I/O Monitoring with ECMWF’s 4D-Var

John Hague. IBM Consultant, Nov 2010

Overview

- The data analysis program 4D-Var is a key part of ECMWF's forecast suite
  - T1279 runs out of scalability on the P6 at about 60 nodes

- This is due to
  - Communication
  - Load imbalance and jitter
  - Serial components
  - I/O

- Purpose of this investigation is to look at the I/O to
  - Determine extent of I/O effect
  - Identify major bottlenecks
  - Look for possible improvements

- In order to get a better handle on the I/O
  - Implement technique (JIO) for profiling most C and Fortran I/O accesses
  - Accumulate and analyse statistics

JIO - Monitoring Details

- JIO Enables interception of Application I/O Calls
  - Intercepts (fread, fwrite, open, close, read, write) from C and Fortran
  - Writes trace file, collects and analyses data depending on environment variables
  - Gets address of above routines in memory with dlsym
  - Passes control to routines in memory

- Three levels of detail are provided depending on the setting of an environment variable

  1. JIO_ENV=JIO_SUMMARY
     - produces a summary table (one line for each of the above calls)

  2. JIO_ENV=JIO_DETAIL
     - produces a detailed table as well as summary table (multiple lines for each file)

  3. JIO_ENV=JIO_TRACE
     - produces one line for each I/O request as well as the tables in 1 and 2.
     - Timestamp recorded to enable post processing analysis

  - output produced for every mpi task

JIO – Setup Details

1. Create a shared object from jio.c and put in library libj.a
   - xlc -G -DRS6K jio.c
   - ar -rv libj.a shr.o

2. At run time, pre load shared object
   - JJJ="$PWD/libj.a(shr.o)"
   - export LDR_PRELOAD64=”$JJJ”

3. In the driver program (e.g. master.F90), add:
   - At start, call mpi_barrier, irtc, and write timestamp
     - To synchronise date stamp for final post processing
   - at end add: open(99,file=”JIO_END”)
     - to enable the shared object routine to print out detailed and summary statistics

4. and export environment variables as required
   - export JIO_ENV=JIO_TRACE # or JIO_DETAIL or JIO_SUMMARY

4D-Var

- 4D-Var used for study was Cycle 36R4 T1279 Early Delivery
  - Elapsed time is critical

- Run on 48 nodes (192 tasks) and 96 nodes (384 tasks)
  - 16 threads/task, 4 tasks per node, uses SMT
  - Sequence of steps (jobs) is
    - **traj0, min0, traj1, min1, traj2 (final trajectory)**

- Uses GPFS filesystem
  - 60 1.1TB RAID6 arrays
  - 4MB Block size

- Many types of I/O
  - Task 0 reads/writes sequential files (and distributes/collects to/from other tasks)
  - Many tasks read/write some (different) sequential files
  - Many tasks read and write observational database – which is 100s of files
    - **First traj reads and write full data base (ECMA), and writes reduced database (CCMA)**
    - **Other steps read and write CCMA**
    - **Final traj also reads and writes ECMA**

Summary output for each task for each step

- Summary for traj0, task 0 on 48 nodes (192 tasks)

Routine	Calls	MB	msec	MB/s
fread	109146	4425	8474	525
fwrite	65924	48	123	394
open	533	0	2461	0
close	553	0	1110	0
read	536	886	1597	555
write	19296	140	1365	103
TOTAL	195988	5528	15133	365

Detailed output

- Snippets for for step 0, task 0 & 1, 48 nodes

```
0:JIO Detail File=ICMSHfeccINIT
0:JIO Detail Routine  Calls       MB      MSEC     MB/s
0:JIO Detail fread:    9049  860.518  1310.282  656.742
0:JIO Detail open:        1    0.000    20.600    0.000
0:JIO Detail close:       1    0.000     0.081    0.000

1:JIO Detail File=TRAJHR00/trajgridL002
1:JIO Detail Routine  Calls       MB      MSEC     MB/s
1:JIO Detail fwrite:     12    5.407   158.062   34.209
1:JIO Detail open:        1    0.000     5.063    0.000
1:JIO Detail close:       1    0.000    45.033    0.000
```

- **ICMSHfeccINIT read only by task 0**
- **trajgridL00<N> written by many tasks**

Trace output

- Snippets for traj0, task 0: 48 nodes

```
open:  file=61               time=19231971nsec stamp=1287765459127298960
                     /fws2/lb/work/rd/das/fecc/2009122100/an/poolmask.CCMA
write: file=61 bytes=4194304 time= 1797322nsec stamp=1287765459390545062
write: file=61 bytes=3253464 time= 1011463nsec stamp=1287765459498535986
close: file=61               time=28673129nsec stamp=1287765459499564990
. . .
. . .
. . .
open:  file=55,              time= 3444983nsec stamp=1287765176910913632
                                                             ICMSHfeccINIT
fread: file=55 bytes=      1 time=47847881nsec stamp=1287765176915349771
fread: file=55 bytes=      1 time=     451nsec stamp=1287765176963215457
fread: file=55 bytes=      1 time=     326nsec stamp=1287765176963220121
fread: file=55 bytes=      1 time=     314nsec stamp=1287765176963224875
fread: file=55 bytes=      3 time=     463nsec stamp=1287765176963427015
fread: file=55 bytes=      1 time=     353nsec stamp=1287765176963438248
fread: file=55 bytes=3281126 time= 1201922nsec stamp=1287765176963443943
```

- Note that many file accesses involve only one byte, but that the time required is only a fraction of a microsecond

Summary Statistics – post processed

- Summary for all steps for task 0 for 48 nodes (192 tasks)

		Task 0 I/O			
Step	Wall	Calls	GB	sec	files
traj0	309	196K	5.5	15	533
min0	359	372K	3.6	9	290
traj1	163	84K	6.7	12	302
min1	312	385K	6.5	21	280
Traj2	322	242K	9.1	39	603
TOTAL	1465	1279K	31.4	96	
%WALL				7%	

- Wall is total elapsed time

Summary Statistics – post processed

- Summary for all steps for task 0 and all tasks for 48 nodes (192 tasks)

		Task 0 I/O				All tasks I/O			
Step	Wall	Calls	GB	sec	files	Calls	GB	sec	Files
traj0	309	196K	5.5	15	533	3184K	29	397	1852
min0	359	372K	3.6	9	290	922K	22	303	977
traj1	163	84K	6.7	12	302	448K	16	650	779
min1	312	385K	6.5	21	280	820K	58	441	811
Traj2	322	242K	9.1	39	603	750K	40	497	1319
TOTAL	1465	1279K	31.4	96		6124K	165	2288	
%WALL				7%				157%	

- Wall is total elapsed time

Summary Statistics – post processed

- Summary for all steps for task 0 and all tasks for 48 nodes (192 tasks)

		Task 0 I/O				All tasks I/O			
Step	Wall	Calls	GB	sec	files	Calls	GB	sec	files
traj0	309	196K	5.5	15	533	3184K	29	397	1852
min0	359	372K	3.6	9	290	922K	22	303	977
traj1	163	84K	6.7	12	302	448K	16	650	779
min1	312	385K	6.5	21	280	820K	58	441	811
Traj2	322	242K	9.1	39	603	750K	40	497	1319
TOTAL	1465	1279K	31.4	96		6124K	165	2288	
%WALL				7%				157%	

- Wall is total elapsed time
- Note large number of files

Summary Statistics – post processed

- Summary for all steps for task 0 and all tasks for 48 nodes (192 tasks)

		Task 0 I/O				All tasks I/O			
Step	**Wall**	**Calls**	**GB**	**sec**	**files**	**Calls**	**GB**	**sec**	**Files**
traj0	309	196K	5.5	15	533	3184K	29	397	1852
min0	359	372K	3.6	9	290	922K	22	303	977
traj1	163	84K	6.7	12	302	448K	16	650	779
min1	312	385K	6.5	21	280	820K	58	441	811
Traj2	322	242K	9.1	39	603	750K	40	497	1319
TOTAL	1465	1279K	31.4	96		6124K	165	2288	
%WALL				7%				157%	

- Time for task 0 is 7% of Wallclock time
- Total time for all tasks is longer than Wallclock time
  - Need to get time when at least one task is “active” with I/O
  - If other tasks have to wait for the task(s) performing I/O, then this is the relevant I/O time
  - Can get “active time by processing Trace output

Summary Statistics – post processed

- Summary for all steps for task 0 and all tasks for 48 nodes (192 tasks)

		Task 0 I/O				All tasks I/O				Active
Step	**Wall**	**Calls**	**GB**	**sec**	**files**	**Calls**	**GB**	**sec**	**Files**	**sec**
traj0	309	196K	5.5	15	533	3184K	29	397	1852	47
min0	359	372K	3.6	9	290	922K	22	303	977	20
traj1	163	84K	6.7	12	302	448K	16	650	779	24
min1	312	385K	6.5	21	280	820K	58	441	811	33
traj2	322	242K	9.1	39	603	750K	40	497	1319	80
TOTAL	1465	1279K	31.4	96		6124K	165	2288		204
%WALL				7%				157%		14%

- Active time is 14% of Wallclock time
  - Active means at least one task performing I/O
- Largest amount of active time is in traj2 (final trajectory)

Summary Statistics – post processed

- Summary for all steps for task 0 and all tasks for 96 nodes (384 tasks)

		Task 0				All tasks				Active
Step	Wall	Calls	GB	sec	files	Calls	GB	sec	files	sec
traj0	367	301K	5.5	19	532	6091K	36	846	2335	40
min0	334	387K	3.6	9	290	1392K	25	1050	1204	22
traj1	158	99K	6.7	16	302	825K	20	2219	1199	33
min1	301	400K	6.6	29	280	1176K	61	1665	1023	47
Traj2	436	407K	9.1	48	602	1421K	47	1281	2021	100
TOTAL	1596	1594K	31.5	122		10905K	189	7061		242
%WALL				8%				442%		15%

“Active” means at least one task performing I/O

Summary Statistics – post processed

- Summary for all steps for task 0 and all tasks for 48 nodes (192 tasks)

		Task 0				All tasks				Active
Step	Wall	Calls	GB	sec	files	Calls	GB	sec	files	Sec
TOTAL	1465	1279K	31.4	96		6124K	165	2288		204
%WALL				7%				157%		14%

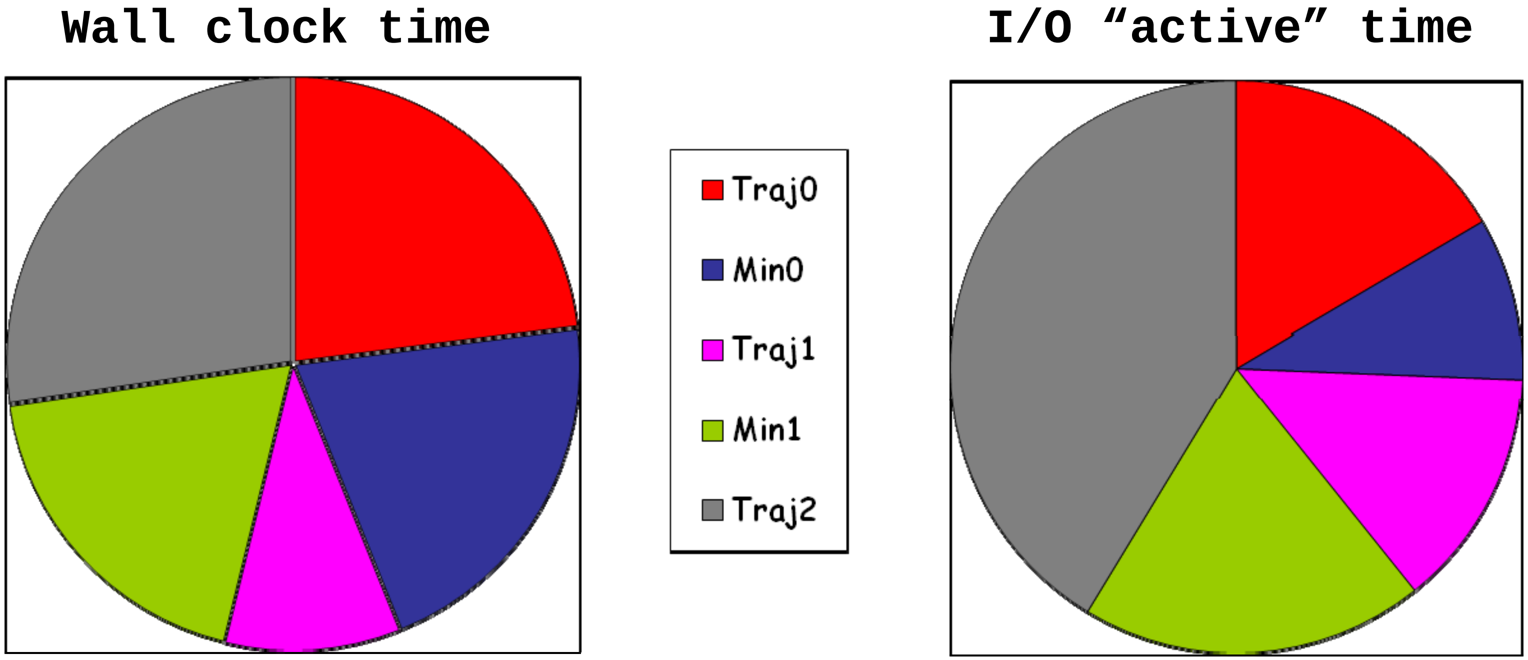
- Summary for all steps for task 0 and all tasks for 96 nodes (384 tasks)

		Task 0				All tasks				Active
Step	Wall	Calls	GB	sec	files	Calls	GB	sec	files	sec
TOTAL	1596	1594K	31.5	122		10905K	189	7061		242
%WALL				8%				442%		15%

- 96 nodes have 38 sec more “active” I/O time than 48 nodes
- “active” means at least one task performing I/O

Summary Statistics – post processed

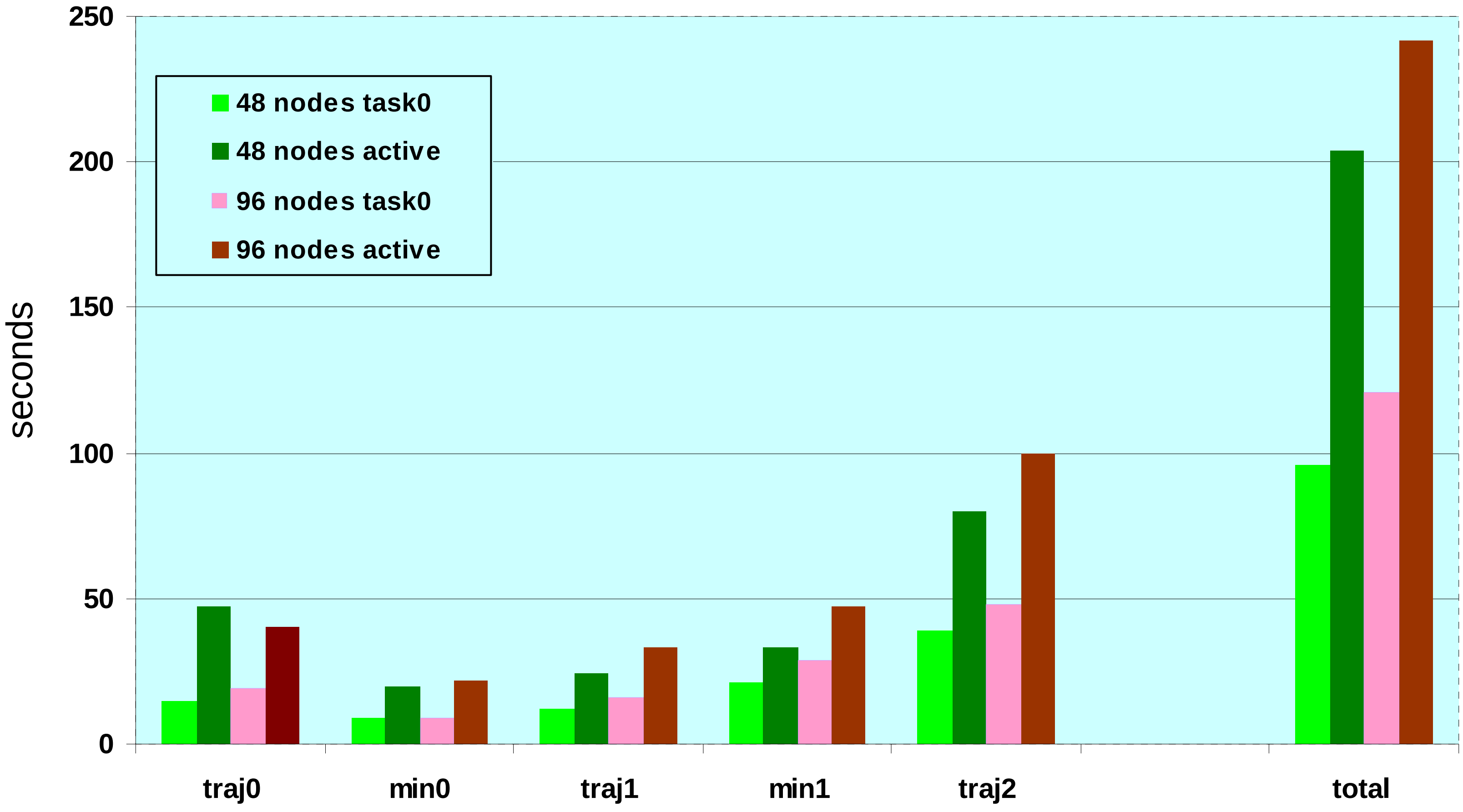
- Summary for all steps for all tasks for 96 nodes



- Final trajectory I/O time is dominant

Summary Statistics – post processed

- Summary: I/O for all steps for all tasks for 48 and 96 nodes



Detailed statistics – post processed

- Statistics for 48 nodes
- Times (> 500msec) per file for traj2 for task 0

```
MSEC   MB RATE CALLS File
2331 2417 1036 26793 ICMGGfeccINIUA
1536  117   76    23 /fws2/lb/fdb/:rd:oper:g:fecc:20091220::/:0:0:0:4v:2100::::::::
1430    0    0     3 /fws2/lb/work/rd/das/fecc/2009122100/an/ECMA/ECMA.iomap
1351  399  295  7778 specwavein
1196  743  621  7068 ICMGGfeccBGHRUA
 991  860  868  9051 ICMSHfeccBGHR
 959    0    0     3 /fws2/lb/work/rd/das/fecc/2009122100/an/ECMA.iasi/ECMA.iomap
 902    0    0     2 /usr/local/lib/metaps/lib/grib_api/1.9.3/share/definitions/grib1/section.1.def
 844    0    0     4 /usr/local/lib/metaps/lib/grib_api/1.9.3/share/definitions/grib1/0.ecmf.table
 835  860 1030  9051 ICMSHfeccINIT
 799    0    0   772 /fws2/lb/work/rd/das/fecc/2009122100/an/ECMA/../ECMA.airs/1/radiance
 750    0    0     3 /fws2/lb/work/rd/das/fecc/2009122100/an/ECMA.hirs/ECMA.iomap
 726    0    0     3 /fws2/lb/work/rd/das/fecc/2009122100/an/ECMA.windsat/ECMA.iomap
 683    0    0     3 /fws2/lb/work/rd/das/fecc/2009122100/an/ECMA.amsre/ECMA.iomap
 611    0    0  2779 /fws2/lb/work/rd/das/fecc/2009122100/an/ECMA.satob/ECMA.iomap
 581  563  968   143 wam_subgrid_2
 579    0    0     2 /usr/local/lib/metaps/lib/grib_api/1.9.3/share/definitions/grib1/boot.def
 545    0    0  2971 /fws2/lb/work/rd/das/fecc/2009122100/an/ECMA.scatt/ECMA.iomap
 531   31   60   378 /fws2/lb/work/rd/das/fecc/2009122100/an/ECMA/../ECMA.satob/1/hdr
 517    0    0   388 /fws2/lb/work/rd/das/fecc/2009122100/an/ECMA/../ECMA.conv/1/timeslot_index
```

- There are a very large number of data base (ECMA) accesses
- Several ECMA accesses involve only very few bytes exchange

Detailed statistics – post processed

- Statistics for 96 nodes
- Top 20 times for files for traj2 for task 0

```
MSEC   MB RATE CALLS File
 2288 2417 1056 26793 ICMGGfeedINIUA
 1739    0    0     3 /fws2/lb/work/rd/das/feed/2009122100/an/ECMA.airs/ECMA.iomap
 1586  743  468  7068 ICMGGfeedBGHRUA
 1550    4    2   772 /fws2/lb/work/rd/das/feed/2009122100/an/ECMA/../ECMA.reo3/1/reo3
 1466    0    0     3 /fws2/lb/work/rd/das/feed/2009122100/an/ECMA.satob/ECMA.iomap
 1401    6    4     4 VARBC.cycle
 1370    0    0   772 /fws2/lb/work/rd/das/feed/2009122100/an/ECMA/../ECMA.mhs/1/index
 1301   31   24   772 /fws2/lb/work/rd/das/feed/2009122100/an/ECMA/../ECMA.satob/1/satob
 1249    0    0     3 /fws2/lb/work/rd/das/feed/2009122100/an/ECMA.gpsro/ECMA.iomap
 1163  860  739  9051 ICMSHfeedINIT
 1154  399  346  7778 specwavein
  997    0    0     2 /usr/local/lib/metaps/lib/grib_api/1.9.3/share/definitions/grib1/boot.def
  980  860  877  9051 ICMSHfeedBGHR
  945    0    0   772 /fws2/lb/work/rd/das/feed/2009122100/an/ECMA/../ECMA.amsub/1/surfemiss
  934    0    0     3 /fws2/lb/work/rd/das/feed/2009122100/an/ECMA.reo3/ECMA.iomap
  923    5    6   772 /fws2/lb/work/rd/das/feed/2009122100/an/ECMA/../ECMA.mhs/1/surfemiss_body
  901  563  625   143 wam_subgrid_2
  843    0    0     3 /fws2/lb/work/rd/das/feed/2009122100/an/ECMA/ECMA.iomap
  784    0    0     9 /usr/local/lib/metaps/lib/grib_api/1.9.3/ifs_samples/grib1/gg_sfc.tmpl
  681    0    0     3 ers_sp.cor
```

- Note that ECMA accesses are longer than for 48 nodes

Conclusions

- I/O is major inhibitor to scalability

- No smoking gun
  - Several smouldering ashes

- Check if ECMA minimal byte accesses can be improved

- GPFS seems to handle transfer of 189GB with10.9M calls across > 2000 files quite well
  - But check whether GPFS filesystem enhancement can help

- Run one binary across all steps
  - Eliminates saving and restoring data
  - An objective of OOPS
  - Still need to write restart files unless not needed because 4D-Var so fast !

Data transfers

- Some major data transfers between steps
  - “0”, “sev”(=several), and “all” indicate task participation

	traj0	min0	traj1	min1	traj2
ICM...INIT	--> 0	--> 0	--> 0	--> 0	--> 0
ICM...INIUA	--> 0		--> 0		--> 0
ICM...BGHR			--> 0		--> 0
ICM...BGHRUA			--> 0		--> 0
specwavein	--> 0		--> 0		--> 0
wavelet.cv		--> 0		--> 0	
traj00		0 -->		--> 0	
traj01				0 -->	
ME...<N>		sev-->		->sev->	
trajspecL<N>	all-->	-->all	all-->	-->all	
trajgridL<N>	all-->	-->all	all-->	-->all	
ECMA	->sev->				->all->
CCMA	sev-->	->sev->	->ev->	->sev->	->sev->

- Initialisation files are different for each step
  - Previous step writes to fdb, and script re-links before next step