

UC Davis

UC Davis Previously Published Works

Title

Performance assessment of a software-based coincidence processor for the EXPLORER total-body PET scanner

Permalink

<https://escholarship.org/uc/item/19g4m4qt>

Journal

Physics in Medicine and Biology, 63(18)

ISSN

0031-9155

Authors

Leung, Edwin K
Judenhofer, Martin S
Cherry, Simon R
[et al.](#)

Publication Date

2018-09-01

DOI

10.1088/1361-6560/aadd3c

Peer reviewed



Published in final edited form as:

Phys Med Biol. ; 63(18): 18NT01. doi:10.1088/1361-6560/aadd3c.

Performance Assessment of a Software-Based Coincidence Processor for the EXPLORER Total-Body PET Scanner

Edwin K. Leung^{1,2}, Martin S. Judenhofer¹, Simon R. Cherry^{1,2}, and Ramsey D Badawi^{1,2}

¹Department of Biomedical Engineering, University of California, Davis, One Shields Avenue, Davis, CA 95616, United States

²Department of Radiology, University of California, Davis Medical Center, 2315, Stockton Blvd., Sacramento, CA 95817, United States

Abstract

Coincidence processing in positron emission tomography (PET) is typically done during acquisition of the data. However, on the EXPLORER totalbody PET scanner we plan, in addition, to store unpaired single events (i.e. singles) for post-acquisition coincidence processing. A software-based coincidence processor was developed for EXPLORER and its performance was assessed. Our results showed that the performance of the coincidence processor could be significantly impacted by the type of data storage (Peripheral Component Interconnect Express (PCIe)-attached solid state drive (SSD) vs RAID 6 hard disk drives (HDDs)) especially when multiple data files were processed in parallel. We showed that a 48-thread computer node with dual Intel Xeon E5-2650 v4 central processing units (CPUs) and a PCIe SSD was sufficient to process approximately 120 M singles/s at an incoming singles rate of approximately 150 Mcps. With 2 computer nodes, near real-time coincidence processing became possible at this incoming singles rate.

Keywords

Monte Carlo; Multi-threading; Coincidence Processing; Positron Emission; Tomography

1. Introduction

Positron emission tomography (PET) relies on annihilation coincidence detection (ACD) to generate data for image reconstruction. In most commercial scanners, ACD is done during acquisition of the data in real-time. However, for the EXPLORER total-body PET scanner (which uses technology described by Lv et al. (2017)) we plan, in addition, to store unpaired single events (here we refer to them as "singles") for post-acquisition software-based coincidence processing. Several groups have implemented software-based coincidence processors in the past (Goldschmidt et al. 2013, Streun et al. 2006, McElroy et al. 2005). Their systems were tested to handle incoming singles rates between 3 – 12 Mcps (Streun et al. 2006, McElroy et al. 2005).

The axial field-of-view (FOV) of the EXPLORER scanner is ~10 times longer than most clinical scanners. Simulations have shown that the EXPLORER scanner can generate 6–7 times more singles data than shorter clinical scanners at their optimal activity concentrations where peak noise-equivalent count (NEC) rates occur (Poon et al. 2012). At such singles data rates, it is essential for the software-based coincidence processor to process singles at near real-time prior to image reconstruction in order for the post-acquisition coincidence processing method to be practical.

In this note, we report an assessment of the coincidence processing performance of a software-based coincidence processor we developed to process EXPLORER singles data at incoming singles rates of up to ~500 Mcps.

2. Methods

2.1. Data format and expected data rates

EXPLORER is expected to produce list-mode singles data at an incoming singles rate of ~150 Mcps with a 370-MBq injection uniformly distributed in an adult phantom, and a maximum incoming singles rate of ~500 Mcps (in ^{82}Rb myocardial perfusion imaging, for example) (Poon et al. 2012). To maximize data compactness, the singles will be stored in a 64-bit list-mode format. The list-mode data contains 2 different event types - individual 64-bit coarse time stamps and 64-bit single events (containing fine time stamps, energy and positional information).

A 30-s total-body scan at an incoming singles rate of 150 Mcps would lead to approximately 34 GiB (1 GiB = 2^{30} bytes) of list-mode singles data. Our goal was to determine the computational requirement for processing 150 M singles/s so that near real-time coincidence processing becomes possible at this incoming singles rate.

2.2. Software-based coincidence processor

Our coincidence processor was designed to process chronologically-ordered list-mode singles data files. The major parameters of the coincidence processor include 1) an energy window, 2) a fixed time window, 3) a coincidence policy and 4) a geometric window. An energy window was first applied to the data to discard any singles outside the energy window. Next, a fixed time window was applied to every energy-qualified single. This procedure is typically referred to as the multiple window (MW) method (Strydhorst and Buvat 2016, Moraes et al. 2015, Oliver et al. 2009). We used MW, takeAllGoods (i.e. accepting all valid coincidences) as our coincidence policy for this experiment. Finally, a geometric window was applied to the remaining coincidences. Some of the coincidences were discarded in this step if their lines-of-response (LORs) fell outside the transaxial FOV or the maximum crystal ring difference of the scanner. A visual example of the coincidence processor handling energy-qualified singles is shown in Figure 1. A pseudocode of the coincidence processor is also available in Algorithm 1 in the Appendix.

To further reduce the number of random coincidences due to the increased axial FOV, the coincidences were processed after the geometric window with a secondary, variable time

window, which relied on the axial positions of the 2 crystals forming the LOR (Poon et al. 2012). The variable time window was defined using the following equation:

$$\tau(R) = \frac{\sqrt{T^2 + (R \cdot W)^2}}{c} + 3 \cdot P$$

where R is the crystal ring difference, T is the transaxial FOV (in m), W is the axial crystal pitch (in m), P is the coincidence timing resolution (in s) and c is the speed of light (in m/s). In this experiment we chose a coincidence timing resolution of 409 ps to mimic the expected coincidence timing performance of the EXPLORER scanner (Lv et al. 2017).

The coincidence processor was designed to run with multiple central processing unit (CPU) threads to improve performance by splitting the list-mode singles file into multiple data sections and processing each section simultaneously. The coincidence processor shares a common read and write thread and uses synchronous file input/output (I/O). Methods that involve splitting the file typically lose some - although most likely negligible depending on the size of the coincidence processing buffer - valid coincidences since the singles at the end of one section are not paired with the singles at the beginning of the following section (Goldschmidt et al. 2013, Strydhorst and Buvat 2016). Our coincidence processor ensured that all potential coincidences are considered by using an oversized buffer that overlaps with adjacent data sections without processing the same reference single (i.e. the single that opens the primary, fixed time window) more than once. This eliminated the hard boundaries between data sections.

Details regarding the compilation environment and specific functions used for the coincidence processor can be found in Table 3 in the Appendix.

2.3. Test data

PET singles data with varying incoming singles rates (from approximately 50 Mcps to 500 Mcps) were simulated using GATE v8.0. The GATE installation was checked using a Poisson validation protocol based on a method described by Tries et al. (1999). The simulations included an LYSO-based total-body PET scanner and a water-based cylindrical phantom (16 cm dia., 150 cm long) with an offset line source (6 mm dia., 150 cm long). The specifications of the LYSO crystals used in the simulations were obtained from Crystal Photonics, Inc. (Sanford, FL, United States) via private communication. The radioactive background from LYSO crystals was also simulated and incorporated into the test data. No dead-time was simulated in the scanner. Simulation parameters of the test data are shown in Table 1.

2.4. Coincidence processing performance

The test data generated from GATE were processed using parameters shown in Table 2. A Dell PowerEdge R730 rack server with dual Intel Xeon E5-2650 v4 CPUs and 128 GiB random-access memory (RAM) was used for this evaluation. The coincidence processing performance was then obtained by dividing the number of singles in the list-mode file by the

program execution time. For each singles data rate, the same list-mode singles file was repeatedly processed 10 times to obtain the mean coincidence processing performance.

The evaluation of the coincidence processor was split into the following categories: 1) storage type and 2) overall performance.

2.4.1. Storage type: An Intel 750 series Peripheral Component Interconnect Express (PCIe)-attached solid state drive (SSD) (model SSDPEDMW012T4X1) and a high-performance Dell PERC H730 integrated redundant array of independent disks (RAID)-6 controller w/ 8, Seagate 8TB Enterprise hard disk drives (HDDs) (model ST8000NM0075) were compared to determine if differences in performance could be observed between the 2 storage types. We reset the file cache after each run to ensure the list-mode file was loaded from the storage drive instead of the RAM. We also evaluated the native performance of the coincidence processor by loading the data into RAM prior to processing to determine if file I/O may be the primary bottleneck of the code.

2.4.2. Overall performance: The coincidence processor was compiled and run with varying number of CPU threads (up to 48 threads) to determine if a CPU bottleneck can be observed in the coincidence processor. We also assessed the overall performance effects of processing multiple files simultaneously (which could be useful for batch processing) between the 48 CPU threads.

3. Results

3.1. Storage type

Figure 2 shows that for both PCIe SSD and RAID 6 HDDs, when processing a single file using a single CPU thread, the coincidence processor could process approximately 2.5 and 1.5 M singles/s at incoming singles rates of approximately 150 and 500 Mcps, respectively. If the data was pre-loaded into the RAM, the performance increased to 5.7 and 3.3 M singles/s, an increase of 128% and 120%.

When processing a single file using all 48 threads, the PCIe SSD configuration outperformed the RAID 6 HDDs configuration by 95% (at 109 M singles/s processing rate) and 50% (at 57 M singles/s processing rate) at incoming singles rates of approximately 150 and 500 Mcps, respectively. The figure also showed that performance increased (with plateauing performance gain) as the number of threads increased for both the PCIe SSD and the RAID 6 HDDs configurations. Pre-loading the data into the RAM led to a performance increase of 10% and 12% over the PCIe SSD configuration, at 120 and 70 M singles/s.

3.2. Number of files

Figure 3 shows that using all 48 threads, a significant difference in performance could be observed between the PCIe SSD and RAID 6 HDDs. In addition, the overall coincidence processing performance increased with increasing number of files processed in parallel with the PCIe SSD configuration, while the coincidence processing performance degraded when multiple files were processed in parallel with the RAID 6 HDDs configuration (especially when 3 files were processed in parallel).

4. Discussion

We have shown there was a noticeable performance gain by using a PCIe SSD over a high-performance RAID 6 HDDs setup. In addition, when multiple files were processed in parallel, the RAID 6 HDDs configuration suffered performance degradation (especially when 3 files were processed in parallel) when compared to the SSD configuration. This was likely due to the RAID controller struggling with multiple parallel and uncoordinated I/O requests from multiple independent processes. We suggest it was not an issue when processing a single file because sequential I/O requests originated from the same process. To overcome this limitation in HDDs, the I/O from all processes would need to be properly sequenced. However, this would only be possible at the expense of increased complexity of the code, which may not be justifiable especially since SSDs have become relatively affordable in recent years.

In our case, the most effective way to maximize the coincidence processing performance was to simultaneously process 6 or 12 files using the available 48 CPU threads for the PCIe SSD configuration. Figure 4 shows the number of computer nodes needed to achieve near real-time coincidence processing for a given singles data rate.

While it is possible to store data directly in RAM and process events in real-time as an alternative to SSD storage, for larger systems (such as the EXPLORER scanner) it may lead to a less exible implementation depending on factors such as the pre-processing requirements of raw data (e.g. sorting incoming singles data from multiple scanner sections in chronological order). Also, with our setup we showed that the maximum performance gain over the PCIe SSD configuration (when processing 1 file) was only about 10%. At an incoming singles rate of 50 and 100 Mcps, the PCIe SSD configuration (when processing 6 or 12 files) marginally outperformed the RAM configuration. We suggest that this may have been due to the overhead penalty from launching multiple threads simultaneously after pre-loading the entire data file into RAM, which may not have been an issue at higher incoming singles rates where processing comprises the bulk of the execution time.

In cases where the data acquisition and real-time coincidence processing are performed on the same disk, we expect the PCIe SSD configuration to be better suited for handling multiple independent I/O requests when compared to the RAID HDD configuration - as we have shown that the RAID HDD configuration performed poorly when handling multiple uncoordinated I/O requests in parallel.

Although dead-time was not modeled in the simulations, we expect the presence of dead-time to improve performance at higher rates as it will reduce the number of single events and subsequently possible LORs for coincidence processing.

5. Conclusion

We have developed a software-based coincidence processor for the EXPLORER total-body PET scanner to process its list-mode singles data. To process single events at an incoming singles rate of 150 Mcps, we have determined that the use of 2, 48-thread computer nodes

with dual Intel Xeon E5–2650 v4 CPUs and a PCIe SSD would enable near real-time coincidence processing.

6. Acknowledgements

This work was supported by NIH R01 CA206187, NIH R01 CA170874 and the UC Davis Research Investment in Science and Engineering Program.

7.: Appendix

Algorithm 1: Pseudocode implementation of the coincidence processing algorithm

```

// DEFINITIONS
// A qualified single is a single that is accepted by the energy window.
// A reference single is a qualified single that opens a time window.
// A non-reference single is a qualified single that is inside a reference single's
time window.

for each reference.single do
  for non-reference.singles in the time window do
    if  $LOR(ref, non-ref)$  is outside the transaxial FOV then
      | continue; // coincidence event is rejected
    else if  $LOR(ref, non-ref)$  is greater than the maximum crystal ring
      difference then
      | continue; // coincidence event is rejected
    else if  $time.difference(ref, non-ref)$  is greater than the variable time
      window then
      | continue; // coincidence event is rejected
    else
      | record.coincidence.pair(ref, non-ref); // coincidence event is accepted
    end
  end
end
end

```

References

- Goldschmidt B, Lerche CW, Solf T, Salomon A, Kiessling F and Schulz V (2013), 'Towards software-based real-time singles and coincidence processing of digital PET detector raw data', *IEEE Transactions on Nuclear Science* 60(3), 1550–1559.
- Lv Y, Lv X, Liu W, Judenhofer MS, Cherry SR and Badawi RD (2017), 'Mini-EXPLORER II - a prototype high-sensitivity PET/CT scanner for human brain and companion animal scanning', presented at the IEEE Medical Imaging Conference, Atlanta, Georgia, United States (DBIS-01–3).
- McElroy DP, Hoose M, Pimpl W, Spanoudaki V, Schuler T and Ziegler SI (2005), 'A true singles list-mode data acquisition system for a small animal PET scanner with independent crystal readout', *Physics in Medicine and Biology* 50(14), 3323–3335. [PubMed: 16177512]
- Moraes ER, Poon JK, Balakrishnan K, Wang WL and Badawi RD (2015), 'Towards component-based validation of GATE: Aspects of the coincidence processor', *Physica Medica* 31, 43–48. [PubMed: 25240897]
- Oliver JF, Torres-Espallardo I, Fontaine R, Ziegler SI and Rafecas M (2009), Comparison of coincidence identification techniques for high resolution PET, *in* 'IEEE Nuclear Science Symposium/Medical Imaging Conference', IEEE Nuclear Science Symposium Conference Record, pp. 4005–+.
- Poon JK, Dahlbom ML, Moses WW, Balakrishnan K, Wang WL, Cherry SR and Badawi RD (2012), 'Optimal whole-body PET scanner configurations for different volumes of LSO scintillator: a simulation study', *Physics in Medicine and Biology* 57(13), 4077–4094. [PubMed: 22678106]
- Streun M, Brandenburg G, Larue H, Parl C and Ziemons K (2006), 'The data acquisition system of ClearPET neuro - a small animal PET scanner', *IEEE Transactions on Nuclear Science* 53(3), 700–703.
- Strydhorst J and Buvat I (2016), 'Redesign of the GATE PET coincidence sorter', *Physics in Medicine and Biology* 61(18), N522–N531. [PubMed: 27589353]

Tries M, Skrable K, French C and Chabot G (1999), 'Basic Applications of the Chi-square Statistic Using Counting Data', *Health Phys* 77, 441–454. [PubMed: 10492352]

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

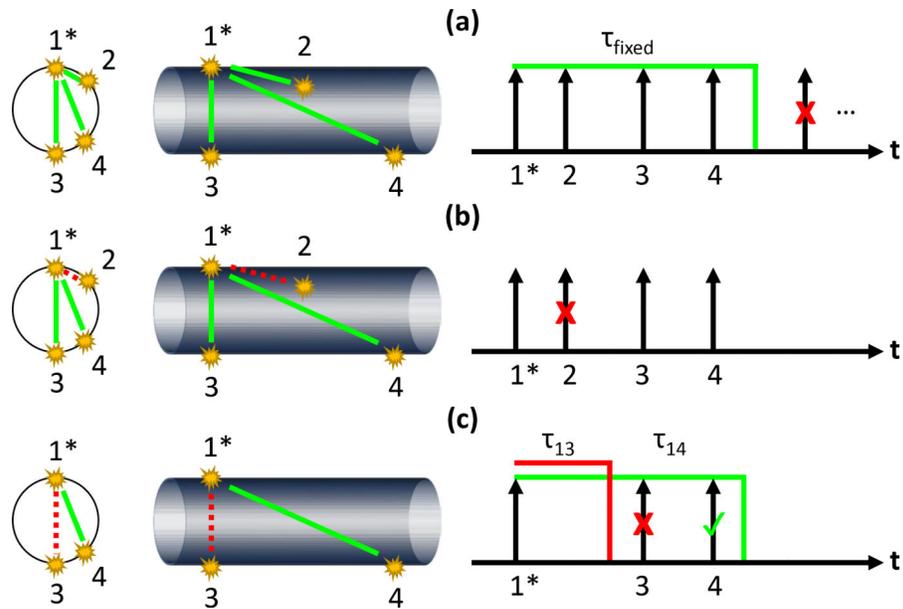


Figure 1.

A visual example of our coincidence processor, shown with 4 total energy-qualified singles in the time window (in 2-D, 3-D views and time domains). (a) 3 potential coincidences (coincidence pairs 1–2, 1–3 and 1–4) in the primary, fixed time window (τ_{fixed}). (b) Coincidence pair 1–2 is rejected by the geometric window. (c) Coincidence pair 1–3 is rejected by the variable time window (τ_{13}), while coincidence pair 1–4 is accepted by the variable time window (τ_{14}). Note: The asterisk (*) denotes the reference single. Also, coincidence pairs 2–3 and 2–4 are not considered in the time window opened by 1 (τ_{fixed}) but will be considered in the time window opened by 2.

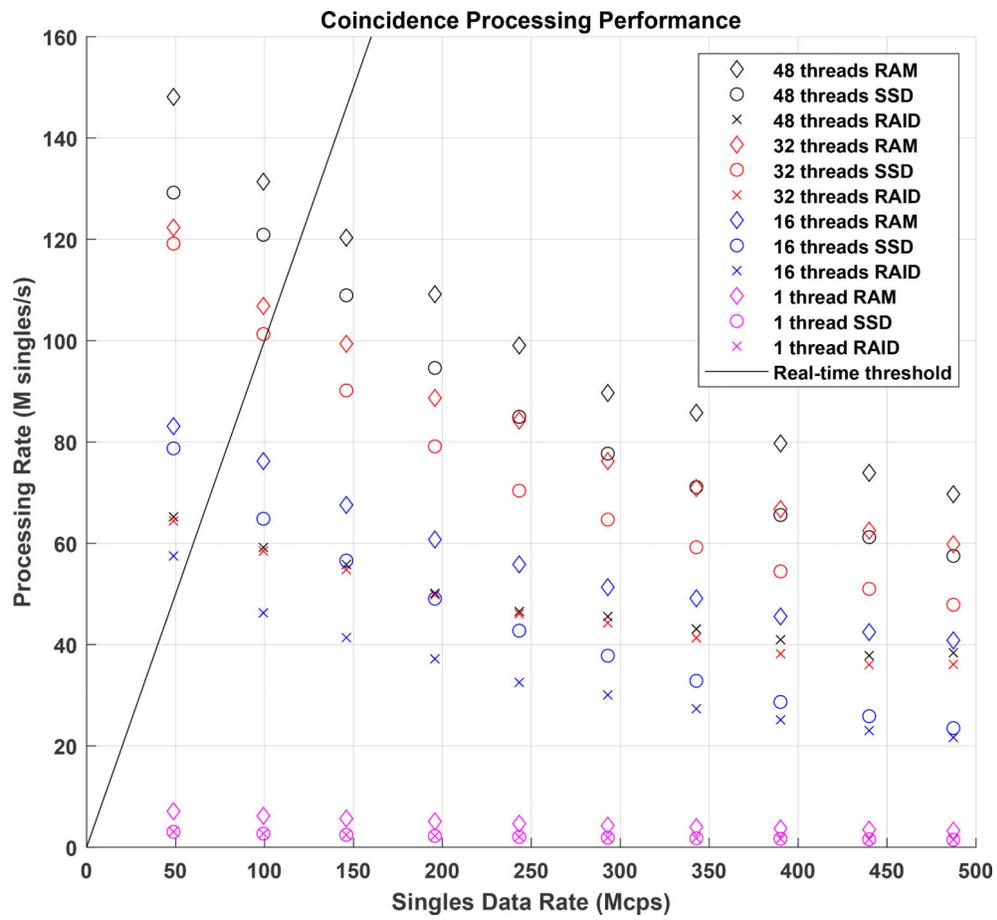


Figure 2. Overall coincidence processing performance of the coincidence processor using different number of threads while processing a single file. The average coefficient of variation (CV) of the data points, calculated over 10 repetitions, is 2.0%. Data points above the real-time threshold reflect data rates that can be processed at near real-time with 1 computer node.

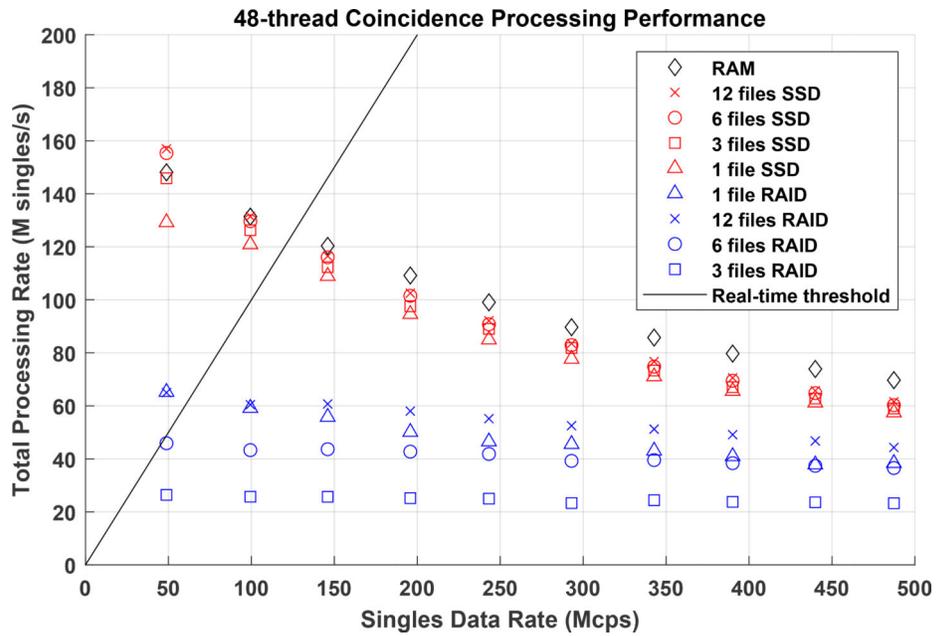


Figure 3. Coincidence processing performance of the coincidence processor with varying number of files. The average coefficient of variation (CV) of the data points, calculated over 10 repetitions, is 3.2%. Data points above the real-time threshold reflect data rates that can be processed at near real-time with 1 computer node.

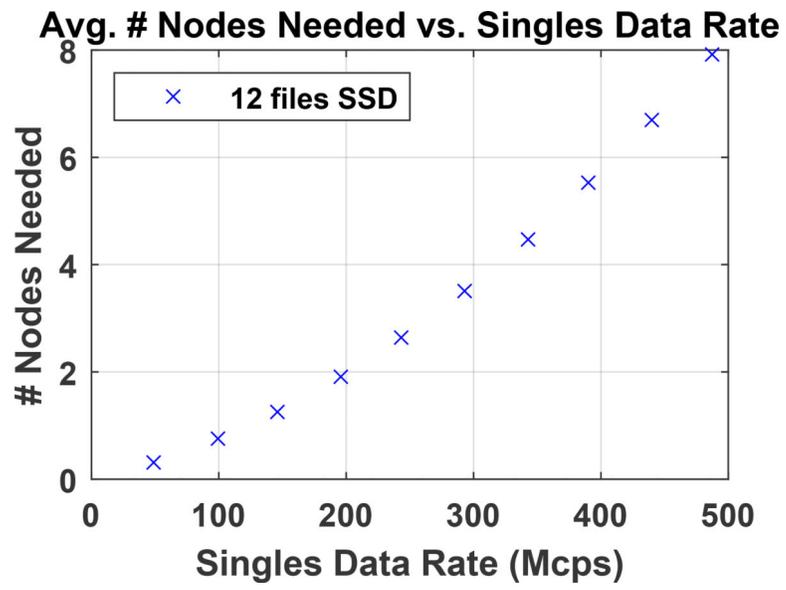


Figure 4. Number of computer nodes needed to achieve near real-time coincidence processing.

Table 1.

Simulation parameters.

PARAMETER	
Crystal size	$2.76 \times 2.76 \times 18.1 \text{ mm}^3$
Crystal pitch	2.85 mm transaxial \times 2.85 mm axial
#crystals per block	7 transaxial \times 6 axial
#block detectors	120-block ring \times 112 rings
Timing resolution	409 ps
Energy range	250 – 712 keV
Energy resolution	11.7% @ 511 keV
Dead time	none
Phantom	water-based cylindrical phantom (16 cm dia. x 150 cm long) with centered line source
Radioactive source	generic 511-keV back-to-back line source (6 mm dia. x 150 cm long)
Source activity	40 – 730 MBq
Resultant incoming singles rate	49 – 487 Mcps (including ~21 Mcps LYSO background)
LYSO density	7.1 g/cm ³
LYSO % by mass	Lu = 71.447%, Y = 4.034%, Si = 6.371% and O = 18.148%
LYSO background	400 Bq/cc
# detected singles simulated	~2 billion singles for each singles data rate

Table 2.

Parameters used for the coincidence processor.

PARAMETER	
Energy window	430 – 712 keV
Time window	fixed (8.0 ns) followed by variable (3.5 to 8.0 ns)
Transaxial FOV restrictions	686 mm
Axial FOV restrictions	restricted via the variable time window
Coincidence policy	Multiple window, takeAllGoods

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Table 3.

Compilation environment and major functions used for the coincidence processor.

PARAMETER	
Operating system	Ubuntu 16.04 LTS
Programming language	C++11
C++ compiler	GCC 5.4
Compilation flags used	--std=c++11 and -pthread
C++ library used	Standard C++ library only
C++ class used for multi-threading	std::thread
C functions used for file I/O	fread() and fwrite()
C++ class used for time measurements	std::chrono::steady clock

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript