Work in Progress: Real-Time GRB Localization for the Advanced Particle-astrophysics Telescope

Marion Sudvarg*, Jeremy Buhler†, Roger Chamberlain‡, Chris Gill§

Department of Computer Science & Engineering
Washington University in St. Louis
St. Louis, Missouri
*msudvarg@wustl.edu, †jbuhler@wustl.edu, ‡roger@wustl.edu, §cdgill@wustl.edu

James Buckley

Department of Physics
Washington University in St. Louis
St. Louis, Missouri
buckley@wustl.edu

Abstract—The Advanced Particle-astrophysics Telescope is a planned mission to perform real-time gamma-ray burst (GRB) detection and localization using SWaP-constrained embedded hardware aboard an orbiting platform. Due to the dynamic and uncertain nature of GRBs, the parallel localization task is dynamic in both workload and deadline. This implies the need for an adaptable framework that adjusts CPU utilization to accommodate overload. To this end, we propose an elastic framework over the workloads of constituent subtasks that allows both continuous and discrete state spaces. Instead of compressing according to constant weights, it instead uses a nonlinear cost function based on the expected angular error in the localized source direction of observed events.

Fig. 1: A Rendering of the APT Instrument

I. Introduction

To study the nature of dark matter and to understand the physics of neutron-star mergers, orbiting gamma-ray telescopes observe gamma-ray bursts (GRBs), collecting and transmitting data for later ground-based analysis. Newly emerging areas of astrophysics seek to perform follow-up observations, enabling the study of GRB emissions across several modalities (e.g., X-rays, visible light, radio and microwaves, cosmic rays, and gravitational waves). However, GRBs are transient events; hence, long delays from initial detection of a GRB's light to ground-based computation of its location in the sky (which is nontrivial to infer from the incoming gamma rays but is necessary to physically aim follow-up instruments) cause lost opportunities for observation.

The Advanced Particle-astrophysics Telescope (APT) [1] (Fig. 1) is a planned space-based observatory that will be deployed at the Sun-Earth Lagrange L_2 orbit, affording it a nearly full-sky field of view. It will fly with onboard computational hardware to detect and localize GRBs in real time [2]; this will enable prompt communication and follow-up observations in multiple spectral bands. We characterize APT's localization as a subtask of multiple other tasks: APT can be considered as just one component of a distributed system with multiple cyberphysical follow-up devices that couple *computation* (e.g., a telemetry system to receive the location of a GRB detected by APT) and *actuation* (the repositioning of a telescope). Each such device is associated with a deadline, after which it can no longer collect useful data. Given the worst-case latency of the associated communication, device computation, and actuation,
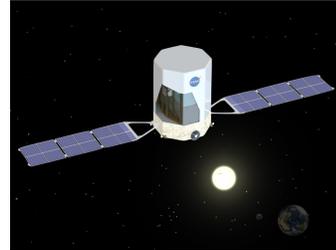
a subdeadline associated with each follow-up device can be assigned to the task of localizing a GRB on APT.

Modeling the GRB detection computation is complicated, since there is no canonical GRB emission spectrum; each GRB is uniquely characterized by how its energy spectrum and brightness evolves over time, which defines the instant after which observing a given band is no longer useful, informing a set of deadlines which are not known a priori. The rate at which data enters APT's onboard computer, as a function of the rate and energies at which photons enter the telescope, is not constant. Further, different physical processes in the detector must be reconstructed by different algorithms [2], [3] in proportions also defined by the spectrum's parameters. Thus, our computational platform must *adapt* to dynamic deadlines and changing workloads to guarantee real-time localization on orbiting hardware with tight SWaP constraints.

To address these problems, we are developing an elastic framework for CPU utilization that aims to estimate workload and deadline constraints based on an initial profile (generated in real time) of a detected GRB. It will then adapt to expected or detected overload first by dedicating CPU resources appropriately to the various interdependent subtasks of *pair reconstruction*, *Compton reconstruction*, and *localization*. If necessary, it will degrade reconstruction accuracy by sampling or dropping a subset of data and reducing refinement iterations (involving elasticity over both continuous and discrete state spaces). Unlike the original elastic scheduling framework, which compresses task utilizations according to proportional weights [4], [5], our framework will need to consider nonlinear weighting over the cost function defined by angular error in source localization. Our framework will target parallel tasks executing on candidate hardware platforms that include both heterogeneous and identical-multiprocessor architectures and will consider compression over each constituent subtask.

II. Background and Related Work

The **Fermi** [6], [7] Gamma-Ray Space Telescope is an existing orbiting observatory with a large field of view (FoV). However, it occupies a low Earth orbit (LEO) and therefore lacks a full-sky FoV. Fermi does not perform onboard GRB localization; while it has produced extensive catalogs of GRBs [8], [9], this limits its ability to contribute to multi-messenger observations of transient astrophysical phenomena. Future planned missions such as **Glowbug** [10] suffer from similar limitations. **APT**, however, will be deployed at the Sun-Earth Lagrange L_2 orbit, where the obscuration of the sky by the earth is minimized and the benefit of the large (nearly 4π-steradian) FoV can be exploited [1]. APT seeks to support efforts in multi-wavelength and multi-messenger astrophysics, allowing follow-up instruments to study detected GRBs across a broad range of emission modalities [11]–[13]. Many such instruments have narrow apertures (often sub-1°) and so must point at the GRB source; APT will perform onboard detection and localization of GRBs in real-time, enabling prompt communication of the source direction.

We have demonstrated that reconstruction of photon trajectories from multiple Compton scattering and subsequent localization of a representative "bright" GRB (i.e., one producing a high volume of data) can be performed in < 200ms on a Raspberry Pi Model 3 B+ (which has a 4-core Cortex-A53) [2]. We built upon the approach in [14] — which reconstructs the path of individual gamma-ray photons by considering all possible orderings of interaction coordinates within a multi-layer detector (Fig. 2 Top) — by instead using a tree search with pruning to provide a deterministic WCET for each photon, then using iterative multilateration over the set of reconstructed photons to estimate a source direction. We extended analysis to heterogeneous platforms [15], with localization implemented in CUDA, achieving estimated < 80ms localization on an NVIDIA Jetson Xavier NX board. In the same work, an FPGA-based approach to infer interaction coordinates consistently completed in 68 cycles (0.23 μs) per event on a Xilinx Alveo U250 accelerator card, which includes an UltraScale+ architecture FPGA.

Our prior work is, however, limited. Evaluation was over a single representative GRB spectrum and did not consider the entire domain of energy spectra and brightness that we desire to detect. All photons were in the Compton regime, and therefore other detection modes (Fig. 2 Bottom) were not considered. Reconstruction was performed over a set of interaction centroids already in a static region of memory when the computation started; realistically, reconstruction will be performed concurrently with data streaming into memory. Finally, the work aimed to minimize execution time but did not consider the various deadlines imposed by the follow-up instruments; therefore, the handling of overload conditions was not considered.

Elastic scheduling [4], [5] provides a framework for dealing with overload by linearly compressing the effective utilizations of individual tasks over a continuous space according to
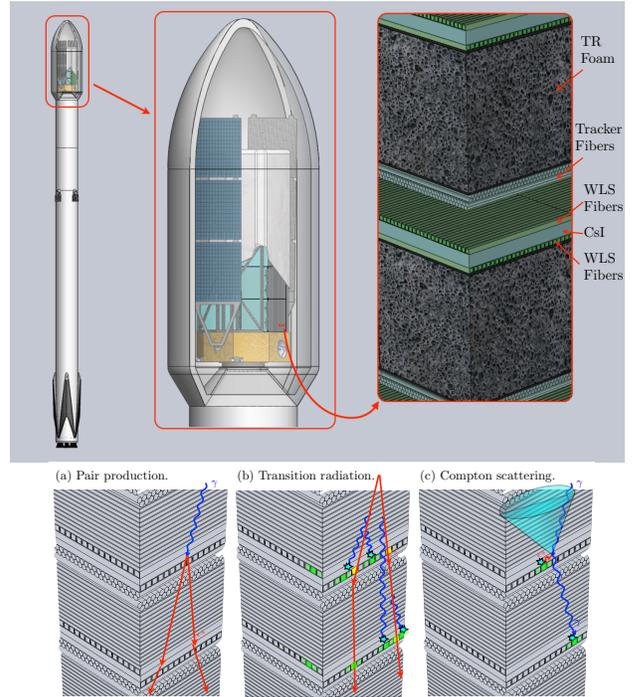

Fig. 2: Top: APT in Falcon-9 faring. Bottom: Detection modes. [1]

weights assigned to each task. It has been reformulated as a quadratic optimization problem [16], [17] and has been extended to federated scheduling of parallel tasks [18], [19] including those tasks constrained to discrete utilization values [20]. Unlike in prior work, our system will need to compress subtasks individually, assigning weights to each according to a nonlinear cost function, while keeping overhead induced by the framework low such that state transitions do not significantly contribute to system overload. In [21], we demonstrated a quasilinear-time solution and a linear admission control algorithm for elastic scheduling on a uniprocessor, and in [22], we demonstrated pseudopolynomial heuristics to assign processors to integer-valued parallel tasks. Similarly efficient methods will need to be developed for more complex elastic scheduling of parallel tasks.

III. System Model

Computation Pipeline: We represent the proposed system as a pipeline consisting of several stages diagrammed in Fig. 3. APT's detector has layers of optical fiber arrays; each fiber is read by a photodetector coupled with an analog-pipeline waveform digitizer ASIC [23]. Each layer array is multiplexed by a single FPGA (e.g., a rad-hard Microchip RT PolarFire), which receives a trigger notification when a constituent ASIC detects signal indicative of a GRB event. The FPGA receives, demultiplexes, and time-integrates signal intensities, then performs centroiding (data reduction to infer the coordinates and energies associated with a photon's interactions in the detector) for each detected photon. The FPGA sends data to the CPU's main memory (e.g., over a time-sensitive network handling transmissions from as many as 40 FPGAs).
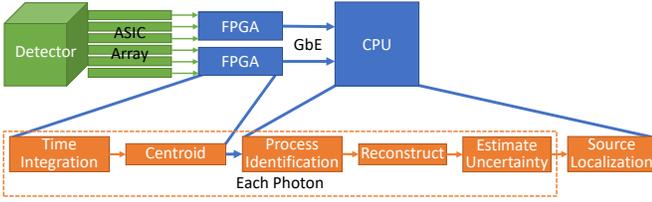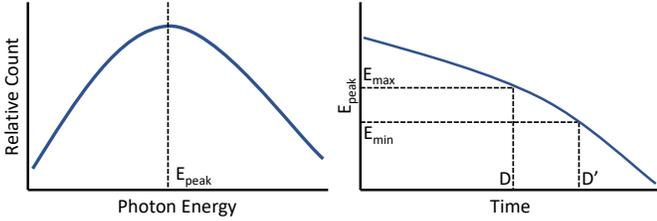
Fig. 3: APT Computation Pipeline

Fig. 4: Left: A GRB spectrum. Right: D, D' from $E_{peak}(t)$, E_{min}, E_{max}

The CPU must combine the received data for each detected photon, then identify the physical process (currently, Compton scattering or pair production) that generated the observed signals. It uses the corresponding algorithm to reconstruct the photon and estimate the uncertainty of the associated result. These are propagated to the source localization stage, which combines data from multiple incident photons. Localization must be completed in time to guarantee the end-to-end deadline requirements for pointing secondary instruments. For now, we assume a sufficient memory buffer such that throughput for processing data transmitted from the FPGA is not a concern; consideration of memory constraints is deferred to future work.

End-to-End Deadline: We define a collection of follow-up instruments $\mathbf{I} = \{I_i\}$, each sensitive to a spectral range $[E_i^{min}, E_i^{max}]$. At a given instant t, a GRB emits a spectrum characterized, among other parameters, by its peak energy $E_{peak}(t)$; the spectrum evolves over time, and the function is unique to each GRB. We define $t = 0$ as the time at which photons emitted by the GRB first enter the detector, t_i^{min} as the time where $E_{peak} = E_i^{max}$ and similarly for t_i^{max}. For now, we assume that E_{peak} is monotonically decreasing in the region $[E_i^{min}, E_i^{max}]$, as in Fig. 4; study of GRB catalogs is ongoing to verify monotonicity and to identify other properties of E_{peak} (e.g., concavity). Peak times imply a soft deadline $D_i = t_i^{max}$, after which the GRB begins to emit fewer photons in the observable spectrum of I_i, and a firm deadline $D_i' = t_i^{min}$, after which I_i cannot make useful observations.

Each instrument is associated with a latency δ_i that includes delivery of the burst alert (speed-of-light from L_2 orbit to Earth is ≈ 5s, though some instruments may also be in an L_2 orbit or even on board APT itself) and the time to repoint the instrument. We additionally assume a worst-case latency δ_{CPU} between a gamma-ray photon's arrival in the detector and the associated data's arrival in memory. This allows us to define a subdeadline D for event reconstruction and source localization on the CPU as $\min\{D_i - \delta_i\} - \delta_{\mathrm{CPU}}$. We expect D to be subsecond for fast GRBs, though it may be on the order of several seconds for slow events.
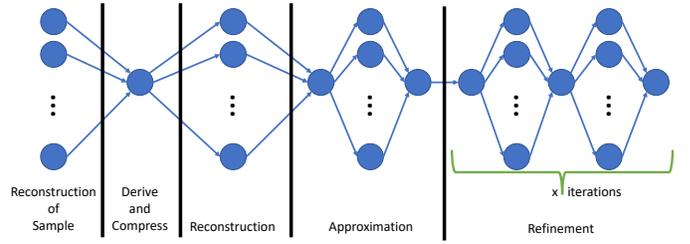

Fig. 5: DAG representation of the parallel CPU execution task.

Overload: Individual photon processing (which includes process identification, reconstruction, and propagation of uncertainty) depends on its associated physical process; we define WCETs C_c for Compton-scattering and C_p for pair-production. The expected fraction $r(t)$ of Compton-scattering photons (a function of the emission spectrum) allows us to define an average WCET $C = r(t)(C_c - C_p) + C_p$ per photon. The rate of photon arrival, R, is a function of the brightness of the GRB.

The function $E_{peak}(t)$ is not known a priori. However, given an initial sample of n_{fit} photons, $E_{peak}(0)$ can be fit to a Band function [24] and matched against known GRB data (e.g., from the Fermi catalogs in [8], [9]) to estimate $E_{peak}(t)$ and derive the deadline and WCET for reconstruction and propagation of uncertainty. For simplicity, we use a constant WCET C_r derived from the worst-case estimated $r(t)$. The derivation subtask has WCET C_{fit} and is released at time $\delta_{\mathrm{CPU}} + n_{fit}/R$.

Given a sufficiently large D, a streaming execution model can be used, where all data is reconstructed as it is collected; then, once data stops arriving (or the arrival rate slows significantly), the source localization stage runs. However, if D cannot be met, the system is considered overloaded. In this case, we model computation on the CPU as the parallel DAG task illustrated in Fig. 5 and elastically compress its constituent subtasks by solving the following optimization problem, which seeks to minimize the expected error in source localization while meeting the deadline constraint:

$$\text{min:} \quad \mathbf{error}(n, n_a, x) \tag{1}$$
$$\text{s.t.:} \quad C_r \cdot n/m + C_{fit} + C_l(n, n_a, x) \leq D \tag{2}$$
$$n \leq R \cdot (D - C_l(n, n_a, x)) \tag{3}$$
$$n_a \leq n \tag{4}$$
$$x \in \mathbb{N} \tag{5}$$

Expected error **(1)** is a nonlinear, monotonically increasing function of three variables: n, the number of photons selected for trajectory reconstruction; n_a, the number of reconstructed photons sampled for an initial approximation of the GRB's location; and x, the number of subsequent refinement iterations to improve the location estimate (the localization algorithm is detailed in [2]). Similarly to E_{peak}, **error** is not known a priori. Using simulations of known GRBs from the catalogs, we can estimate error functions offline. This allows the online compression framework to select an **error** according to the Band function fit from the initial sample of photons.

Because of the highly parallelizable nature of several stages of the pipeline, latency can be characterized according to the

expression on the left side of **(2)**. This constraint guarantees that the time between photon arrival in the instrument and associated data arrival in main memory, plus total reconstruction time (parallelized over the CPU's m cores), fitting, and localization WCET C_l does not exceed the deadline. C_l is polynomial in n, n_a, x and is characterized by the following equation (described in [2]), where each a_i is constant:

$$x(a_0 \cdot n^2 + a_1 \cdot n) + a_2 \cdot n + a_3 \cdot n_a + a_4 \qquad (6)$$

In **(3)**, the photons selected for reconstruction are constrained by the number that have become available before localization must begin. Equation **(4)** constrains the photons sampled for initial approximation to those that have been reconstructed. The values n and n_a are expected to be large enough to approximate a continuous space, but **(5)** restricts x to the natural numbers.

Solving this optimization problem is the topic of ongoing work. While the functions **error** and C_l have not yet been fully characterized, we suspect the nonlinearity will make it too computationally intensive to solve online. Generating an offline solution for each representation GRB spectrum from the catalog would reduce the execution time for online compression. However, neither R nor $\mathbf{I}$ are known a priori: the collection of available instruments changes as ground-based telescopes may be out-of-view due to Earth's rotation, and instruments may be occupied or taken offline. However, as this is not a hard real-time problem, approximate solutions should be sufficient. An offline solution might be given as a function of R (or for a set of discrete values of R). Further, we might define a few sets of available instruments depending on the time of day, which would allow a deadline D to be assigned to each representative GRB from the catalog, similarly to E_{peak} and **error**. We are also considering fast methods to search for an approximate solution online, e.g., by using a genetic algorithm [25].

IV. CPU AND OS REQUIREMENTS

Our task pipeline will run atop SWaP-constrained embedded hardware onboard an orbiting platform. We have tested several of its algorithms on a Raspberry Pi Model 3B+ ([2], [15]). A suborbital demonstration mission, in which a smaller version of the APT instrument will fly on a high-altitude balloon, is currently being designed with an Intel Atom-based single-board computer. APT, however, will fly at the L_2 Lagrange point, which presents the additional challenge of radiation hardening.

The current APT architecture requires an FPGA per detector layer; at 40 layers, sufficient networking capabilities, including possible support for a TSN protocol, will be required. The CPU's board will need a high-bandwidth (e.g., gigabit) network adapter with DMA capabilities, requiring OS and driver support. It will additionally need to communicate burst alerts, requiring additional support for telemetry equipment (which will likely be accessed over a serial bus).

Execution of the CPU stages of our pipeline (process identification, photon trajectory reconstruction, estimation of

uncertainty, and localization) can execute as a single binary which can also encode the logic for both determining and implementing task compression. As such, a targeted unikernel compile of Linux [26] that integrates all necessary drivers and the GRB source localization program might be ideal for our purposes. However, other processes may need to execute concurrently, including real-time mission-critical instrument control tasks. For such task sets, the target operating system may need to provide both priority-based scheduling and strong temporal isolation. For example, CPU reservations (such as those provided by cgroups and real-time group scheduling in Linux) can be used to enforce the target utilization of the localization task and prevent overruns from affecting other tasks on the system. Furthermore, mechanisms such as scCaps in seL4 [27] can, in addition to providing bandwidth constraints, enable a system to switch criticality modes in the event of overruns.

V. CONCLUSION

We have presented an elastic model for compressing task utilization by reducing individual subtask workloads according to a nonlinear cost function. Characterization of representative GRB spectra, their evolution in time, and the corresponding parameters of the optimization problem presented in Sec. III are ongoing through simulations, measurements, and study of GRB catalogs. However, we suspect the problem will be computationally intensive to solve online as part of the onboard localization pipeline. But because this is a soft real-time problem, we intend to instead find an approximate solution (either by precomputing a set of compression modes from which the closest one can be selected, or with online approximation using a fast search technique such as a genetic algorithm). Overrun might result in missed opportunities for follow-up observations but will not cause system failure. Time remaining before the deadline can be used to reconstruct additional photons, then perform additional refinement over the larger set of data.

Our model has room for further refinement. As other tasks may run concurrently, we need to consider how this affects schedulability of the parallel localization task. Under federated scheduling, our pipeline would be assigned dedicated cores, but with only 4 cores on the considered hardware platforms, this may result in unnecessary resource waste. Alternative analytical frameworks, such as semi-federated scheduling, could reduce resource waste but would further complicate the proposed elastic scheduling model. Additionally, memory constraints must be considered: to avoid dropping data transmitted from the FPGA (which must additionally be saved to secondary storage), a buffer must be allocated according to the maximum expected data volume and rate and the reconstruction throughput, which itself is elastic. A suitable OS, such as a real-time microkernel (e.g. seL4 [27]) or a targeted unikernel compile of Linux [26], is still being sought. We welcome suggestions and feedback from the community.

REFERENCES

[1] J. Buckley, S. Alnussirat, C. Altomare *et al.*, "The Advanced Particle-astrophysics Telescope (APT) Project Status," in *Proc. of 37th International Cosmic Ray Conference — PoS(ICRC2021)*, vol. 395, Jul. 2021, pp. 655:1–655:9.

[2] M. Sudvarg, J. Buhler, J. H. Buckley, W. Chen *et al.*, "A Fast GRB Source Localization Pipeline for the Advanced Particle-astrophysics Telescope," in *Proc. of 37th International Cosmic Ray Conference — PoS(ICRC2021)*, vol. 395, Jul. 2021, pp. 588:1–588:9.

[3] W. Chen, J. Buckley, S. Alnussirat *et al.*, "The Advanced Particle-astrophysics Telescope: Simulation of the Instrument Performance for Gamma-Ray Detection," in *Proc. of 37th Int'l Cosmic Ray Conference — PoS(ICRC2021)*, vol. 395, 2021, pp. 590:1–590:9.

[4] G. C. Buttazzo, G. Lipari, and L. Abeni, "Elastic task model for adaptive rate control," in *IEEE Real-Time Systems Symposium*, 1998.

[5] G. C. Buttazzo, G. Lipari, M. Caccamo, and L. Abeni, "Elastic scheduling for flexible workload management," *IEEE Transactions on Computers*, vol. 51, no. 3, pp. 289–302, Mar. 2002. [Online]. Available: http://dx.doi.org/10.1109/12.990127

[6] C. Meegan, G. Lichti, P. N. Bhat *et al.*, "The Fermi gamma-ray burst monitor," *The Astrophysical Journal*, vol. 702, no. 1, pp. 791–804, aug 2009. [Online]. Available: https://doi.org/10.1088%2F0004-637x%2F702%2F1%2F791

[7] W. B. Atwood, A. A. Abdo, M. Ackermann *et al.*, "The large area telescope on the Fermi gamma-ray space telescope mission," *The Astrophysical Journal*, vol. 697, no. 2, pp. 1071–1102, may 2009. [Online]. Available: https://doi.org/10.1088%2F0004-637x%2F697%2F2%2F1071

[8] M. Ackermann, M. Ajello, K. Asano *et al.*, "The first Fermi LAT gamma-ray burst catalog," *The Astrophysical Journal Supplement Series*, vol. 209, no. 1, p. 11, oct 2013. [Online]. Available: https://doi.org/10.1088%2F0067-0049%2F209%2F1%2F11

[9] M. Ajello, M. Arimoto, M. Axelsson *et al.*, "A decade of gamma-ray bursts observed by Fermi-LAT: The second GRB catalog," *The Astrophysical Journal*, vol. 878, no. 1, p. 52, jun 2019. [Online]. Available: https://doi.org/10.3847/1538-4357/ab1d4e

[10] J. E. Grove, C. C. Cheung, M. Kerr *et al.*, "Glowbug, a low-cost, high-sensitivity gamma-ray burst telescope," 2020. [Online]. Available: https://arxiv.org/abs/2009.11959

[11] I. Bartos and M. Kowalski, *Multimessenger Astronomy*, ser. 2399-2891. IOP Publishing, 2017. [Online]. Available: https://dx.doi.org/10.1088/978-0-7503-1369-8

[12] A. Neronov, "Introduction to multi-messenger astronomy," in *Journal of Physics: Conference Series*, vol. 1263, no. 1. IOP Publishing, 2019, p. 012001.

[13] P. Mészáros, D. B. Fox, C. Hanna, and K. Murase, "Multi-messenger astrophysics," *Nature Reviews Physics*, vol. 1, no. 10, pp. 585–599, 2019.

[14] S. Boggs and P. Jean, "Event reconstruction in high resolution Compton telescopes," *Astronomy and Astrophys. Supp. Series*, vol. 145, no. 2, pp. 311–321, 2000.

[15] J. Wheelock, W. Kanu, M. Sudvarg *et al.*, "Supporting multi-messenger astrophysics with fast gamma-ray burst localization," in *Proc. of IEEE/ACM HPC for Urgent Decision Making Workshop (UrgentHPC)*, Nov. 2021.

[16] T. Chantem, X. S. Hu, and M. D. Lemmon, "Generalized elastic scheduling," in *IEEE International Real-Time Systems Symposium*, 2006.

[17] ——, "Generalized elastic scheduling for real-time tasks," *IEEE Transactions on Computers*, vol. 58, no. 4, pp. 480–495, April 2009.

[18] J. Orr, C. Gill, K. Agrawal, S. Baruah *et al.*, "Elasticity of workloads and periods of parallel real-time tasks," in *Proceedings of the 26th International Conference on Real-Time Networks and Systems*, ser. RTNS '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 61–71. [Online]. Available: https://doi.org/10.1145/3273905.3273915

[19] J. Orr and S. Baruah, "Multiprocessor scheduling of elastic tasks," in *Proceedings of the 27th International Conference on Real-Time Networks and Systems*, ser. RTNS '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 133–142. [Online]. Available: https://doi.org/10.1145/3356401.3356403

[20] J. Orr, J. C. Uribe, C. Gill, S. Baruah *et al.*, "Elastic scheduling of parallel real-time tasks with discrete utilizations," in *Proceedings of the 28th International Conference on Real-Time Networks and Systems*, ser. RTNS 2020. New York, NY, USA: Association for Computing Machinery, 2020, pp. 117–127. [Online]. Available: https://doi.org/10.1145/3394810.3394824

[21] M. Sudvarg, C. Gill, and S. Baruah, "Linear-time admission control for elastic scheduling," *Real-Time Systems*, vol. 57, no. 4, pp. 485–490, Oct 2021. [Online]. Available: https://doi.org/10.1007/s11241-021-09373-4

[22] M. Sudvarg and C. Gill, "Analysis of federated scheduling for integer-valued workloads," in *Proceedings of the 30th International Conference on Real-Time Networks and Systems*, ser. RTNS 2022. New York, NY, USA: Association for Computing Machinery, 2022, pp. 12–23. [Online]. Available: https://doi.org/10.1145/3534879.3534892

[23] K. Bechtol, S. Funk, A. Okumura, L. Ruckman, A. Simons, H. Tajima, J. Vandenbroucke, and G. Varner, "TARGET: A multi-channel digitizer chip for very-high-energy gamma-ray telescopes," *Astroparticle Physics*, vol. 36, no. 1, pp. 156–165, 2012.

[24] D. Band, J. Matteson, L. Ford, B. Schaefer, D. Palmer, B. Teegarden, T. Cline, M. Briggs, W. Paciesas, G. Pendleton, G. Fishman, C. Kouveliotou, C. Meegan, R. Wilson, and P. Lestrade, "BATSE observations of gamma-ray burst spectra. I. spectral diversity," *Astrophys. J.*, vol. 413, p. 281, Aug. 1993.

[25] K. A. De Jong, "An analysis of the behavior of a class of genetic adaptive systems." Ph.D. dissertation, University of Michigan, USA, 1975, aAI7609381.

[26] A. Raza, P. Sohal, J. Cadden, J. Appavoo, U. Drepper, R. Jones, O. Krieger, R. Mancuso, and L. Woodman, "Unikernels: The next stage of linux's dominance," in *Proceedings of the Workshop on Hot Topics in Operating Systems*, ser. HotOS '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 7–13. [Online]. Available: https://doi.org/10.1145/3317550.3321445

[27] "The sel4 microkernel," https://docs.sel4.systems/projects/sel4/, seL4 Foundation, accessed: January 23, 2022.