Combining Data and Visual Aggregation Techniques to Build a Coherent Spatiotemporal Overview

Damien Dosimont*†‡§, Robin Lamarche-Perrin¶, Lucas Mello Schnorr‖, Guillaume Huard†‡*§ and Jean-Marc Vincent†‡*§

*Inria

†Univ. Grenoble Alpes, LIG, F-38000 Grenoble, France

‡CNRS, LIG, F-38000 Grenoble, France

§Email: firstname.lastname@imag.fr

¶MPI for Mathematics in the Sciences, 04103 Leipzig, Germany – Email: robin.lamarche-perrin@mis.mpg.de

‖Informatics Institute, UFRGS, Porto Alegre – Email: schnorr@inf.ufrgs.br

***Abstract*—Analysts commonly use execution traces collected at runtime to understand the behavior of applications running on parallel and distributed systems. These traces are inspected *post mortem* using various visualization techniques that are generally incapable to scale properly for many events. This issue, mainly due to human perception limitations, is also the result of the screen size, which prevents the proper drawing of many graphical objects. Several visualization techniques tackle these issues by reducing the representaton complexity, using visual or data aggregation, or even clustering. Nevertheless, these solutions have drawbacks that hinder the analysis. We first evaluate existing trace visualization techniques using different criteria, involving how they are readable, their fidelity to represent the trace content without modifying its meaning, and so on. The objective is to determine which factors are responsible for the issues mentioned above. Second, we show how the combination of several aggregation techniques, data and visual, through a coherent and uniform treatment on spatial and temporal dimension, helps us to fulfill better the different criteria. This example enable us to claim the necessity of formalizing an aggregation methodology to provide decent spatiotemporal trace overviews for performance analysis.**

***Keywords*-Performance analysis, trace visualization, spatio-temporal aggregation, information theory, NASPB, Grid'5000.**

I. Introduction

Trace analysis techniques based on time visualization help to show the different execution phases, synchronization barriers, causality relations, time dependencies, deadlocks, race conditions and communication sequences. Adding structural information, such as the resource hierarchy or the platform topology, helps to make the link between the temporal behavior and the components involved in computation and communications. Space-time views, inspired by Gantt charts [1], are frequently used by analysts because they are intuitive and combine both dimensions. However, in large scale scenarios with many resources and events, users cannot be provided with a correct overview of the whole trace because of screen limitations, resulting in cluttered visualizations with very small graphical objects and pixelization artifacts [2]. By zooming in and panning to get more details, the analyst usually loses context, making it difficult to figure out which part of the trace is drawn and how representative it is considering the overall application behavior. If the user could dispose of a very large screen, data would be so numerous that it would be difficult to understand them. The limitation of a microscopic space-time view is that it is incapable to depict a *higher-level understanding* of the system states and dynamics. Because anomalies can be global (like bottlenecks), this high-level understanding is necessary to debug the system.

This paper is organized as follows. Section II presents how the different evaluation criteria can be combined to provide a decent trace overview. Section III presents a summary of our efforts to provide a coherent spatiotemporal overview for large-scale scenarios. Section IV concludes the paper with a summary of our claims.

II. Providing a Decent Trace Overview

Different analysis tools propose several methods to improve scalability and provide decent overviews over time and space. Table I gives a summary of these tools and the corresponding techniques. In this section, we compare them by relying on different criteria [3]. We also consider multidimensional criteria in our comparisons, as defined below. Our objective is to determine the current characteristics of these visualizations that cause the disrespect of these criteria.

A. Elmqvist-Fekete criteria

Elmqvist and Fekete [3] propose a methodology to build an overview based on hierarchical aggregation. They insist on six criteria that a consistent visualization using aggregation must fulfill. We summarize them here:

G1. Entity budget: Screen resolution allows the representaton of a finite number of entities, the smallest one being the size of a pixel. Moreover, the human capability to perceive and analyze a great number of graphical objects and data on a screen is also limited.

Table I

SPATIO-TEMPORAL SCALABILITY TECHNIQUES IMPLEMENTED IN TRACE ANALYSIS TOOLS. THE GX ARE ELMQVIST AND FEKETE CRITERIA [3], MX ARE OUR SPATIOTEMPORAL CRITERIA. A CRITERION MIGHT BE SATISFIED: ONLY FOR TIME (⋆), SPACE (○), OR FOR BOTH DIMENSIONS (●)

Visualization	**Technique**	**Tools**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**	**M1**	**M2**
Space-Time	Pixel-guided (⋆), No aggregation (○)	Vampir [4], Paraver [5]	⋆	●	●				●	
Space-Time	Visual Aggregation (⋆), No aggregation (○)	Pajé [6], LTTng Eclipse Viewer [7]	⋆		●	●	●	●	●	
Space-Time	Time compression (⋆), Hierarchical aggregation (○)	KPTrace Viewer [8]	○		●			●	●	
Space-Time	Time abstraction (⋆), No aggregation (○)	Jumpshot [9]	⋆	●	●	●	●	●	●	
Timeline	Pixel-guided (⋆, ○)	Vampir [4]	●	⋆	●					●
Timeline	Information aggregation (⋆, ○)	Ocelotl [10], [11]	●	●	●	●	●	●		●
Task Profile	Clustering (○), Mean Operation (⋆)	Vampir [4]	●	●	●	●	●	●		●
Treemap/Topology	Hierarchical aggregation (○), Time integration (⋆)	Viva [2], [12]	●	●	●	●	●	●		●

G2. Visual Summary: An overview should contain enough details to bring meaning to the analyst, but keep a reasonable quantity of data to avoid the information overload, which leads to a cluttered and unreadable visualization.

G3. Visual Simplicity: Visual aggregates should be clean and simple. Even with a good G1, many aggregates can be used to keep some details, avoiding strong information loss. Complex shapes, too much data associated with a single aggregate might be source of an unreadable representation.

G4. Discriminability: Visual aggregates and data should be easily distinguishable by using a different way to represent them. Indeed, use the same shape may mislead the analyst by masking the nature of the graphical objects.

G5. Fidelity: Data aggregation techniques reduce the amount of data to the detriment of the information quality. In the visualization, data reduction may deform the behavior understanding. If information loss is unavoidable, it is important to inform the user.

G6. Interpretability: This principle counterbalances G1: the intensity of an aggregation should be adapted in order to keep the visualization semantics, being activated only when necessary. The aggregation process should also be understandable by the user.

We mark Gx if criterion x is satisfied, and $\overline{\text{Gx}}$ if it is not.

B. Multidimensional criteria

The G criteria of the previous section disregard the time and space (resources) dimensions. This problem hinders the representation of phenomena that involve the application dynamics and the platform structure. We define two criteria, included in the Table I, to evaluate the management of spatiotemporal dimensions:

M1. Multidimensional representation: The M criterion indicates that a technique manages the representation of temporal and spatial dimensions.

M2. Aggregation coherence: The reduction process is applied simultaneously and coherently to both dimensions. However, it does not mean that both dimensions are represented. Moreover, the aggregation applied on both dimension is not necessarily of the same nature.

C. Discussion

Pixel Guided Representations Issues: Pixel-guided representations, present in some tools [4], [5], associate each screen pixel to a set of data. As the pixel is incapable to represent all the information, the rendering algorithm decides what is shown or hidden. We claim that this aggregation process misguides the user: a pixel might correspond to a raw data or be the aggregation of several data ($\overline{\text{G4}}$), and we do not know which aggregation operation is done ($\overline{\text{G6}}$), except for Paraver [5]. Pixel-guided representations also suffer from a fidelity issue ($\overline{\text{G5}}$): for instance, resizing the window may modify strongly the visualization content because the pixel allocation changes.

Visual Aggregation and G2: In a visual aggregation, the rendering tries to preserve the graphical object scales, whose size depends on their contents. When it is impossible, e.g., when the size is less than one pixel, it generates aggregates gathering close objects. In Pajé [6] and LTTng Eclipse Viewer [7], such aggregates are just used to avoid visual clutter, but do not represent the data they contain ($\overline{\text{G2}}$).

Relation between G1, M1, and M2: In Table I, all techniques based on Space-Time [4], [5], [6], [7], [8], [9] fulfill criterion M1 by proposing a multidimensional view. However, all these tools have an issue with their entity budget ($\overline{\text{G1}}$). Indeed, as the temporal and the spatial dimensions behave differently, visualization techniques does not treat both axis the same way ($\overline{\text{M2}}$). KPTrace Viewer [8], for instance, proposes an interactive hierarchical aggregation for the space dimension and a mechanism that compresses time whenever an event lasts too long. However, if the spatial reduction technique is efficient, the time compression does not manage sufficiently the time entity budget. Other Space-Time tools lack reduction technique on space and disrespect the vertical budget. We determine therefore that $\overline{\text{M2}}$ is at least partially responsible of $\overline{\text{G1}}$.

Conversely, other techniques [4], [10], [11], [2], [12] proposing a unidimensional overview ($\overline{\text{M1}}$) fulfill G1 and M2, and, except for the Vampir Timeline [4], also fulfill all the other G criteria. Vampir's task profile [4] clusters the most similar processes according to a distance measure based on the duration of the functions executed by each process. Here, the temporal dimension is lost in the process. One of our previous works, Viva [12] provides a multiresolution treemap view showing the hardware and software component hierarchy. Entities having the most homogeneous behavior are aggregated using a compromise between the representation complexity and the information loss induced by the aggregation. This technique highlights troubles characterized by an heterogeneous behavior. However, time dimension is missing from the representation, although it is used to compute the entities values through a time integration. Ocelotl [10], [11] uses the same compromise to build a timeline where homogeneous time periods of the trace are aggregated, highlighting macroscopic phases and temporal perturbations. Spatial data is used to determine time aggregates (M2), but is not represented ($\overline{\text{M1}}$).

III. Define a Coherent Spatiotemporal Overview

This section describes the building of a spatiotemporal visualization technique that addresses the scalability of both dimensions, implemented in the Ocelotl visualization tool. It is composed by several aggregations, enabling to fulfill progressively the G and M criteria. The original trace is composed by events which are *produced* by a resource s, at a timestamp τ. Time is an ordered and continuous dimension.

A. Microscopic Description of The Trace

The trace microscopic description is a first reduction technique that enables to abstract it and choose a metric to represent its behavior. To generate this model, we cut the time dimension into several timeslices, discretizing the time axis. The number of timeslices is defined by the user. We organize the set of resources S as a tree, where each level correspond to a subpart of the system. For instance, the root is the whole platform, its nodes are the sites, and the following levels are the clusters, the machines, and so on. This information can be hard-coded in the trace, or added by the user. Then, for each spatiotemporal area, constituted by a resource and a timeslice, we assign a value computed from the raw trace, for instance, a vector containing the time passed in different application states. Fig. 1.a is a graphical example of a microscopic model applied on a trace containing two different types of states x_1 and $x_2 \in X$, where the color intensity translates the ratio between the time passed in both.

B. Information-Based Data Aggregation

The second step consists in aggregating the microscopic model by giving priority to the homogeneous spatiotemporal areas, while preserving data about the heterogeneous ones. Henceforth, aggregation consists in optimizing a trade-off between data reduction and information loss. Information-theoretic measures have been proposed in previous work [13], [12], [10], [11] to express such a trade-off: Kullback-Leibler divergence [14] as a measure of information loss, Shannon entropy [15] as a measure of data reduction, and a *parametrized Information Criterion* [13]:

$$\mathrm{pIC}_x = p\,\mathrm{gain}_x - (1-p)\,\mathrm{loss}_x \tag{1}$$

where $p \in [0,1]$ is the gain/loss ratio used to balance this trade-off. For $p = 0$, the analyst wants to be as accurate as possible (the microscopic partition is optimal) and, for $p = 1$, he wants to be the simplest (the full aggregation is optimal). When p varies from 0 to 1, a whole class of nested representations arises. The choice of this parameter is deliberately left to the analyst, so he can adapt the entity budget to the analysis purposes.

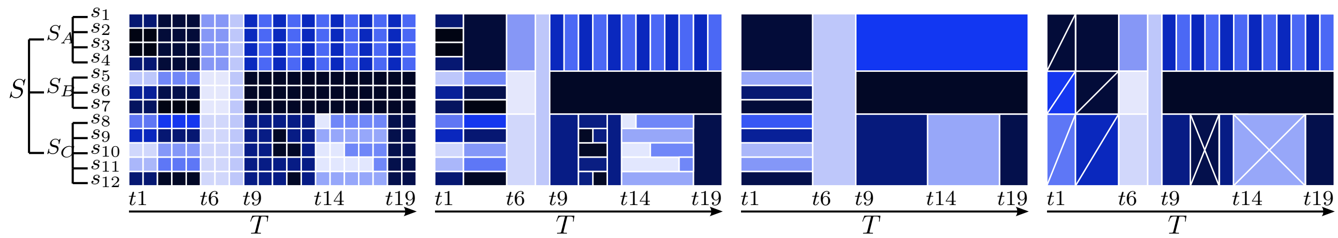
Example of aggregations are depicted in Fig. 1.b and c. p values of these aggregations are ordered as follow: $p_a = 0 < p_b < p_c < 1$. Both aggregations meet criteria G2 (aggregates show homogeneous state proportions), G3 (rectangles are clear and simple), G4 (we distinguish aggregates from microscopic data thanks to their size), G5 (complexity reduction and information loss are provided to the user), and G6 (aggregation is interpreted as a process to detect spatiotemporal areas with homogeneous behavior). M1 and M2 (the aggregation process handles both dimensions symmetrically and simultaneously) are also fulfilled.

In the case where more than two different states are involved in the representation, we associate a color to each possible state and represent in each aggregate the state mode, i.e., the color of the state that has the highest proportion. We also provide information regarding the state mode proportion by applying transparency during the rendering to change the color intensity. Fig. 2 shows the result of this choice. Three state modes `MPI_init` (yellow), `MPI_send` (green) and `MPI_wait` (red) are visible. These features bring useful information (G2) and help to reduce clutter (G3) by avoiding to represent the proportion of all the states.

Figure 2. Ocelotl showing an overview of the execution of the NAS-CG application, class C, 64 processes, on the Grid'5000 Rennes site. The trace is partitioned into aggregates that correspond to a locally homogeneous behavior of the application over time and among a set of computing resources. We distinguish a perturbation around 3,00E9, caused by the concurrent execution of applications competing for network access.

C. Controlled Visual Aggregation

When the number of resources $|S|$ is greater than the amount of pixels in the spatial axis, the visual entity budget is not respected for the microscopic representation and may also be problematic for small spatial aggregates ($\overline{\text{G1}}$). We use visual aggregation (during the rendering) to maintain



a. The microscopic model (12 resources and 20 microscopic time periods)

b. An optimal spatiotemporal aggregation (56 spatiotemporal areas)

c. A higher-level spatiotemporal aggregation (15 spatiotemporal areas)

d. Visual aggregation of 1.b (21 data aggregates and 7 visual aggregates)

Figure 1. Aggregation and visualization of an artificial trace giving the behavior of 12 resources during 20 microscopic time periods (two possible states)

this budget: if an aggregate has a visual height inferior to a threshold (in pixels), its parent is drawn instead. Fig. 1.d shows the difference between *data* and *visual* aggregation. Visually-aggregated areas are marked differently (G4): by a diagonal line, if underlying resources have the same temporal data partitioning, or by a cross, on the contrary.

IV. Conclusion and Positioning

We have compared existing trace visualization techniques that tackle temporal and spatial scalability issues, according to criteria proposed by Elmqvist and Fekete [3] to define a good quality overview, and multidimensional criteria we propose. This comparison brings us the following conclusions. Pixel-guided representations suffer from issues related to an unclear aggregation process, instability, and an implicit information loss. Visual aggregation techniques fail to represent relevant information to the user. Data aggregation techniques are efficient on unidimensional representation, but no technique manages both dimensions correctly.

We also have described our work to obtain temporal and spatial scalability by fulfilling the criteria we propose for our comparison. Our strategy applies data aggregation simultaneously on both dimensions, and helps to distinguish homogeneous part of the trace. It also provides metrics related to the information loss caused by aggregation. We also propose a visual aggregation, merging very small elements, but by staying coherent with the data aggregation pattern and still showing relevant information to the user.

Our position is that a decent trace overview should at least respect the criteria we have gathered. As reduction techniques like aggregation are limited alone, we incite the use of a combination between several of them, applied coherently on both dimensions to avoid an unfair complexity reduction. The criteria we propose are binary, and it seems necessary to elaborate a more subtle notation that enables a true quantitative comparison.

References

[1] J. M. Wilson, "Gantt charts: A centenary appreciation," *European Journal of Operational Research*, vol. 149, no. 2, 2003.

[2] L. M. Schnorr, A. Legrand, and J.-M. Vincent, "Detection and Analysis of Resource Usage Anomalies in Large Distributed Systems Through Multi-Scale Visualization," *Concurrency and Computation*, vol. 24, no. 15, 2012.

[3] N. Elmqvist and J.-D. Fekete, "Hierarchical Aggregation for Information Visualization: Overview, Techniques, and Design Guidelines," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 3, 2010.

[4] A. Knüpfer, H. Brunst, J. Doleschal, M. Jurenz, M. Lieber, H. Mickler, M. S. Müller, and W. E. Nagel, "The Vampir Performance Analysis Tool-Set," in *Tools for High Performance Computing*. Springer-Verlag Berlin, 2012.

[5] V. Pillet, J. Labarta, T. Cortes, and S. Girona, "Paraver: A tool to visualize and analyze parallel code," in *Proceedings of WoTUG: Transputer & Occam Developments*, vol. 44, 1995.

[6] J. Chassin de Kergommeaux, "Pajé, an Interactive Visualization Tool for Tuning Multi-Threaded Parallel Applications," *Parallel Computing*, vol. 26, no. 10, 2000.

[7] "Linux Tools Project/LTTng2/User Guide - Eclipsepedia," http://wiki.eclipse.org/index.php/Linux_Tools_Project/LTTng2/User_Guide.

[8] C. Prada-Rojas, F. Riss, X. Raynaud, S. De Paoli, and M. Santana, "Observation tools for debugging and performance analysis of embedded linux applications," in *Conference on System Software, SoC and Silicon Debug-S4D*, 2009.

[9] E. Lusk and A. Chan, "Early Experiments with the OpenMP/MPI Hybrid Programming Model," in *OpenMP in a New Era of Parall.*, ser. LNCS, vol. 5004. Springer, 2008.

[10] G. Pagano, D. Dosimont, G. Huard, V. Marangozova-Martin, and J.-M. Vincent, "Trace Management and Analysis for Embedded Systems," in *Proceedings of the IEEE 7th International Symposium on Embedded Multicore SoCs*, 2013.

[11] D. Dosimont, L. M. Schnorr, G. Huard, and J.-M. Vincent, "A Trace Macroscopic Description based on Time Aggregation," INRIA, Tech. Rep. RR-8524, 2014.

[12] R. Lamarche-Perrin, L. M. Schnorr, J.-M. Vincent, and Y. Demazeau, "Evaluating Trace Aggregation for Performance Visualization of Large Distributed Systems," in *Proceedings of the 2014 IEEE International Symposium on Performance Analysis of Systems and Software*, 2014.

[13] R. Lamarche-Perrin, Y. Demazeau, and J.-M. Vincent, "Building the Best Macroscopic Representations of Complex Multi-Agent Systems," in *Transactions on Computational Collective Intelligence*, ser. LNCS. Springer-Verlag Berlin, 2014.

[14] S. Kullback and R. Leibler, "On Information and Sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, 1951.

[15] C. E. Shannon, "A Mathematical Theory of Communication," *The Bell System Technical Journal*, vol. 27, no. 3, 1948.