ParforEngine>ParforEngine.getCompleteIntervals (Calls: 459, Time: 800.996 s)

Generated 22-May-2024 23:14:57 using performance time.
Class method in file C:\Program Files\MATLAB\R2023b\toolbox\parallel\cluster\+parallel\+internal\+parfor\ParforEngine.m
Copy to new window for comparing multiple runs

Parents (calling functions)

Function Name	Function Type	Calls
parallel_function>distributed_execution	Subfunction	459

Lines that take the most time

Line Number	Code	Calls	Total Time (s)	% Time	Time Plot
214	`r = obj.ParforController.waitForNextCompletedInterval…`	2235	800.314	99.9%	
239	`data = parallel.internal.pool.optionallyDeserialize(r…`	1632	0.173	0.0%	
219	`if r.isEmpty()`	2235	0.109	0.0%	
222	`if ~obj.isSessionValidAndRunning()`	603	0.070	0.0%	
218	`parallel.internal.pool.yield();`	2235	0.056	0.0%	
All other lines			0.274	0.0%	
Totals			800.996	100%	

Children (called functions)

Function Name	Function Type	Calls	Total Time (s)	% Time	Time Plot
optionallyDeserialize	Function	1632	0.123	0.0%	
ParforEngine>ParforEngine.isSessionValidAndRunning	Class method	603	0.062	0.0%	
Self time (built-ins, overhead, etc.)			800.811	100%	
Totals			800.996	100%	

Code Analyzer results

No Code Analyzer messages.

Coverage results

Show coverage for parent folder

Total lines in function	47
Non-code lines (comments, blank lines)	14
Code lines (lines that can run)	33
Code lines that did run	29
Code lines that did not run	4
Coverage (did run/can run)	87.88 %

Function listing

Time	Calls	Line	
		204	`function [tags, results] = getCompleteIntervals(obj, numIntervals)`
		205	
0.001	459	206	`    tags = nan(numIntervals, 1);`
0.001	459	207	`    results = cell(numIntervals, 2);`
< 0.001	459	208	`    for i = 1:numIntervals`
< 0.001	1632	209	`        err = [];`
0.055	1632	210	`        r = parallel.internal.pool.ParforIntervalResult();`
0.023	1632	211	`        while r.isEmpty()`
0.002	2235	212	`            assert(obj.NumIntervalsInController > 0, ...`
		213	`                'Internal error in PARFOR - no intervals to retrieve.');`
800.314	2235	214	`            r = obj.ParforController.waitForNextCompletedInterval(obj.AwaitIntervalPollPeriod);`
		215	
		216	`            % In each poll tick of the polling wait, yield to any`

```
                    % events allowed to interrupt the parallel language.
0.056   2235  218       parallel.internal.pool.yield();
0.109   2235  219       if r.isEmpty()
                          % Only test to see if the session is failing if we didn't get a
                          % results from the queue
0.070    603  222           if ~obj.isSessionValidAndRunning()
                              errorMessageInput = iGetParpoolLinkForError();
                              error(message('parallel:lang:parfor:SessionShutDown', errorMessageInput));
< 0.001  603  225           end
< 0.001 1632  226       else
0.036   1632  227           obj.NumIntervalsInController = obj.NumIntervalsInController - 1;
0.028   1632  228           if r.hasError()
                              % Maybe try again
                              [r, err] = obj.handleIntervalErrorResult(r);
< 0.001 1632  231           end
< 0.001 2235  232       end
0.027   2235  233   end
                    % Check to see if the interval result has an error
0.017   1632  235   if r.hasError()
                        throw(err);
< 0.001 1632  237   else
0.027   1632  238       tags(i) = r.getTag();
0.173   1632  239       data = parallel.internal.pool.optionallyDeserialize(r.getResult());
0.003   1632  240       assert(numel(data) == 2, ...
                            'Unexpectedly received the incorrect number of outputs.');
0.015   1632  242       results(i,:) = data;
0.026   1632  243       obj.IncompleteIntervalsCell{tags(i)} = [];
< 0.001 1632  244   end
0.001   1632  245 end
                % Yield one final time to ensure all events directly generated by
                % code in the PARFOR loop are executed before we give control back
                % to the user.
0.003    459  249 parallel.internal.pool.yield();
0.004    459  250 end
```

Local functions in this file are not included in this listing.