Load Balancing Techniques for Parallel Ray Tracing

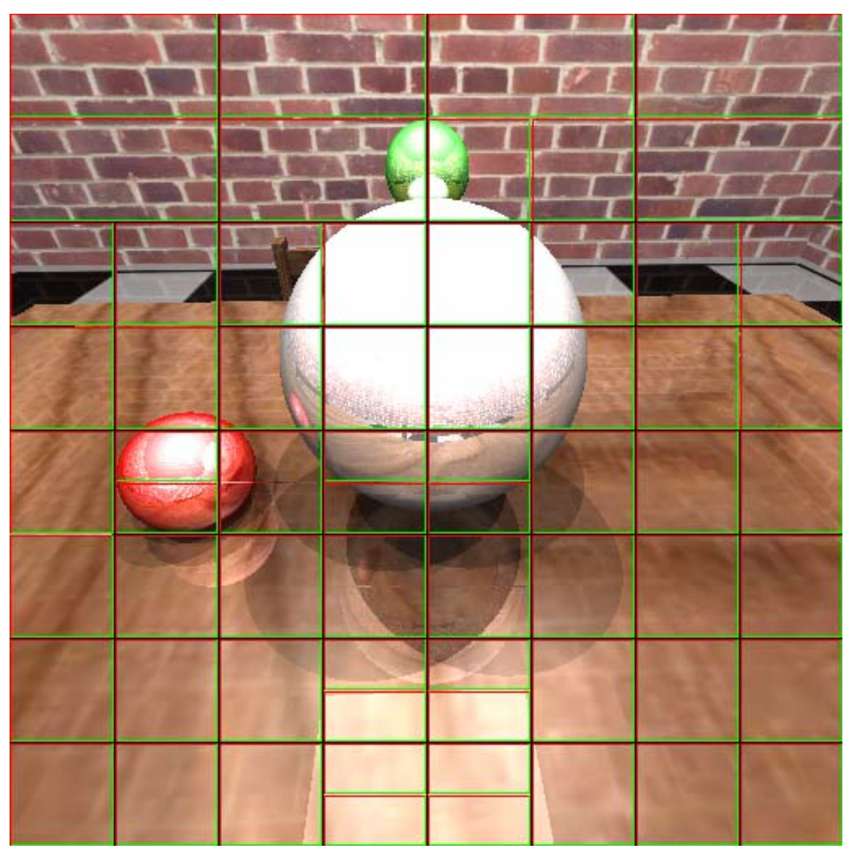
Biagio Cosenza
ISISLab, Dipartimento di Informatica ed Applicazioni
Università degli Studi di Salerno, Italy
email: cosenza@dia.unisa.it

Introduction: Ray Tracing Algorithms

Ray Tracing (RT) algorithms closely model physical light transport by shooting rays into the scene. In the simplest form, RT traces a ray for each pixel, from an eye or view point through the pixel into the scene. Secondary rays can occur e.g. when rays hit reflective or refractive surfaces, for shadows, or for a more accurate simulation of the light transport in the scene, generating a huge computational cost.

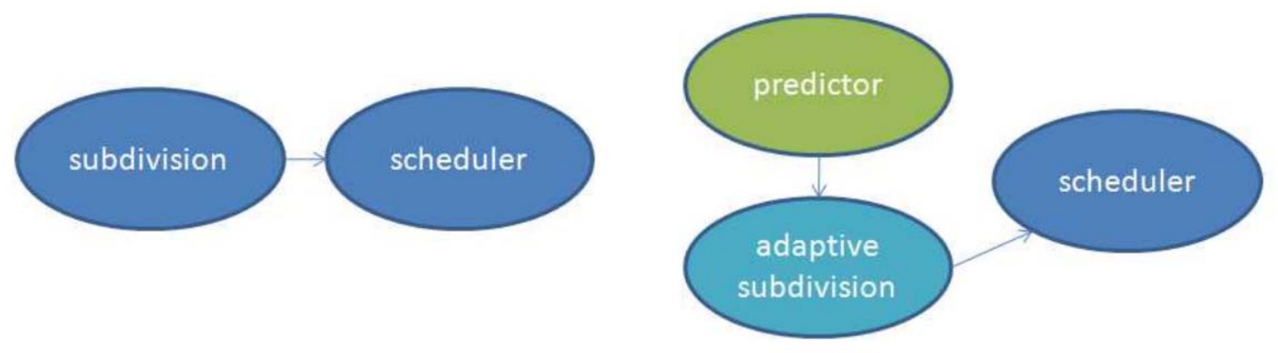
Parallel Ray Tracing

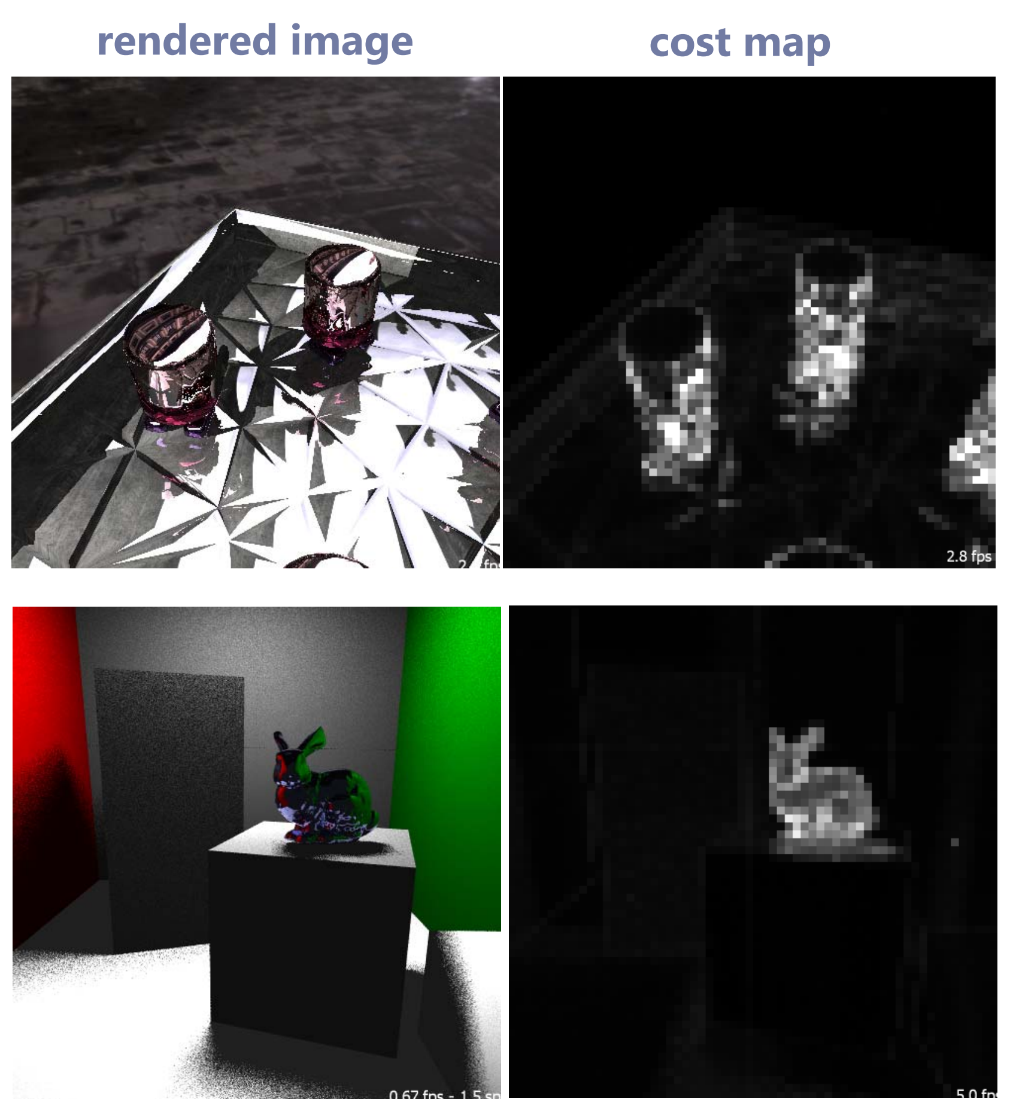
Using complex shading drastically increases the number of rays, requiring a huge computational power. Improving performances to reach interactive rates requires to combine highly optimazed RT implementations with massive amounts of computational power.
Our system implements a **screen-based** Parallel RT: the image is subdivided into tiles distributed among workers, and the scene description is replicated. Each task, or **tile**, can be computed independently from every other task. Our implementation is tailored for distributed memory architectures.



The Load Balancing Problem

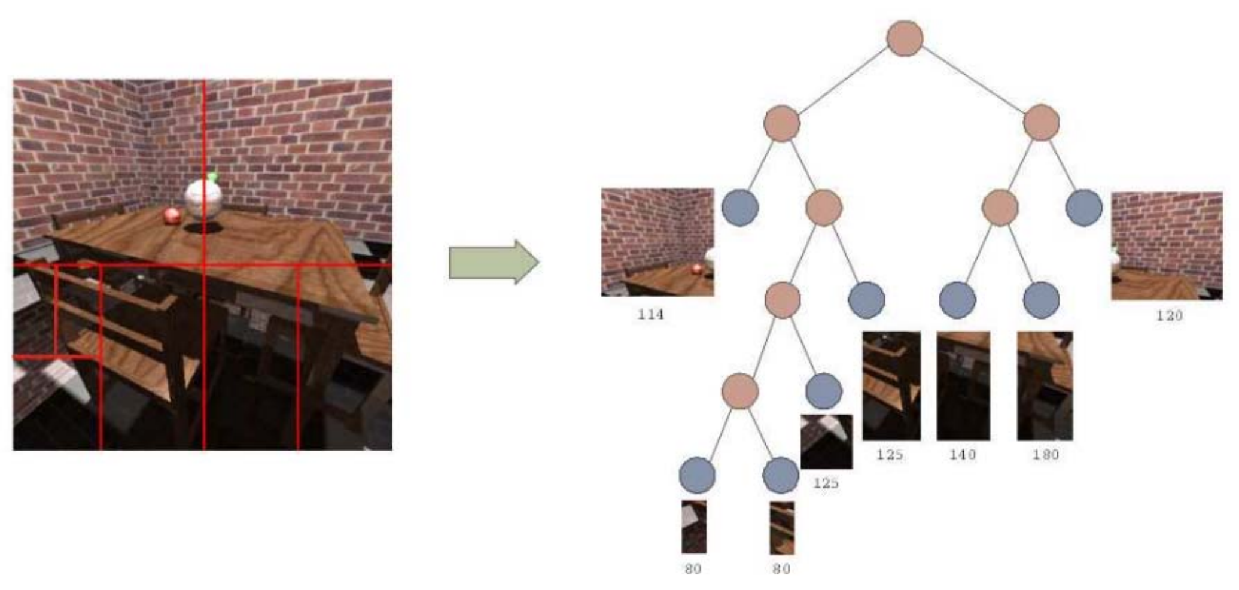
Rendering time of each pixel is influenced by a multitude of factors, such as shading algorithms, texture accesses, acceleration data structure queries.
Uneven load balancing represents a bottleneck to the scalability.
The general approach consists in subdividing the image in equally sized tiles, then using a balancing strategy during the scheduling.
An important choice is the granularity of the subdivision of the images in tiles: a finer granularity, for instance, is good for balancing but needs an higher communication cost.
We consider a new way of managing load balancing based on adaptive subdivision: introducing a **predictor** unit, we are able to estimate the computation time needed by a tile and to arrange a "better tiling".





An Adaptive Subdivision Schema

In order to help balancing we present a balancing schema based on a **Prediction Binary Tree** (PBT).



The Prediction Binary Tree

A PBT stores the current subdivision in tiles in a full binary tree. The root represents the complete image. The two children of an internal node store two halves of the tile assigned to their parent.
The leaves of the tree identify a partition of the image.
The predictor unit assigns an estimate of the expected compute time to each tile. In our implementation, this estimate is done by exploiting temporal coherence with a timing of the tile rendered in the previous frame.

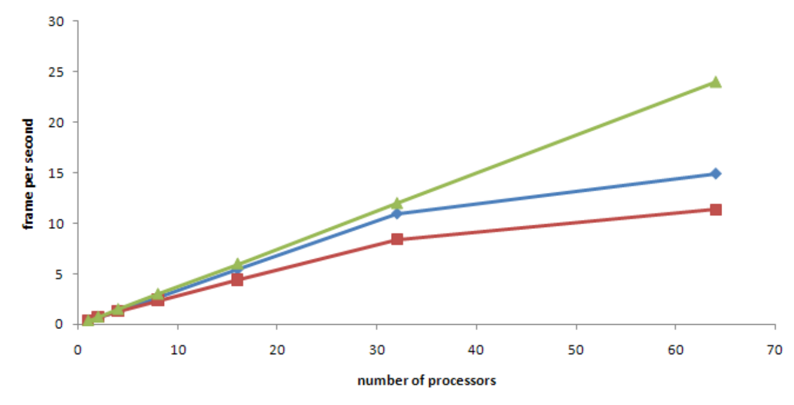
The **update** of the PBT is performed defining a tree cost function and minimizing the cost of the tree at each frame.
The **cost function** measures the unbalance of the tree.

$$\sigma_T^2 = \frac{1}{m} \sum_{\ell \in L(T)} (e(\ell) - \mu_T)^2$$

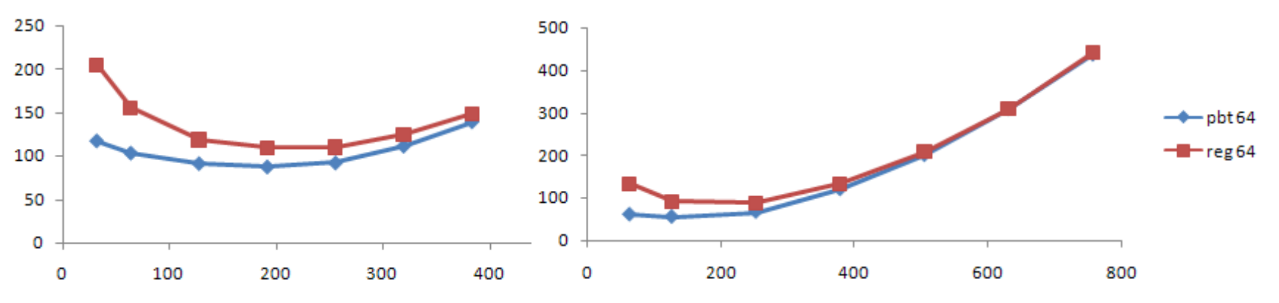
The update produces a new tree by performing several split&merge operation: it splits the tile whose estimated load is "high", and merge two tiles (stored at sibling nodes) whose (combined) estimated load is "small".
The update produces a minimum cost, minimum unbalancing PBT.

Results

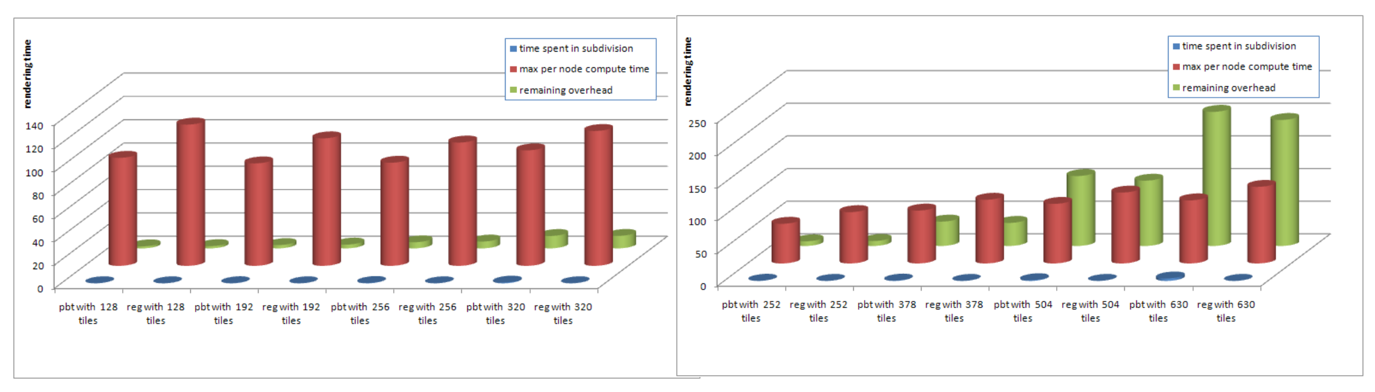
Our strategy provides a better scalability compared with a regular subdivision schema.



The optimal granularity for different number of processors needs a lower number of tiles, compared with a regular subdivision schema.



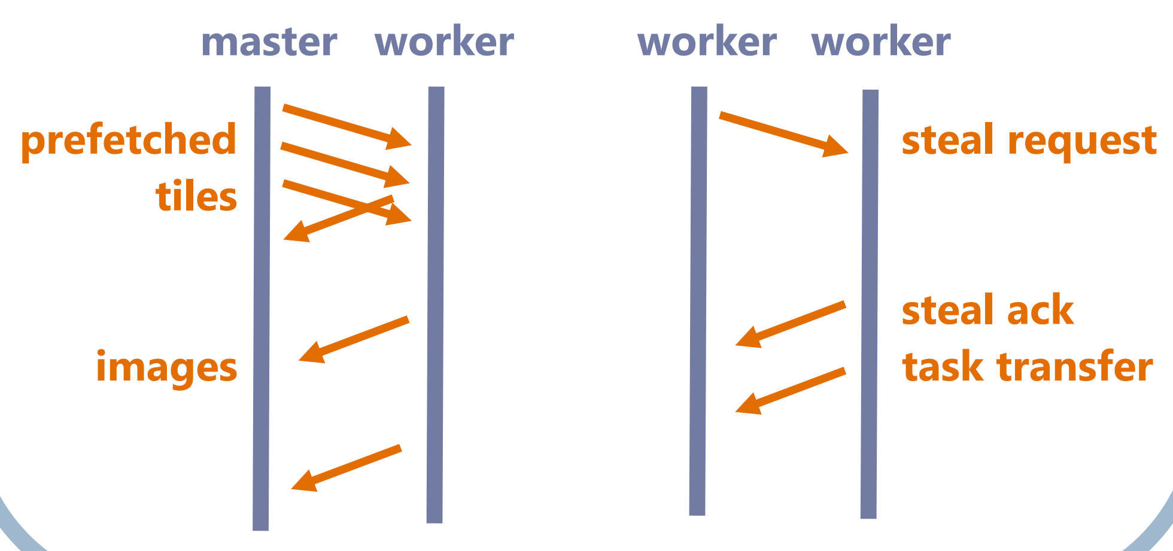
Moreover the overhead introduced by the adaptive subdivision schema is small compared with the gain in balancing tree and, hence, assures better performance.



Integrating this technique with known load balancing strategies is easy and effective. The technique can be extended to more general mesh-like computations.

Dynamic Load Balancing

Adaptive subdivision heuristics are compatible with other commonly used balancing techniques.
We showed that it can be used effectively in a system using task prefetching and dynamic distributed load balancing with work stealing.



References

"Load Balancing in Mesh-like Computations using Prediction Binary Trees"
B. Cosenza, G. Cordasco, R. De Chiara, U. Erra and V. Scarano
In Proc. of 7th International Symposium on Parallel and Distributed Computing (ISPDC 2008). July 1-5, Krakow, Poland.

Collaborations and credits

This is a joint work with G. Cordasco, R. De Chiara and V. Scarano of the D.I.A. at the Università degli Studi di Salerno di Salerno and U. Erra of the D.M.I. at the Università degli Studi della Basilicata.
The author would like to thank Thomas Ertl and Carsten Dachsbacher of the Institut für Visualisierung und Interaktive Systeme at the Stuttgart Universität for supporting the project and helpful discussions.
Tests have been performed on a NEC Xeon EM64T cluster available at HLRS High Performance Computing Center at the Stuttgart Universität.

HPC-Europa++ 2008
Pan-European Research Infrastructure on High Performance Computing

Dipartimento di Informatica e Applicazioni R. M. Capocelli
Università degli Studi di Salerno
ISISLab

This work was carried out under the HPC-EUROPA++ project (project number: 211437), with the support of the European Community, Research Infrastructure Action of the FP7.