Outils de visualisation de traces

Damien DOSIMONT

23 mars 2012

Sommaire

Complexité croissante des systèmes embarqués

- Systèmes embarqués actuels : de plus en plus complexes
- Matériel :
  - Nombre de coeurs importants, hétérogènes
  - Interconnexions plutôt que mémoire partagée
- Logiciel :
  - OS : noyaux linux
  - POSIX, etc.
- Rapprochement peut être fait avec systèmes distribués de grande échelle
  - Nombre d'entités important
  - Accès aux ressources non uniforme
  - Hétérogéneité (processeurs différents, coprocesseurs, etc.)
  - Cohérence (cache, mémoire, synchronisation)

Intérêt des traces d'exécution

- Traces d'éxecution : répondre (en partie) aux difficultés de la programmation parallèle
  - Analyse des performances
  - Déterminer comportements "anormaux", bugs
  - Optimiser
- Difficultés : traces volumineuses
  - Gestion du stockage
  - Elimination du bruit
  - Reconnaissance de motifs
  - **Réprésentation visuelle**

Difficultés inhérentes à la visualisation

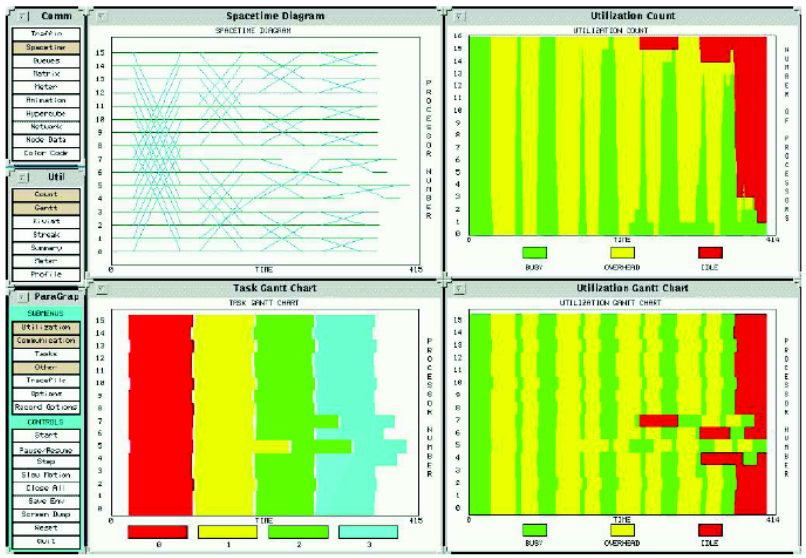
- Espace limité : écran avec résolution fixe
- Nombre d'informations important
  - **Temporalité** : évenements en fonction du temps
  - **Hiérarchie** : entités mères et filles (ex : thread, processus, cœur, machine, cluster, site)
  - **Topologie** : répartition géographique des entités, liens
- Outils actuels :
  - Représentation majoritairement **temporelle** (diagramme de Gantt)
  - Agrégation plus ou moins sommaire
  - Nécessité de **scroller** (abscisse ou ordonnée)
  - Plus ou moins dépendants d'un **format de trace particulier**

Thèse et travaux de Lucas Schnorr

- Répondre aux problématiques précédentes, dans le cadre des systèmes distribués de grande échelle
  - Etat de l'art des techniques et outils de visualisation existants
  - Focus sur Pajé : outil et format de trace
  - Triva :
    - Time-Slice Algorithm
    - Treemap
    - Modèle 3D
    - Graphe topologique
  - Etudes de cas
    - KAAPI
    - MPI
    - BOINC/SimGrid

Outils de visualisation : état de l'art (1)

- ParaGraph
  - Concept de "Simulation"
  - Implémentation de plusieurs techniques de visualisation
  - Architecture basée sur évenements
  - Interface interactive : 25 affichages (synchronisés), 3 familles
    - Utilisation
    - Communication
    - Tâches
  - Développement non maintenu : couplage avec bibliothèque PICL

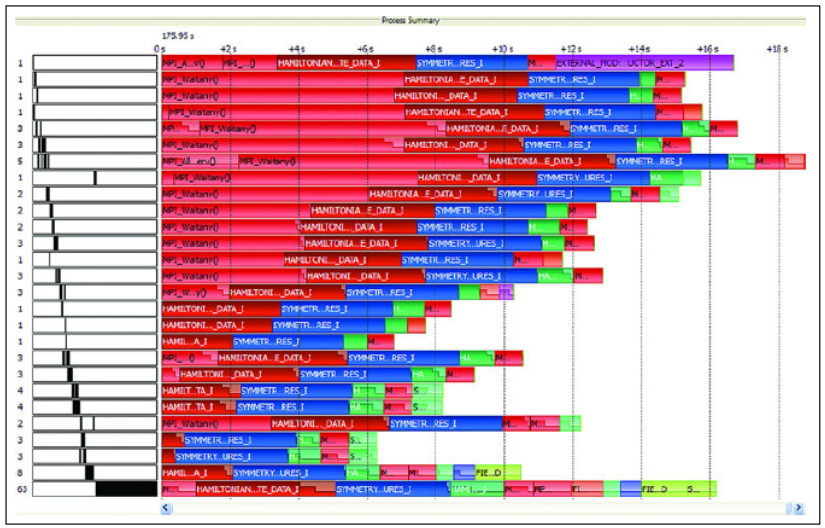


Outils de visualisation : état de l'art (2)

- TraceView
  - Généralisation, et définition de l'organisation des outils de visualisation
- Pablo
  - Modularité
  - Format SDDF
- Paradyn
  - Axé sur les performances des systèmes parallèles
  - Instrumentation dynamique
- Jumpshot
  - MPI
- ParaProf
  - Reprend les techniques les plus performantes des autres outils
  - Différentes techniques 3D
  - OpenGL

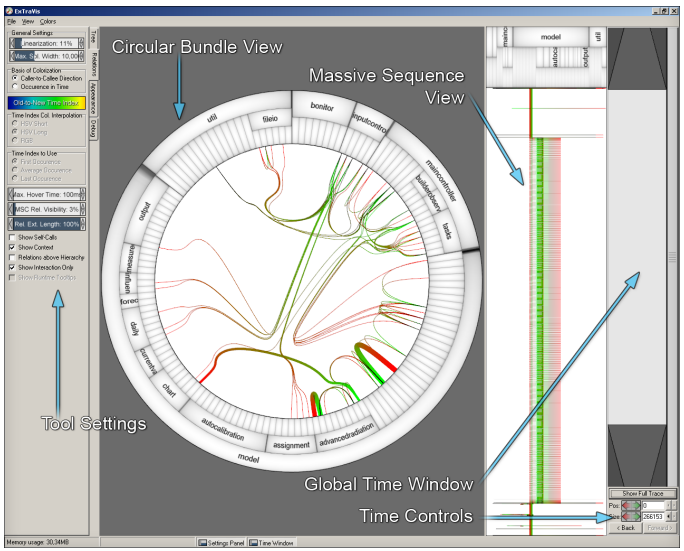
Outils de visualisation : état de l'art (3)

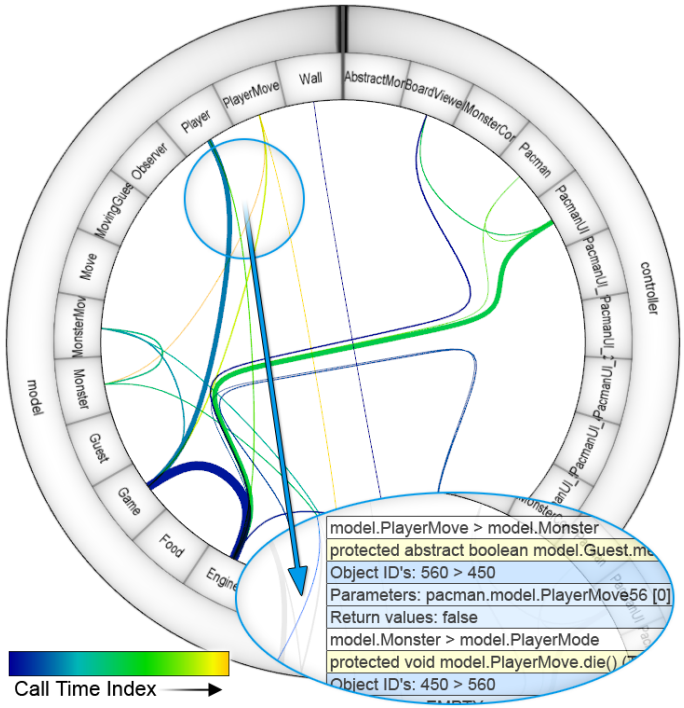
- Vampir
  - Analyse des applications parallèles + OpenMP/MPI
  - Grande scalabilité
  - Set de filtres flexibles
  - Grand nombre de visualisations disponibles (Gantt, matrix, histogrammes, timelines)
  - Visualisation hiérarchique basée sur diagrammes de Gantt (3 niveaux)
  - Regroupement des tâches par profile (clusterisation)
  - Format OTF
  - Commercial :-(



Outils de visualisation : état de l'art (4)

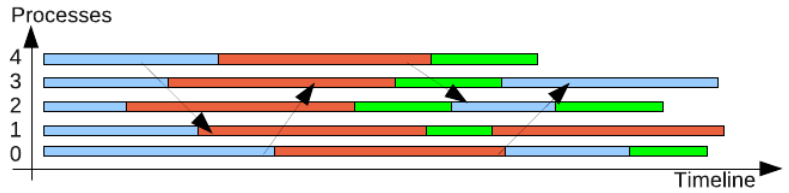
- Extravis



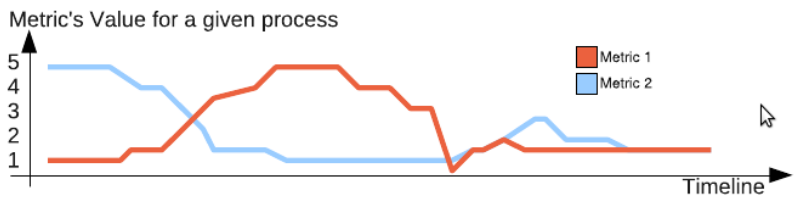


Techniques de visualisation

- Comportementales
  - Gantt-Charts
    - Implémenté dans la majorité des outils
    - Pajé et Vampir -> mécanismes de groupement hiérarchique

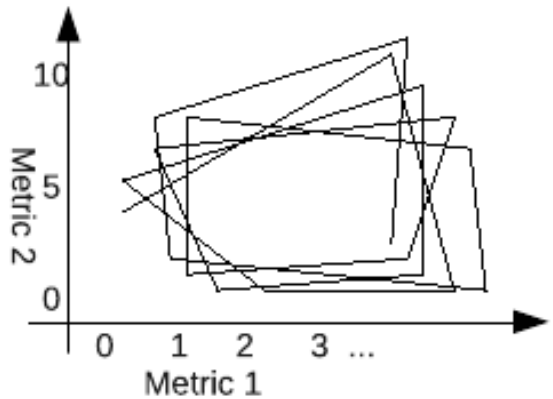


  - Variables en 2 ou 3D
    - Implémenté dans la majorité des outils
    - Variables dans Pajé, Communication traffic, Utilization Count dans ParaGraph, Performance Counters dans Vampir
    - Paradyn : 3D -> corréler deux métriques + ligne de temps



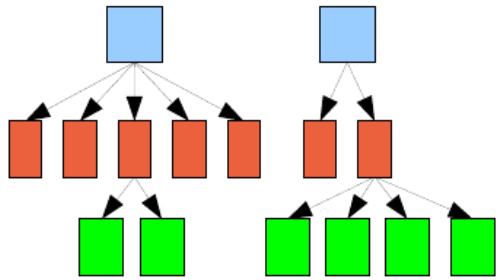
Techniques de visualisation (2)

- Comportementales (2)
  - Time-Tunnel (Virtue)
    - 2 dimensions pour placer processes, au milieu d'un cercle, et 3e pour la ligne de temps
  - Phase Portraits

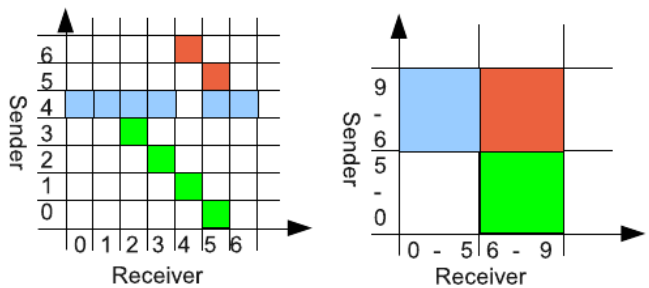


Techniques de visualisation (3)

- Structurelles
  - Call Graphs
    - Interactions parmi les composants de l'application -> ParaProf, Virtue

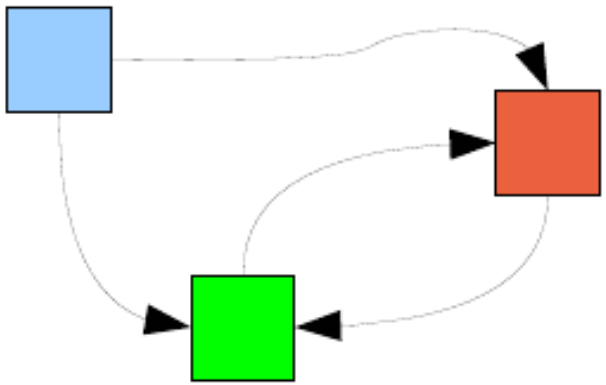


  - Matrix
    - Proposé en premier par ParaGraph
    - ex : Communications
    - Problème de scalabilité résolu dans Vampir en regroupant les processus (nombre ou autres caractéristiques)



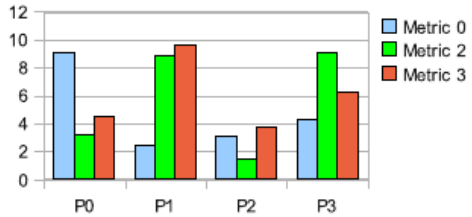
Techniques de visualisation (4)

- Structurelles (2)
  - Graphes avec communications
    - Présent dans ParaGraph, mais trop peu d'informations, par exemple sur les liens

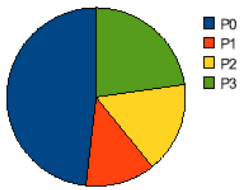


Techniques de visualisation (5)

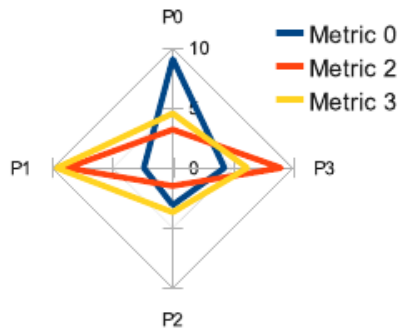
- Statistiques
  - Histogrammes



  - Diagramme circulaire



  - Diagramme de Kiviat



  - Représentations 3D

Outil de Visualisation Pajé

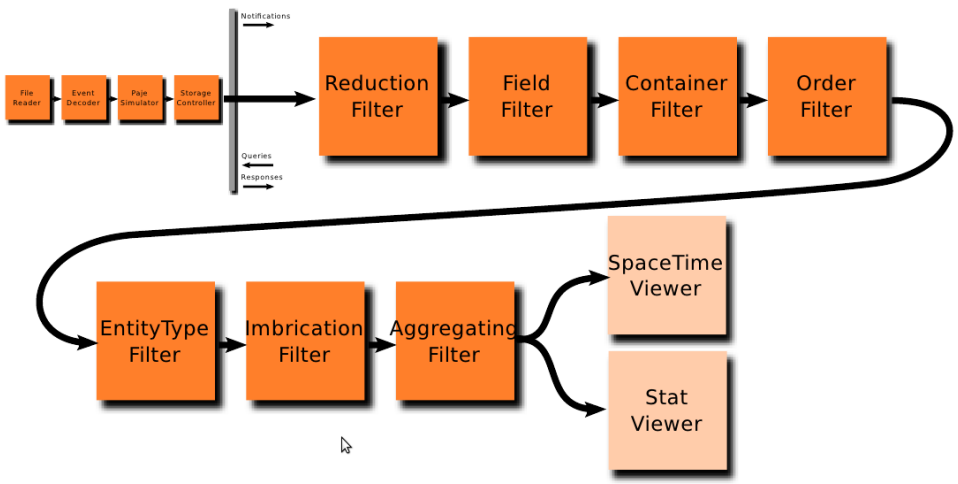
- Développé par LIG puis par l'UFSM (Brésil)
- Outil de visualisation générique conçu pour être :
  - Interactif
    - Possibilité d'intéragir avec les entités monitorées afin d'obtenir des informations à travers la fenêtre de visualisation
  - Scalable
    - Gestion d'un grand nombre d'entités (threads, processes)
  - Extensible
    - Possibilité d'ajouter de nouvelles fonctions/modules, de nouveaux types de traces, de nouvelles techniques de visualisation, etc.

Format de traces Pajé

- Format de trace Pajé, associé à l'outil
- Textuel, pas de sémantique : évenements décrivent le comportement des entités observées
- Types basiques :
  - Containers
  - State
  - Link
  - Variable
  - Event
- Possibilité de décrire le comportement d'un grand nombre de systèmes à partir de ce set d'entités seulement

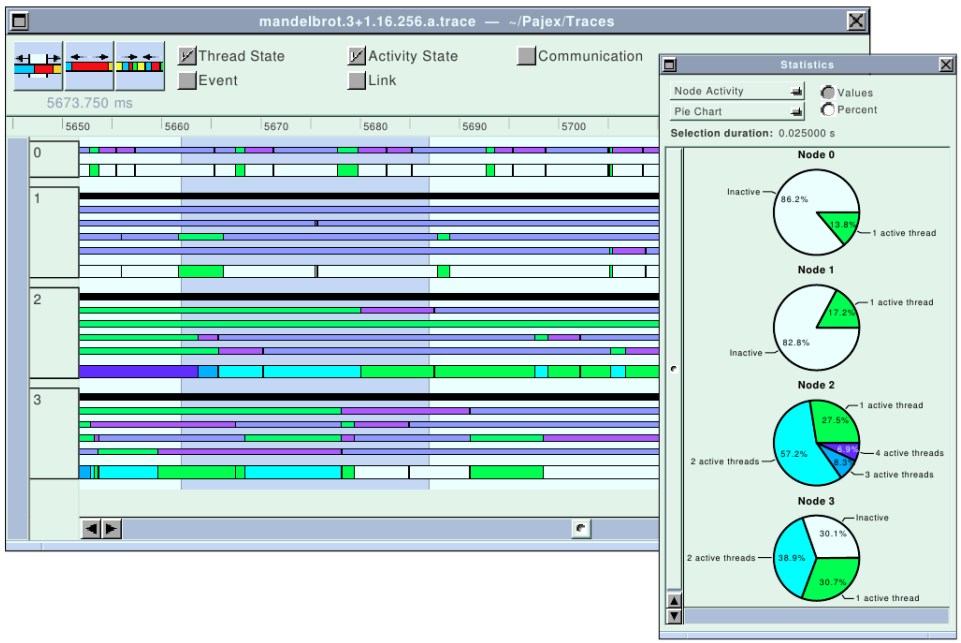
Architecture

- Modules
  - Connectés entre eux par un graphe
  - Possibilité de réorganiser l'ensemble pour ajouter de nouveaux composants
  - Types de composants :
    - Trace file reader
    - Event decoder
    - Simulator
    - Storage controller
    - Filters
    - ...



Deux types de visualisation

- Diagramme de Gantt
  - Le plus utilisé
  - Permet de représenter les intéractions entre les entités
  - Agrégation temporelle
  - Agrégation hiérarchique
- Statistiques
  - Diagrammes circulaires
  - Synthèse de ce qui est représenté dans le diagramme de Gantt



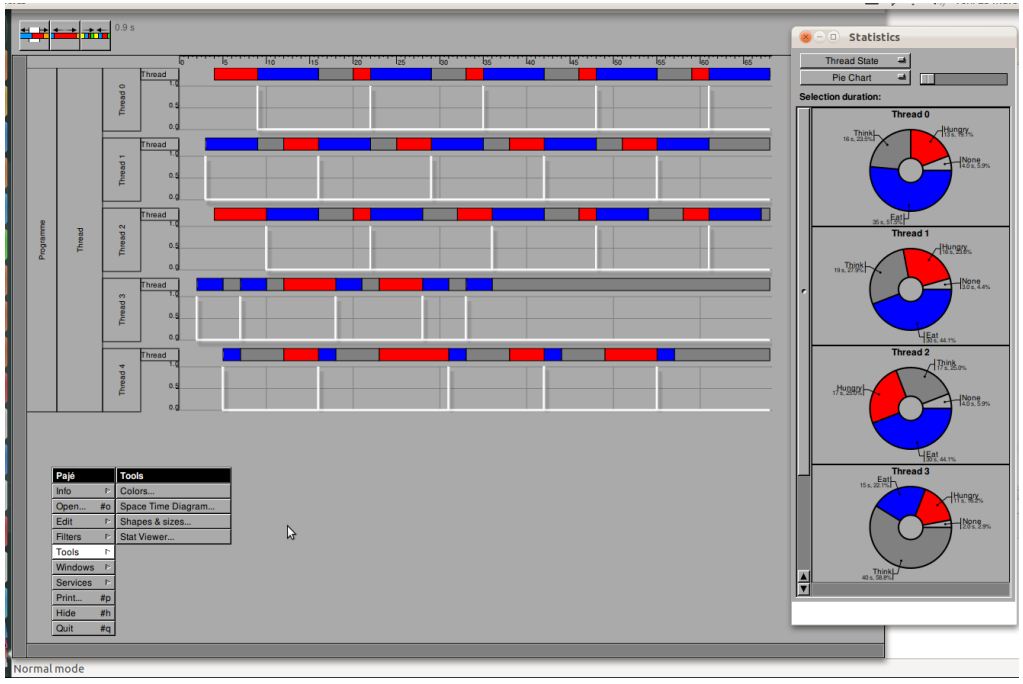
Limites de l'outil

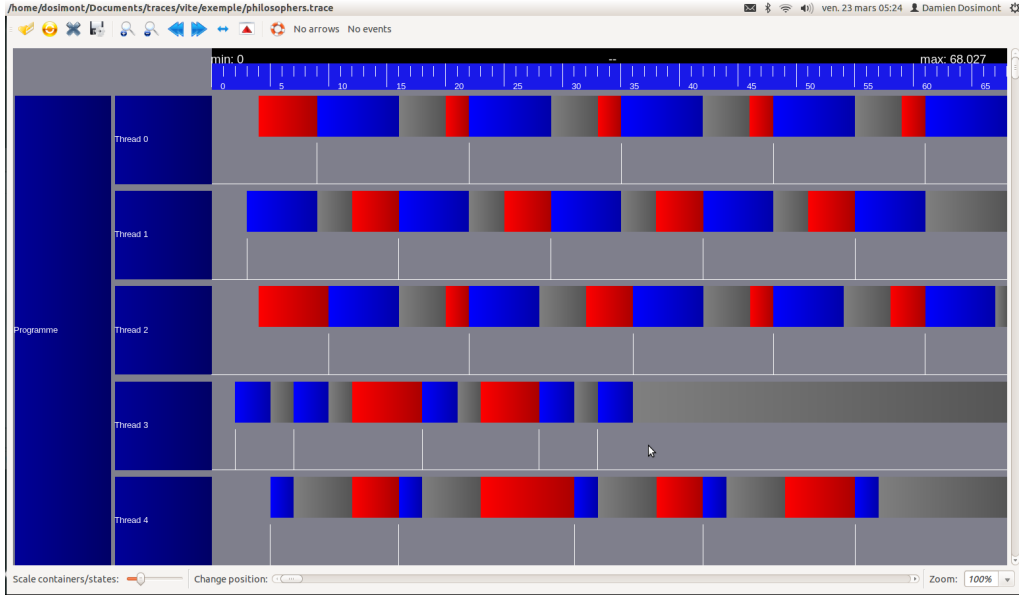
- Repose sur le Framework GNUSTEP
  - ObjectiveC, difficile à maintenir
  - Compilation de GNUSTEP peut être problématique
- Moteur graphique peu puissant
- Agrégation sommaire

Outil de visualisation ViTE

- Rendu visuel proche de Pajé mais moteur OPENGL -> plus fluide
- Ecrit en C++
- Non modulaire comme Pajé : un seul binaire mais possibilité de mettre des plug-ins
- Lit les formats de trace :
  - Pajé
  - Pajé *extended*
  - OTF (Vampir)
  - Tau
- Pas d'agrégation :(

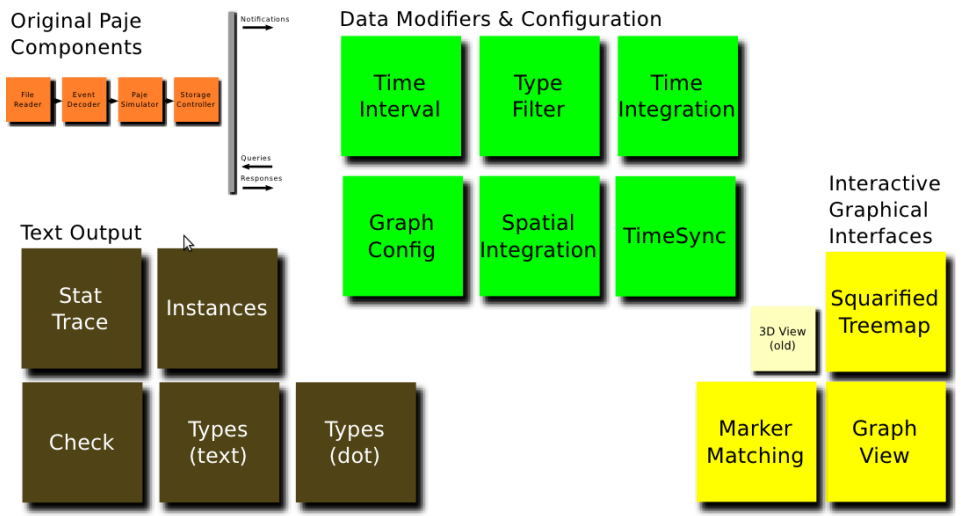
Comparaison Pajé/ViTE





Outil de visualisation Triva

- Développé par Lucas Schnorr au cours de sa thèse
- Utilise le Framework Pajé : modules supplémentaires
- Permet 3 types d'affichage
  - Squarified Treemap
  - Graph View
  - 3D View (plus utilisé actuellement)
- Intégrations spatiale (somme) et temporelle (tranche de temps)
- Objectif : palier les failles du Gantt Chart, en particulier pour la représentation de la topologie

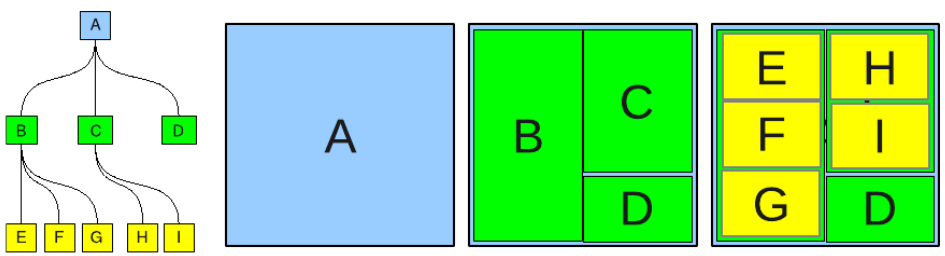


Algorithme de la tranche de temps

- Objectif : définir une structure hiérarchique reflétant comportement programme pour intervalle de temps donné
- Analyste : choisit une tranche de temps : deux marqueurs
  - Longueur peut être changée dynamiquement
- Entités
  - Etats
  - Variables
  - Liens
  - Evenements
- Intégration temporelle afin d'obtenir une valeur **moyenne**, ou, pour les évenements et les liens, somme

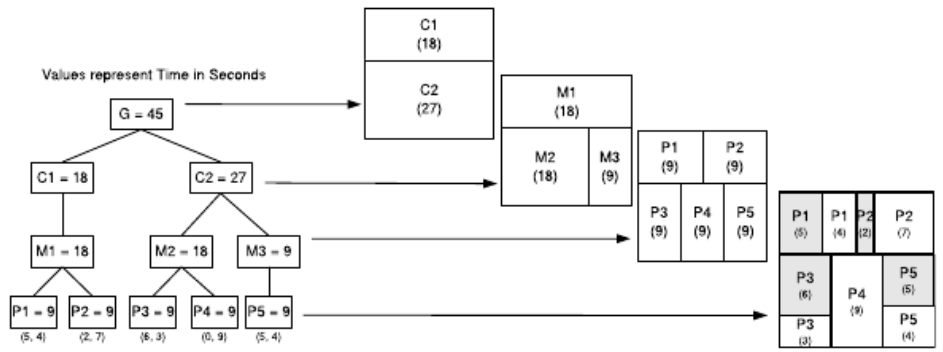
Treemap

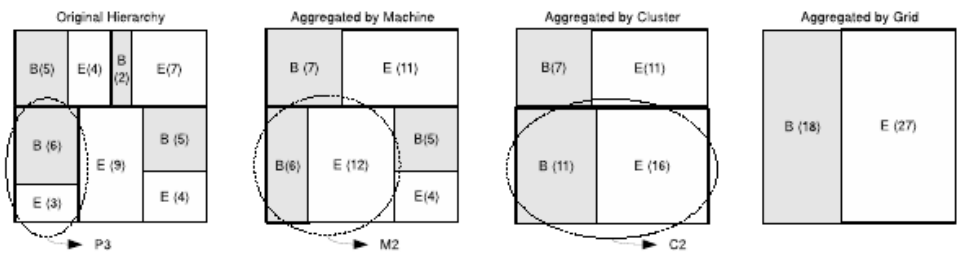
- Résoudre le problème de scalabilité des représentations hiérarchiques
- Algorithme de Space-Filling
  - Divise l'espace dédié pour dessiner la hiérarchie, suivant l'organisation de l'arbre
  - Chaque entité possède une valeur = poids dans la hiérarchie
- Version squarified : ratio longueur/largeur proche de 1



Treemap (2)

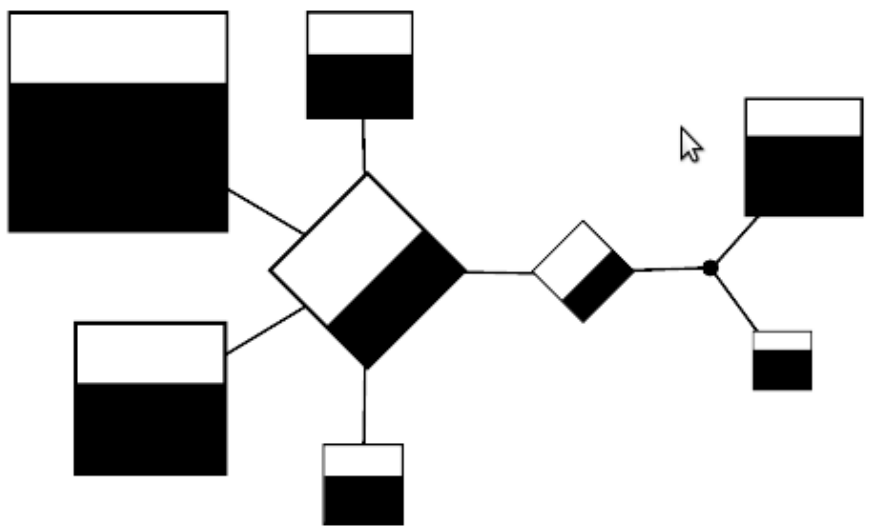
- Organisation des entités sous forme de hiérarchie
- Agrégation Spatiale





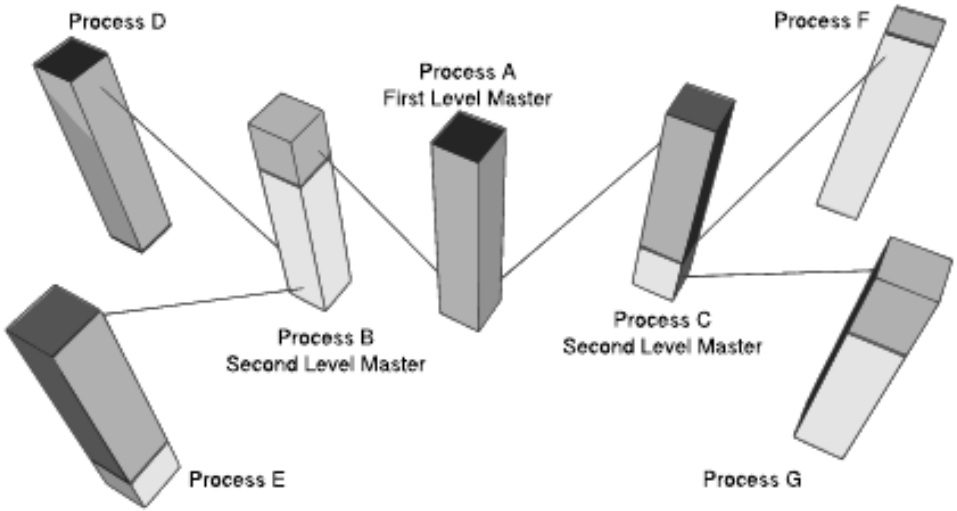
Graphe Topologique

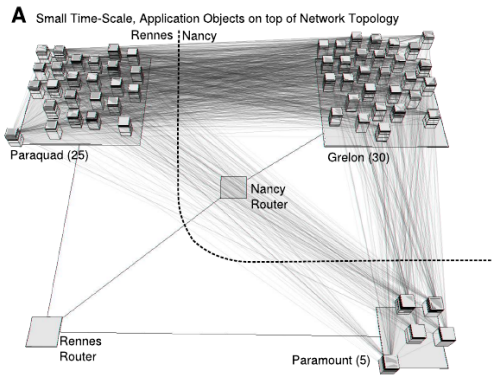
- Introduire une notion de topologie non présente dans treemap
- Basé sur la visualisation de ressources
- Données agrégées en espace et en temps
- Idéal pour identifier les goulots d'étranglement
- Cependant, pas de space-filling

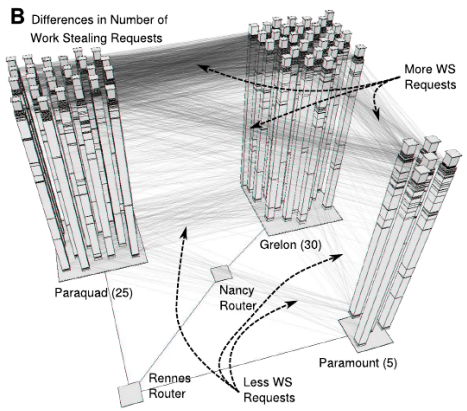


Représentation 3D

- Plus utilisée aujourd'hui







Conclusion

- Multitude d'outils
- Multitude de formats
- Mais les principes restent les mêmes
- Techniques principales : Gantt, statistiques
- Principaux problèmes : représenter un nombre important de données
- Solutions ? Agrégation, représentation topologique, optimisation de l'espace (treemap), techniques plus extravagantes ?