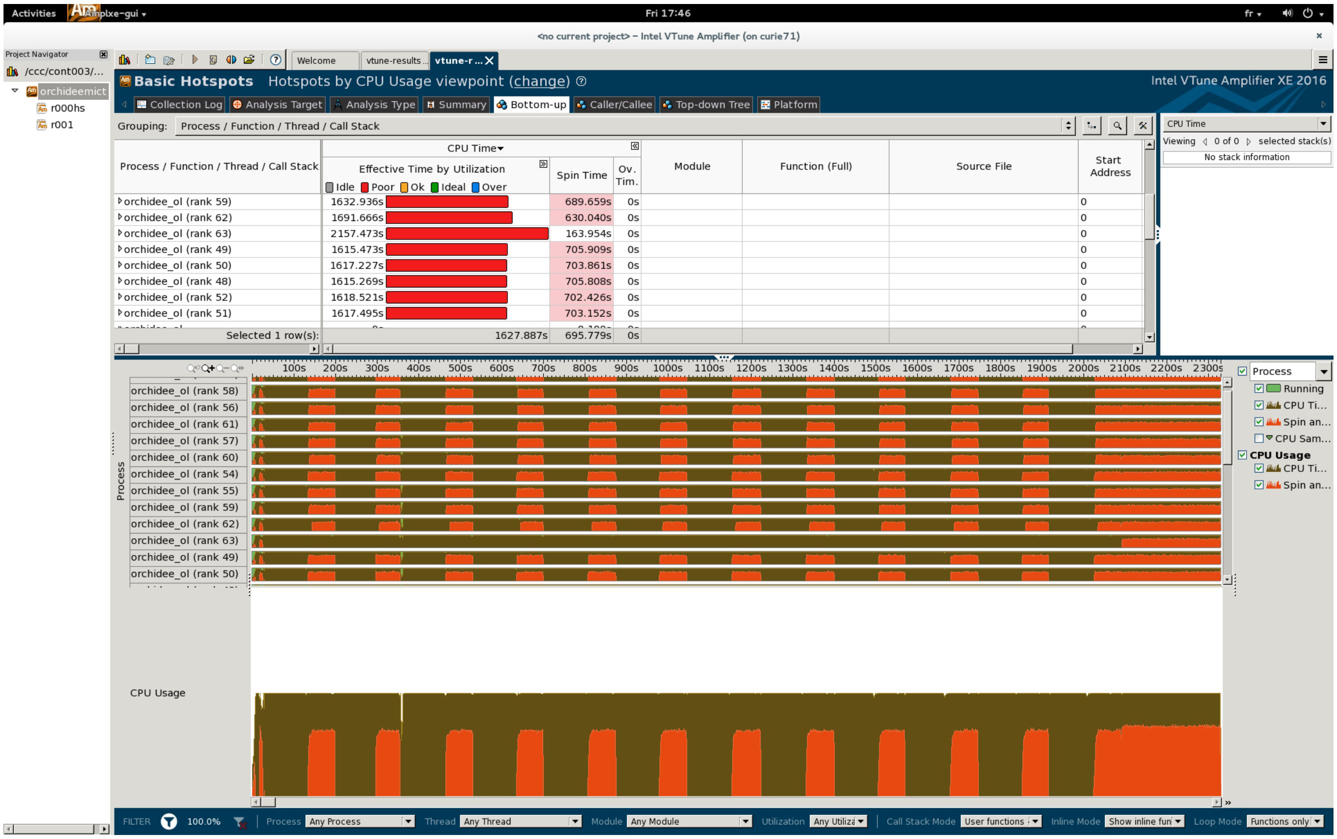
Orchidee history output files have different file sizes. It might be interesting to understand why. This could lead to major issues or not.

Simulation description

Orchidee MICT revision 3526 (interpolation, on going development):

- First year
- 64 cores
- 1 year
- 1 degree
- IOIPSL library
- Monthly history
- Total time: ~2350 seconds

Profiling (Intel vTune)



<no current project> – Intel VTune Amplifier (on curie71)

Basic Hotspots Hotspots by CPU Usage viewpoint (change)

Intel VTune Amplifier XE 2016

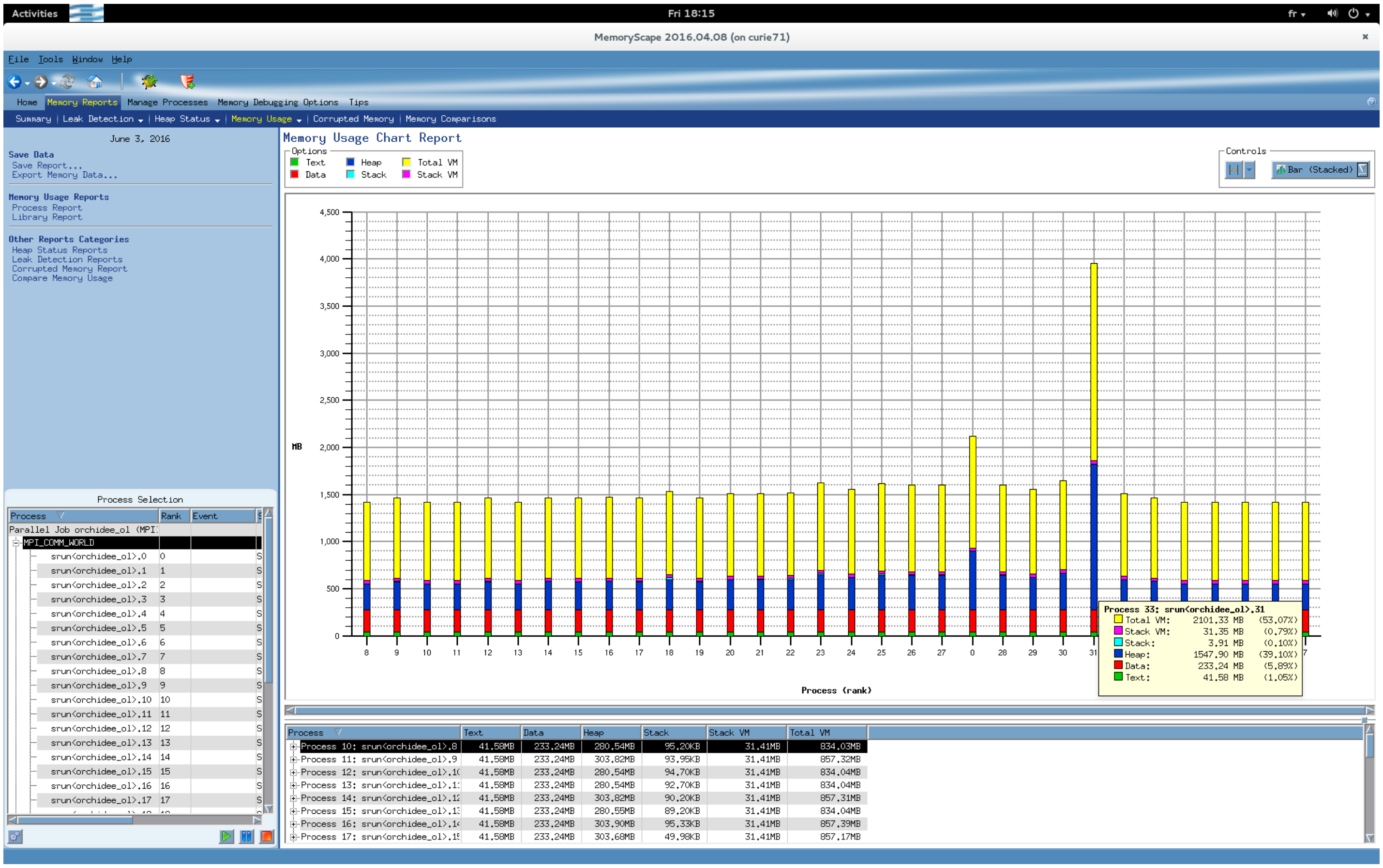
Grouping: Process / Function / Thread / Call Stack

Process / Function / Thread / Call Stack	CPU Time: Effective Time by Utilization	CPU Time: Spin Time	CPU Time: Ov. Tim.	Module	Function (Full)	Source File	Start Address
orchidee_ol (rank 59)	1632.936s	689.659s	0s				0
orchidee_ol (rank 62)	1691.666s	630.040s	0s				0
orchidee_ol (rank 63)	2157.473s	163.954s	0s				0
orchidee_ol (rank 49)	1615.473s	705.909s	0s				0
orchidee_ol (rank 50)	1617.227s	703.861s	0s				0
orchidee_ol (rank 48)	1615.269s	705.808s	0s				0
orchidee_ol (rank 52)	1618.521s	702.426s	0s				0
orchidee_ol (rank 51)	1617.495s	703.152s	0s				0
Selected 1 row(s):	1627.887s	695.779s	0s				

Red color: the processor is waiting. It is doing nothing.
Middle red colors (12 months): at the end of every month history is written.
Latest red part: restart files are written.
Latest processor is always unbalanced. It is slower than the rest. Due to MPI blocking calls they need to wait.

Memory (Totalview)

Note: this is a test with 32 processors. Randomly stop at the beginning to show off the memory usage:



Process 33: srun<orchidee_ol>.31

🟨 Total VM:	2101.33 MB	(53.07%)
Stack VM:	31.35 MB	(0.79%)
Stack:	3.91 MB	(0.10%)
🟦 Heap:	1547.90 MB	(39.10%)
🟥 Data:	233.24 MB	(5.89%)
🟩 Text:	41.58 MB	(1.05%)

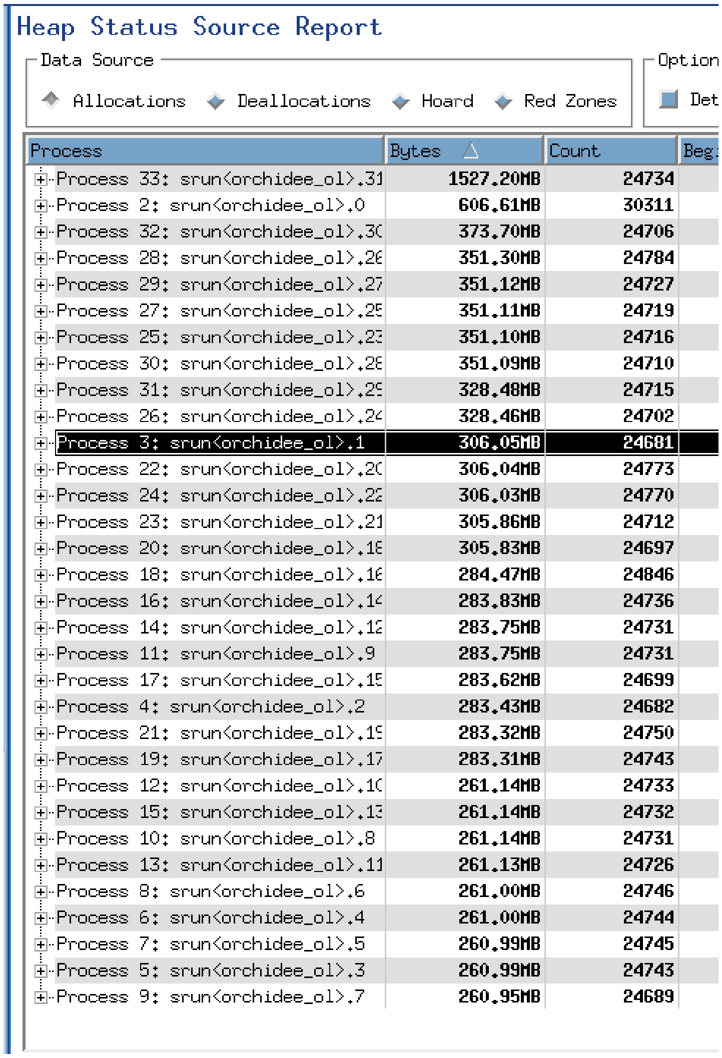
Process	Rank	Event
Parallel Job orchidee_ol (MPI		
MPI_COMM_WORLD		
srun<orchidee_ol>.0	0	
srun<orchidee_ol>.1	1	
srun<orchidee_ol>.2	2	
srun<orchidee_ol>.3	3	
srun<orchidee_ol>.4	4	
srun<orchidee_ol>.5	5	
srun<orchidee_ol>.6	6	
srun<orchidee_ol>.7	7	
srun<orchidee_ol>.8	8	
srun<orchidee_ol>.9	9	
srun<orchidee_ol>.10	10	
srun<orchidee_ol>.11	11	
srun<orchidee_ol>.12	12	
srun<orchidee_ol>.13	13	
srun<orchidee_ol>.14	14	
srun<orchidee_ol>.15	15	
srun<orchidee_ol>.16	16	
srun<orchidee_ol>.17	17	

Process	Text	Data	Heap	Stack	Stack VM	Total VM
Process 10: srun<orchidee_ol>.8	41.58MB	233.24MB	280.54MB	95.20KB	31.41MB	834.03MB
Process 11: srun<orchidee_ol>.9	41.58MB	233.24MB	303.82MB	93.95KB	31.41MB	857.32MB
Process 12: srun<orchidee_ol>.1(	41.58MB	233.24MB	280.54MB	94.70KB	31.41MB	834.04MB
Process 13: srun<orchidee_ol>.1:	41.58MB	233.24MB	280.54MB	92.70KB	31.41MB	834.04MB
Process 14: srun<orchidee_ol>.12	41.58MB	233.24MB	303.82MB	90.20KB	31.41MB	857.31MB
Process 15: srun<orchidee_ol>.13	41.58MB	233.24MB	280.55MB	89.20KB	31.41MB	834.04MB
Process 16: srun<orchidee_ol>.14	41.58MB	233.24MB	303.90MB	95.33KB	31.41MB	857.39MB
Process 17: srun<orchidee_ol>.15	41.58MB	233.24MB	303.68MB	49.98KB	31.41MB	857.17MB

The latest processors uses more memory:

- The heap (allocate/deallocate) is much higher
- VM is a consequence of the Heap size

Let's take a deeper look. Process 33 (the latest) uses more memory.

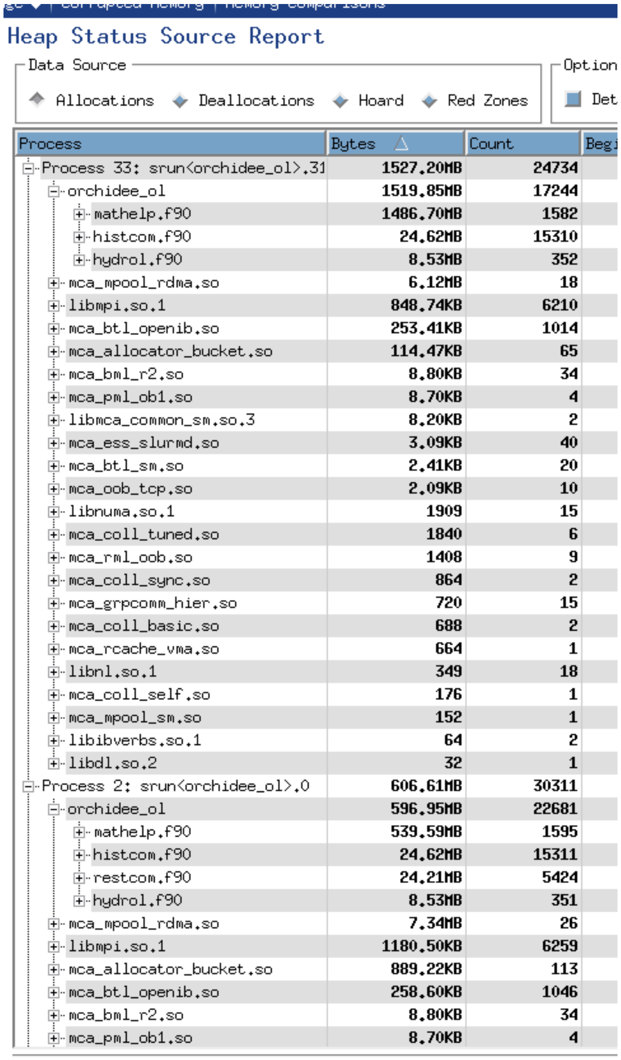


Heap Status Source Report

Data Source: ◆ Allocations ◆ Deallocations ◆ Hoard ◆ Red Zones

Option: □ Det

Process	Bytes △	Count	Beg
⊞ Process 33: srun<orchidee_ol>.31	1527.20MB	24734	
⊞ Process 2: srun<orchidee_ol>.0	606.61MB	30311	
⊞ Process 32: srun<orchidee_ol>.30	373.70MB	24706	
⊞ Process 28: srun<orchidee_ol>.26	351.30MB	24784	
⊞ Process 29: srun<orchidee_ol>.27	351.12MB	24727	
⊞ Process 27: srun<orchidee_ol>.25	351.11MB	24719	
⊞ Process 25: srun<orchidee_ol>.23	351.10MB	24716	
⊞ Process 30: srun<orchidee_ol>.28	351.09MB	24710	
⊞ Process 31: srun<orchidee_ol>.29	328.48MB	24715	
⊞ Process 26: srun<orchidee_ol>.24	328.46MB	24702	
⊞ Process 3: srun<orchidee_ol>.1	306.05MB	24681	
⊞ Process 22: srun<orchidee_ol>.20	306.04MB	24773	
⊞ Process 24: srun<orchidee_ol>.22	306.03MB	24770	
⊞ Process 23: srun<orchidee_ol>.21	305.86MB	24712	
⊞ Process 20: srun<orchidee_ol>.18	305.83MB	24697	
⊞ Process 18: srun<orchidee_ol>.16	284.47MB	24846	
⊞ Process 16: srun<orchidee_ol>.14	283.83MB	24736	
⊞ Process 14: srun<orchidee_ol>.12	283.75MB	24731	
⊞ Process 11: srun<orchidee_ol>.9	283.75MB	24731	
⊞ Process 17: srun<orchidee_ol>.15	283.62MB	24699	
⊞ Process 4: srun<orchidee_ol>.2	283.43MB	24682	
⊞ Process 21: srun<orchidee_ol>.19	283.32MB	24750	
⊞ Process 19: srun<orchidee_ol>.17	283.31MB	24743	
⊞ Process 12: srun<orchidee_ol>.10	261.14MB	24733	
⊞ Process 15: srun<orchidee_ol>.13	261.14MB	24732	
⊞ Process 10: srun<orchidee_ol>.8	261.14MB	24731	
⊞ Process 13: srun<orchidee_ol>.11	261.13MB	24726	
⊞ Process 8: srun<orchidee_ol>.6	261.00MB	24746	
⊞ Process 6: srun<orchidee_ol>.4	261.00MB	24744	
⊞ Process 7: srun<orchidee_ol>.5	260.99MB	24745	
⊞ Process 5: srun<orchidee_ol>.3	260.99MB	24743	
⊞ Process 9: srun<orchidee_ol>.7	260.95MB	24689	



Heap Status Source Report

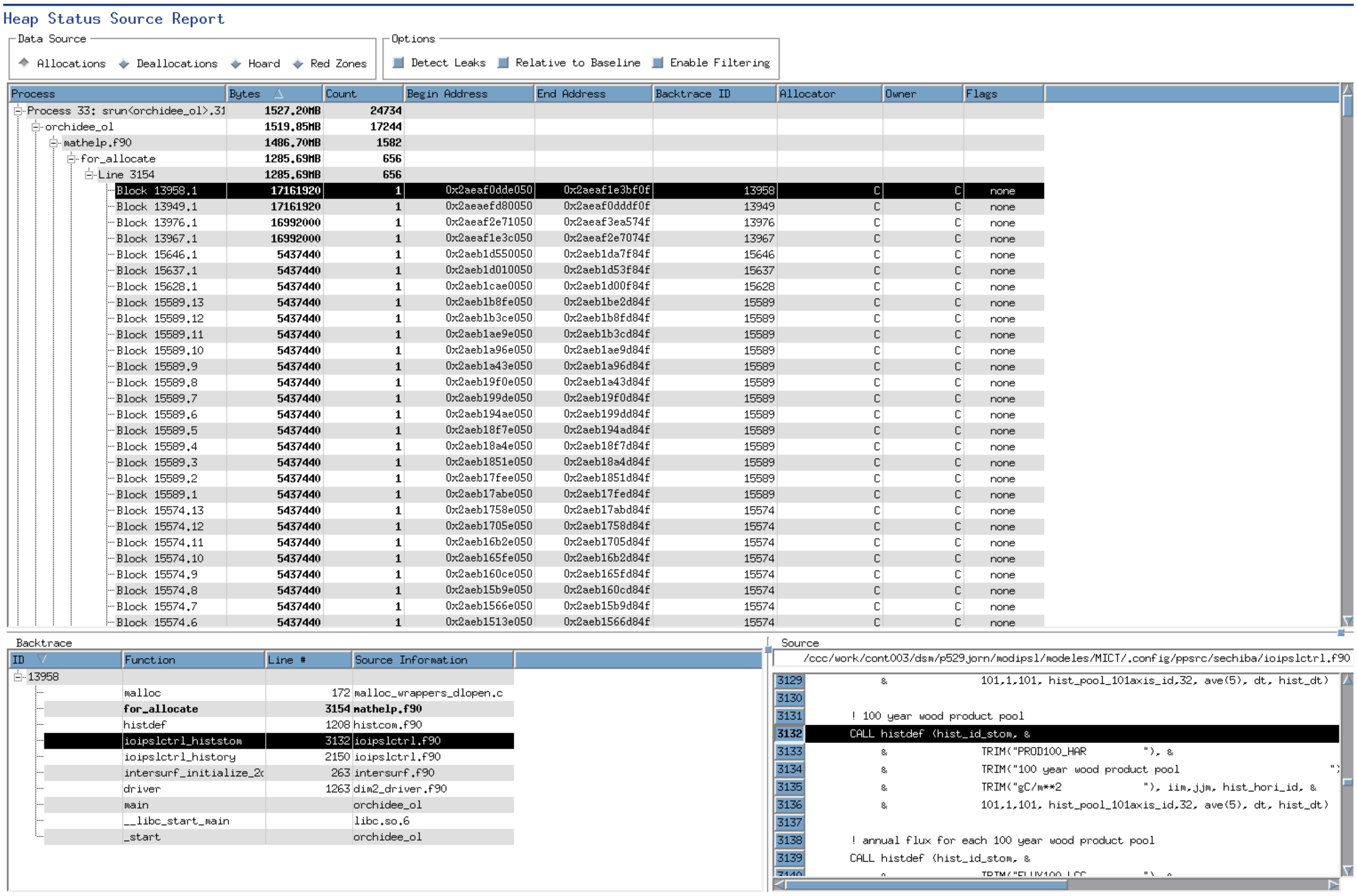
Data Source: ◆ Allocations ◆ Deallocations ◆ Hoard ◆ Red Zones

Option: □ Det

Process	Bytes △	Count	Beg
⊟ Process 33: srun<orchidee_ol>.31	1527.20MB	24734	
⊟ orchidee_ol	1519.85MB	17244	
⊞ mathelp.f90	1486.70MB	1582	
⊞ histcom.f90	24.62MB	15310	
⊞ hydrol.f90	8.53MB	352	
⊞ mca_mpool_rdma.so	6.12MB	18	
⊞ libmpi.so.1	848.74KB	6210	
⊞ mca_btl_openib.so	253.41KB	1014	
⊞ mca_allocator_bucket.so	114.47KB	65	
⊞ mca_bml_r2.so	8.80KB	34	
⊞ mca_pml_ob1.so	8.70KB	4	
⊞ libmca_common_sm.so.3	8.20KB	2	
⊞ mca_ess_slurmd.so	3.09KB	40	
⊞ mca_btl_sm.so	2.41KB	20	
⊞ mca_oob_tcp.so	2.09KB	10	
⊞ libnuma.so.1	1909	15	
⊞ mca_coll_tuned.so	1840	6	
⊞ mca_rml_oob.so	1408	9	
⊞ mca_coll_sync.so	864	2	
⊞ mca_grpcomm_hier.so	720	15	
⊞ mca_coll_basic.so	688	2	
⊞ mca_rcache_vma.so	664	1	
⊞ libnl.so.1	349	18	
⊞ mca_coll_self.so	176	1	
⊞ mca_mpool_sm.so	152	1	
⊞ libibverbs.so.1	64	2	
⊞ libdl.so.2	32	1	
⊟ Process 2: srun<orchidee_ol>.0	606.61MB	30311	
⊟ orchidee_ol	596.95MB	22681	
⊞ mathelp.f90	539.59MB	1595	
⊞ histcom.f90	24.62MB	15311	
⊞ restcom.f90	24.21MB	5424	
⊞ hydrol.f90	8.53MB	351	
⊞ mca_mpool_rdma.so	7.34MB	26	
⊞ libmpi.so.1	1180.50KB	6259	
⊞ mca_allocator_bucket.so	889.22KB	113	
⊞ mca_btl_openib.so	258.60KB	1046	
⊞ mca_bml_r2.so	8.80KB	34	
⊞ mca_pml_ob1.so	8.70KB	4	

Unfold the latest as well as another processors to compare the memory. Modules mathelp and histcom are using most of the memory. They belong to IOIPSL library. It manages input/output.

Again, a more detailed screenshot:



Heap Status Source Report

Data Source: ◆ Allocations ◆ Deallocations ◆ Hoard ◆ Red Zones

Options: □ Detect Leaks □ Relative to Baseline □ Enable Filtering

Process	Bytes △	Count	Begin Address	End Address	Backtrace ID	Allocator	Owner	Flags
Process 33: srun<orchidee_ol>.31	1527.20MB	24734						
orchidee_ol	1519.85MB	17244						
mathelp.f90	1486.70MB	1582						
for_allocate	1285.69MB	656						
Line 3154	1285.69MB	656						
Block 13958.1	17161920	1	0x2aeaf0dde050	0x2aeaf1e3bf0f	13958	C	C	none
Block 13949.1	17161920	1	0x2aeaefd80050	0x2aeaf0dddf0f	13949	C	C	none
Block 13976.1	16992000	1	0x2aeaf2e71050	0x2aeaf3ea574f	13976	C	C	none
Block 13967.1	16992000	1	0x2aeaf1e3c050	0x2aeaf2e7074f	13967	C	C	none
Block 15646.1	5437440	1	0x2aeb1d550050	0x2aeb1da7f84f	15646	C	C	none
Block 15637.1	5437440	1	0x2aeb1d010050	0x2aeb1d53f84f	15637	C	C	none
Block 15628.1	5437440	1	0x2aeb1cae0050	0x2aeb1d00f84f	15628	C	C	none
Block 15589.13	5437440	1	0x2aeb1b8fe050	0x2aeb1be2d84f	15589	C	C	none
Block 15589.12	5437440	1	0x2aeb1b3ce050	0x2aeb1b8fd84f	15589	C	C	none
Block 15589.11	5437440	1	0x2aeb1ae9e050	0x2aeb1b3cd84f	15589	C	C	none
Block 15589.10	5437440	1	0x2aeb1a96e050	0x2aeb1ae9d84f	15589	C	C	none
Block 15589.9	5437440	1	0x2aeb1a43e050	0x2aeb1a96d84f	15589	C	C	none
Block 15589.8	5437440	1	0x2aeb19f0e050	0x2aeb1a43d84f	15589	C	C	none
Block 15589.7	5437440	1	0x2aeb199de050	0x2aeb19f0d84f	15589	C	C	none
Block 15589.6	5437440	1	0x2aeb194ae050	0x2aeb199dd84f	15589	C	C	none
Block 15589.5	5437440	1	0x2aeb18f7e050	0x2aeb194ad84f	15589	C	C	none
Block 15589.4	5437440	1	0x2aeb18a4e050	0x2aeb18f7d84f	15589	C	C	none
Block 15589.3	5437440	1	0x2aeb1851e050	0x2aeb18a4d84f	15589	C	C	none
Block 15589.2	5437440	1	0x2aeb17fee050	0x2aeb1851d84f	15589	C	C	none
Block 15589.1	5437440	1	0x2aeb17abe050	0x2aeb17fed84f	15589	C	C	none
Block 15574.13	5437440	1	0x2aeb1758e050	0x2aeb17abd84f	15574	C	C	none
Block 15574.12	5437440	1	0x2aeb1705e050	0x2aeb1758d84f	15574	C	C	none
Block 15574.11	5437440	1	0x2aeb16b2e050	0x2aeb1705d84f	15574	C	C	none
Block 15574.10	5437440	1	0x2aeb165fe050	0x2aeb16b2d84f	15574	C	C	none
Block 15574.9	5437440	1	0x2aeb160ce050	0x2aeb165fd84f	15574	C	C	none
Block 15574.8	5437440	1	0x2aeb15b9e050	0x2aeb160cd84f	15574	C	C	none
Block 15574.7	5437440	1	0x2aeb1566e050	0x2aeb15b9d84f	15574	C	C	none
Block 15574.6	5437440	1	0x2aeb1513e050	0x2aeb1566d84f	15574	C	C	none

Backtrace

ID ▽	Function	Line #	Source Information
13958			
	malloc	172	malloc_wrappers_dlopen.c
	for_allocate	**3154**	**mathelp.f90**
	histdef	1208	histcom.f90
	ioipslctrl_histstom	3132	ioipslctrl.f90
	ioipslctrl_history	2150	ioipslctrl.f90
	intersurf_initialize_2c	263	intersurf.f90
	driver	1263	dim2_driver.f90
	main		orchidee_ol
	__libc_start_main		libc.so.6
	_start		orchidee_ol

Source

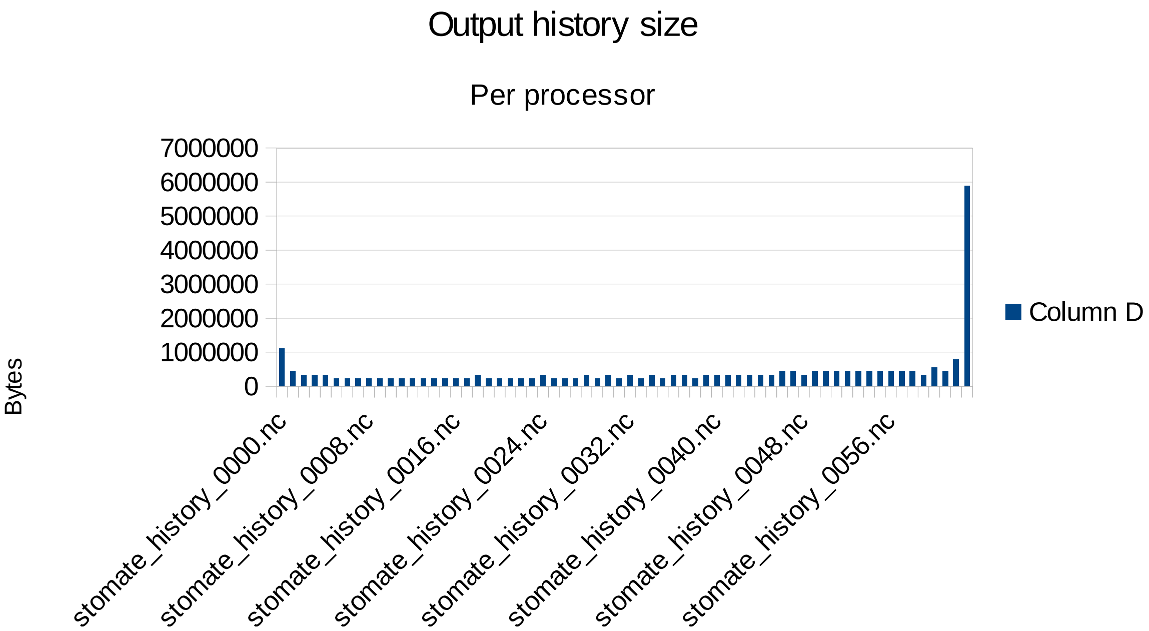
/ccc/work/cont003/dsm/p529jorn/modipsl/modeles/MICT/.config/ppsrc/sechiba/ioipslctrl.f90

```
3129          &                 101,1,101, hist_pool_101axis_id,32, ave(5), dt, hist_dt)
3130
3131     ! 100 year wood product pool
3132     CALL histdef (hist_id_stom, &
3133          &                 TRIM("PROD100_HAR         "), &
3134          &                 TRIM("100 year wood product pool                          ")
3135          &                 TRIM("gC/m**2             "), iim,jjm, hist_hori_id, &
3136          &                 101,1,101, hist_pool_101axis_id,32, ave(5), dt, hist_dt)
3137
3138     ! annual flux for each 100 year wood product pool
3139     CALL histdef (hist_id_stom, &
3140          &                 TRIM("FLUX100_LCC         "), &
```

Histdef is responsible of the memory allocation.

Output File Size

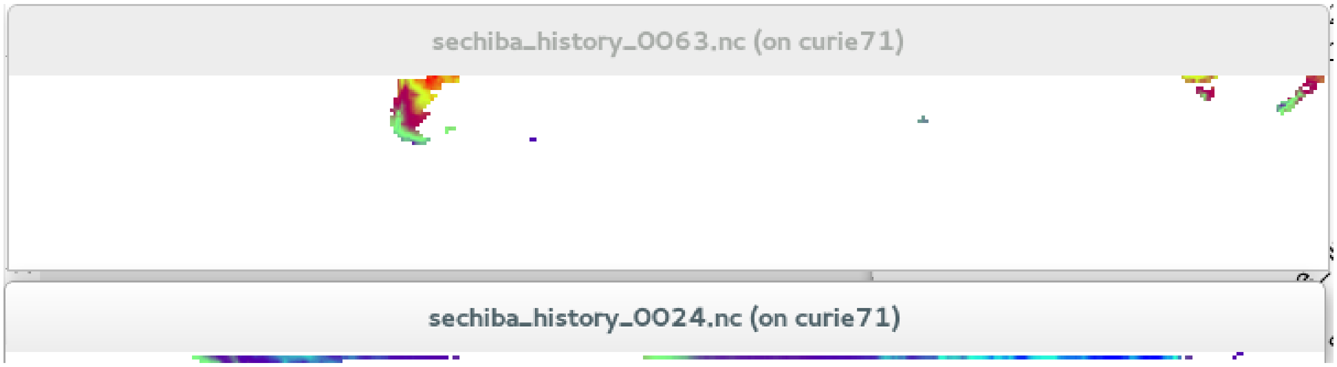
Stomate output file history vs its size.



The latest processor is several times bigger the others.

Ncview

Using ncview it shows how big is the area managed by the latest proc. Let's compare it with the latest (63) and another random processor output.


sechiba_history_0063.nc (on curie71)
sechiba_history_0024.nc (on curie71)

Latest proc (63) has much more data than the other (e.g 24).

Conclusion

Orchidee selects for each processor the same number of land pixels. But the outputs have different sizes because they represent different world areas (managed by IOIPSL). A direct consequence of is an increase time writing data to netcdf file. The rest of the processors must wait for the latest due to MPI calls.

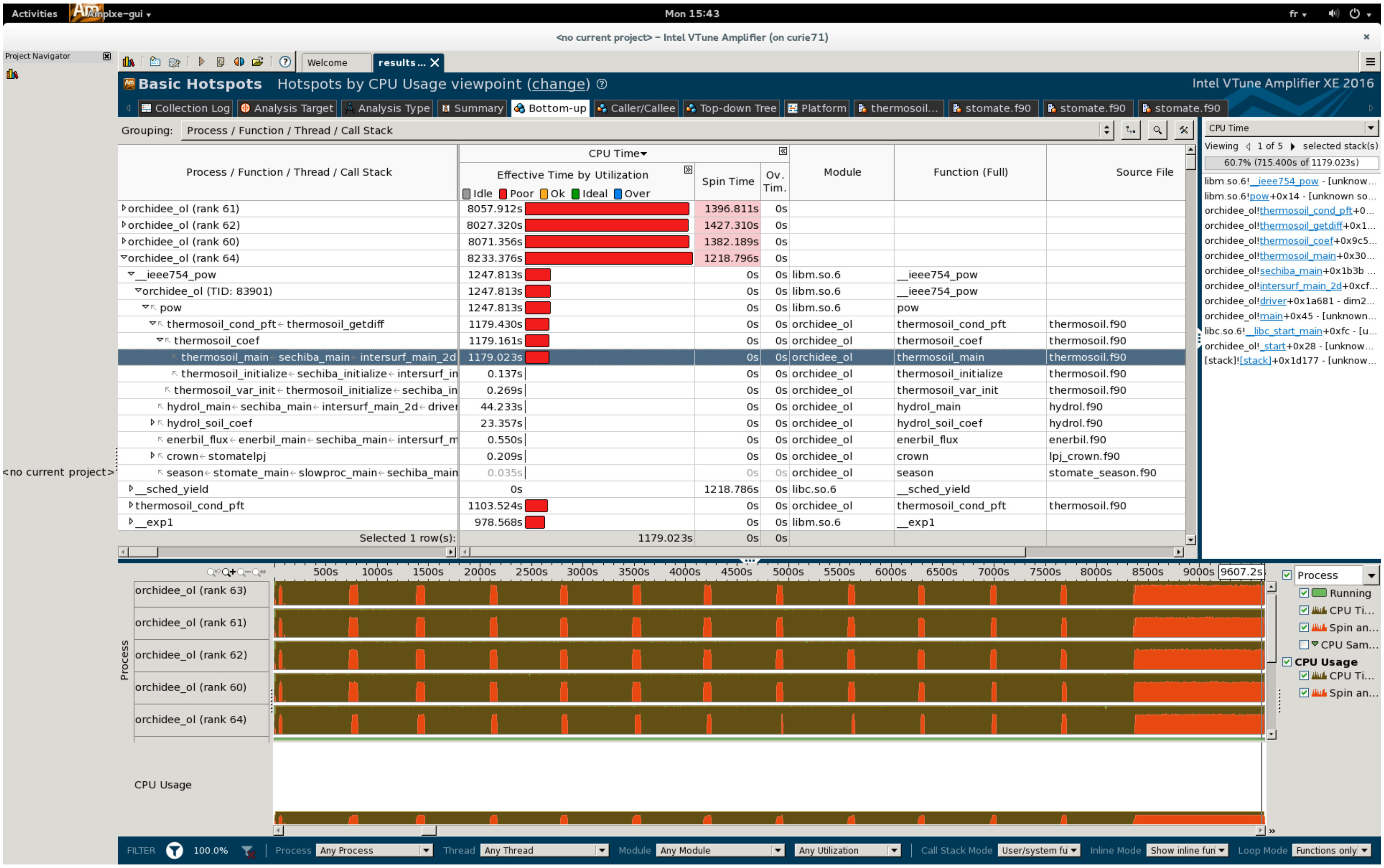
XIOS 1 Profiling

To deactivate IOIPSL history it is necessary to change in run.def:

- XIOS_ORCHIDEE_OK=y
- WRITE_STEP=0
- STOMATE_HIST_DT=0

Simulation:

- 64 cores (+ XIOS1 1core),
- 1Y
- 05DEG
- Simulation time: 2h40 (9600 seconds)
- yearly output
- /ccc/work/cont003/dsm/p529jorn/experiments/SECHSTOM.DGVM.10336_vtune_xios



Basic Hotspots Hotspots by CPU Usage viewpoint (change)

Grouping: Process / Function / Thread / Call Stack

Process / Function / Thread / Call Stack	CPU Time: Effective Time by Utilization (■ Idle ■ Poor ■ Ok ■ Ideal ■ Over)	CPU Time: Spin Time	CPU Time: Ov. Tim.	Module	Function (Full)	Source File
orchidee_ol (rank 61)	8057.912s	1396.811s	0s			
orchidee_ol (rank 62)	8027.320s	1427.310s	0s			
orchidee_ol (rank 60)	8071.356s	1382.189s	0s			
orchidee_ol (rank 64)	8233.376s	1218.796s	0s			
__ieee754_pow	1247.813s	0s	0s	libm.so.6	__ieee754_pow	
orchidee_ol (TID: 83901)	1247.813s	0s	0s	libm.so.6	__ieee754_pow	
pow	1247.813s	0s	0s	libm.so.6	pow	
thermosoil_cond_pft ← thermosoil_getdiff	1179.430s	0s	0s	orchidee_ol	thermosoil_cond_pft	thermosoil.f90
thermosoil_coef	1179.161s	0s	0s	orchidee_ol	thermosoil_coef	thermosoil.f90
thermosoil_main ← sechiba_main ← intersurf_main_2d	1179.023s	0s	0s	orchidee_ol	thermosoil_main	thermosoil.f90
thermosoil_initialize ← sechiba_initialize ← intersurf_in	0.137s	0s	0s	orchidee_ol	thermosoil_initialize	thermosoil.f90
thermosoil_var_init ← thermosoil_initialize ← sechiba_in	0.269s	0s	0s	orchidee_ol	thermosoil_var_init	thermosoil.f90
hydrol_main ← sechiba_main ← intersurf_main_2d ← driver	44.233s	0s	0s	orchidee_ol	hydrol_main	hydrol.f90
hydrol_soil_coef	23.357s	0s	0s	orchidee_ol	hydrol_soil_coef	hydrol.f90
enerbil_flux ← enerbil_main ← sechiba_main ← intersurf_m	0.550s	0s	0s	orchidee_ol	enerbil_flux	enerbil.f90
crown ← stomatelpj	0.209s	0s	0s	orchidee_ol	crown	lpj_crown.f90
season ← stomate_main ← slowproc_main ← sechiba_main	0.035s	0s	0s	orchidee_ol	season	stomate_season.f90
__sched_yield	0s	1218.786s	0s	libc.so.6	__sched_yield	
thermosoil_cond_pft	1103.524s	0s	0s	orchidee_ol	thermosoil_cond_pft	thermosoil.f90
__exp1	978.568s	0s	0s	libm.so.6	__exp1	
Selected 1 row(s):	1179.023s	0s	0s			

CPU Time

Viewing 1 of 5 selected stack(s)

60.7% (715.400s of 1179.023s)

```
libm.so.6!__ieee754_pow - [unknow...
libm.so.6!pow+0x14 - [unknown so...
orchidee_ol!thermosoil_cond_pft+0...
orchidee_ol!thermosoil_getdiff+0x1...
orchidee_ol!thermosoil_coef+0x9c5...
orchidee_ol!thermosoil_main+0x30...
orchidee_ol!sechiba_main+0x1b3b ...
orchidee_ol!intersurf_main_2d+0xcf...
orchidee_ol!driver+0x1a681 - dim2...
orchidee_ol!main+0x45 - [unknown...
libc.so.6!__libc_start_main+0xfc - [u...
orchidee_ol!_start+0x28 - [unknow...
[stack]![stack]+0x1d177 - [unknow...
```

FILTER 100.0% Process: Any Process · Thread: Any Thread · Module: Any Module · Any Utilization · Call Stack Mode: User/system fu · Inline Mode: Show inline fun · Loop Mode: Functions only

MICT vs CROP

Overview
Compare MICT vs MICT merged CROP disabled. Time performance and its outputs.

Simulation Description
1Year
2 Degree

MICT
/ccc/work/cont003/dsm/p529jorn/experiments/SECHSTOM.DGVM.10336_interpol_improve

```
JobID         JobName           Ntasks  Ncpus Nnodes     Layout      Elapsed   Ratio     CPusage    Eff
State
------------   ------------       ------  ----- ------     -------      -------   -----      -------    ---  -----
4996984       M7_test                -     32      2          -     00:12:55     100          -      -  -
4996984.0     orchidee_ol            32     32      2     BBlock    00:12:54    99.8    00:12:39   98.0
COMPLETED
774s
```

MICT-CROP
/ccc/work/cont003/dsm/p529jorn/experiments/SECHSTOM.DGVM.10336

```
JobID         JobName           Ntasks  Ncpus Nnodes     Layout      Elapsed   Ratio     CPusage    Eff
State
------------   ------------       ------  ----- ------     -------      -------   -----      -------    ---  -----
5001560       M65_test               -     32      2          -     00:16:39     100          -      -  -
5001560.0     orchidee_ol            32     32      2     BBlock    00:16:38    99.8    00:16:22   98.3
COMPLETED
998s
```

20% slower