Optimal Orthogonal Tiling

R. Andonov, S. Rajopadhye & N. Yanev

Univ. Valenciennes, IRISA, Rennes & Bulgarian Academy, Sofia

Outline

- Introduction and related work
- Problem formulation
  - Abstract model building
  - Machine and program specific model
  - Simplifications and special cases
- Solution of the optimization problem
- Experimental results
- Conclusions and perspectives

Problem Formulation

Execute an n-dimensional uniform dependence loop in SPMD mode on DMM of P processors,

- loop visits all points in rectangular parallelepiped domain *Dom* $\subset Z_{+}^{n}$
- f is a single-valued function
- dependencies: $\mathbf{D} = [\mathbf{d_1}, \ldots, \mathbf{d_m}]$ is full rank & $\mathbf{D} \geq 0$.
- So $\mathbf{I}$ is a valid tiling !

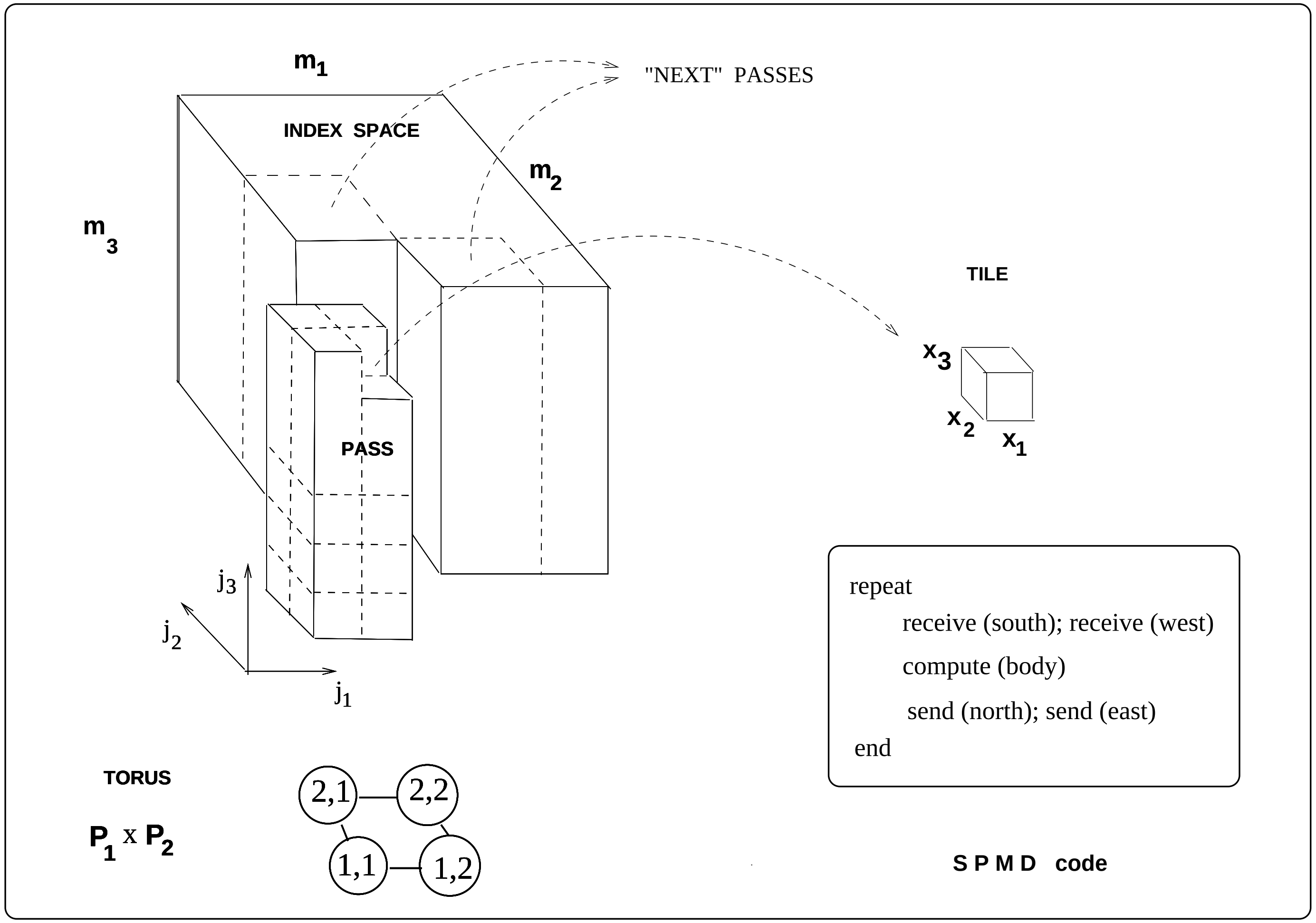
Problem: What is the tile size that minimizes the total execution time of the program

Related Work (non exhaustive)

- [Wol 87], [IT 88] Definition and foundations of tiling
- [BDRR 94] Optimal tile shape
- [HKT 94] Optimal granularity (2D)
- [PSCB 94] Improved model (2D)
- [OSKO 95] Optimal granularity (2 & n-D)
- [AR 95], [ABR 96], [AY 97] Analytical solution (2D & 3D)

- Here: Closed form solution for the general (n-D) case

Abstract model building



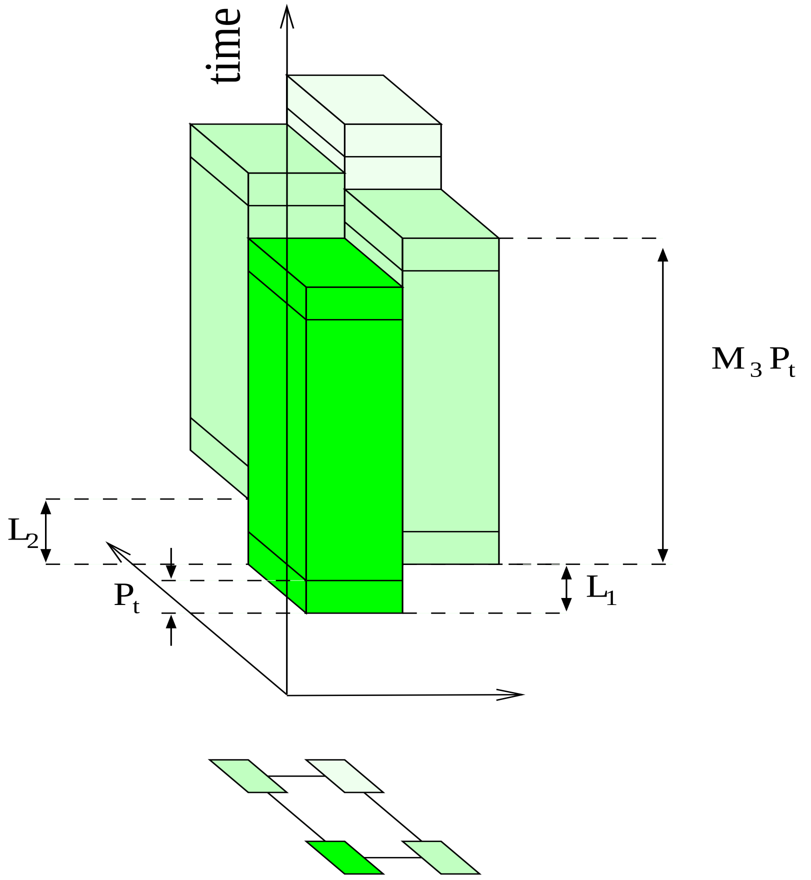
- If $m_i = \frac{N_i}{x_i}$, the tile graph is a $m_1 \times \ldots m_n$, n-dimensional regular graph with $\mathbf{I}$ as dependence matrix

- Map this to a k-dimensional $(k < n)$ torus of $p_1 \times p_2 \ldots \times p_k$ processors à la systolic space-time mapping (tiles are atomic)

- Two abstract parameters — tile period, $\mathcal{P}_t$ & and (inter) tile latency, $\mathcal{L}_i$

- Use simple counting arguments — abstract running time

Abstract model building: a single pass



ONE PASS AFTER SPACE-TIME MAPPING

- pass period $P_p = \frac{N_n}{x_n}\mathcal{P}_t$

- pass time $T_p = P_p + (p_1 - 1)L_1 + (p_2 - 1)L_2 = \sum_{i=1}^{k} \tilde{p}_i \mathcal{L}_i$

- # passes $= \dfrac{\text{total \# tiles}}{\text{\# tiles in 1 pass}} = \dfrac{\prod \frac{N_i}{x_i}}{P\frac{N_n}{x_n}}$

- total time $T = (\#\text{ passes} - 1)P_p + T_p = \frac{W\mathcal{P}_t}{Pv} + \sum_{i=1}^{k} \tilde{p}_i \mathcal{L}_i$

Machine specific model (the parameters)

- β_s/β_r: overhead of **`send`** (**`receive`**)
- τ_c: time to copy from/to user system memory
- τ_t: transmission rate
- α: computation time (per iteration)
- tile volume: $v(\mathbf{x}) = \prod_{i=1}^{n} x_i$
- total computation volume $W = \prod_{i=1}^{n} N_i$
- # of processors $P = \prod_{i=1}^{k} p_k$
- $\widetilde{p_i} = p_i - 1$
- $\widetilde{P_k} = \sum_{i=1}^{k} \widetilde{p_i}$

Nonlinear integer programming problem

- Period: $\mathcal{P}_t = k(\beta_r + \beta_s) + \left(2\tau_c v \sum_{i=1}^{k} \frac{1}{x_i}\right) + \alpha v$

- Latency: $\mathcal{L}_i = k\beta_s + \left(2\tau_c v \sum_{i=1}^{k} \frac{1}{x_i}\right) + \alpha v + \tau_t \frac{v}{x_i}$

- Total running time:

$$\begin{aligned} T_k(\vec{x}) &= \alpha v \widetilde{P_k} + 2\tau_c \widetilde{P_k} v \sum_{i=1}^{k} \frac{1}{x_i} + \tau_t v \sum_{i=1}^{k} \frac{\widetilde{p_i}}{x_i} + \frac{2\tau_c W}{P} \sum_{i=1}^{k} \frac{1}{x_i} + \\ &+ \; k(\beta_r + \beta_s)\frac{W}{Pv} + k\beta_s \widetilde{P_k} + \frac{\alpha W}{P} \end{aligned}$$

Recap slide

- β_s/β_r: `send/receive` overhead
- τ_c: memory copy time
- τ_t: transmission rate
- α: computation time (per iteration)
- $v(\mathbf{x}) = \prod_{i=1}^{n} x_i$, $W = \prod_{i=1}^{n} N_i$, $P = \prod_{i=1}^{k} p_k$, $\widetilde{p_i} = p_i - 1$, $\widetilde{P_k} = \sum_{i=1}^{k} \widetilde{p_i}$
- Total running time:

$$T_k(\vec{x}) = \alpha v \widetilde{P_k} + 2\tau_c \widetilde{P_k} v \sum_{i=1}^{k} \frac{1}{x_i} + \tau_t v \sum_{i=1}^{k} \frac{\widetilde{p_i}}{x_i} + \frac{2\tau_c W}{P} \sum_{i=1}^{k} \frac{1}{x_i} + k(\beta_r + \beta_s)\frac{W}{Pv} + k\beta_s \widetilde{P_k} + \frac{\alpha W}{P}$$

IRISA

Simplifying assumptions (I): HKT model

Set $\tau_c = \tau_t = 0$, (and typically) $\beta_s = \beta_r = \beta$ ("startup cost")

$$T_k(\mathbf{x}) = \frac{\alpha W}{P} + k(\beta_r + \beta_s)\frac{W}{Pv} + (\alpha v + k\beta_s)\widetilde{P_k}$$

Solution: (all x_i with this tile volume are equally good)

- Optimal tile volume $\tilde{v} = \sqrt{\frac{k(\beta_r+\beta_s)W}{P\alpha\widetilde{P_k}}}$
- Optimal time $\widetilde{T_k} = \frac{\alpha W}{P} + k\beta_s\widetilde{P_k} + 2\sqrt{\frac{k\alpha(\beta_r+\beta_s)W\widetilde{P_k}}{P}}$

Optimal Architecture (HKT model)

If we let k vary, what is its optimal value?

Let $k^* = \arg\min_k T_k(\tilde{v}) = 0.625 \ln P$

Result: The optimal architecture is a balanced torus (all sides equal) of

- $n-1$ dimensions, if $n \leq \lceil k^* \rceil$
- $\lceil k^* \rceil$ (resp $\lfloor k^* \rfloor$) dimensions if $n > \lceil k^* \rceil$

Simplifying assumptions (II:) The BSP model

Set $\tau_t = \beta_r = 0, \beta_s = \frac{\beta}{k}$ (β is the BSP synchronization cost)

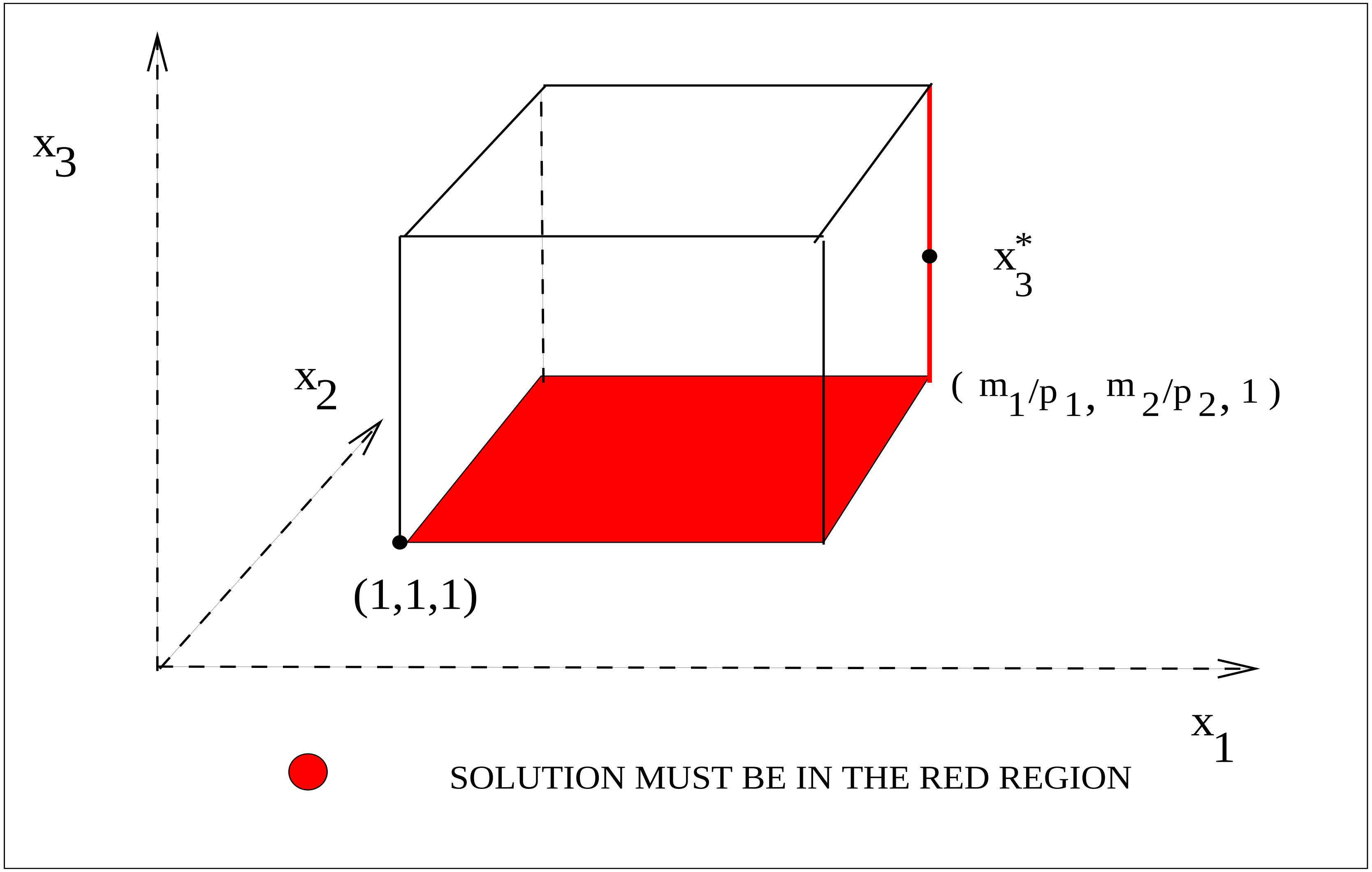
Total running time:

$$T_k(\mathbf{x}) = \frac{\alpha W}{P} + \frac{\beta W}{Pv} + (\alpha v + \beta)\widetilde{P_k} + 2\tau_c \left(\widetilde{P_k} v + \frac{W}{P}\right) \sum_{i=1}^{k} \frac{1}{x_i}$$

Solution: The minimum of $T_k(\mathbf{x})$ is attained at:

either	x_i	$=$	$\frac{N_i}{p_i}$	for $i = 1 \ldots k$	Case I
or	x_i	$=$	1	for $i = k+1 \ldots n$	Case II

Feasible solution space (3-d case)



Case I: One pass solution

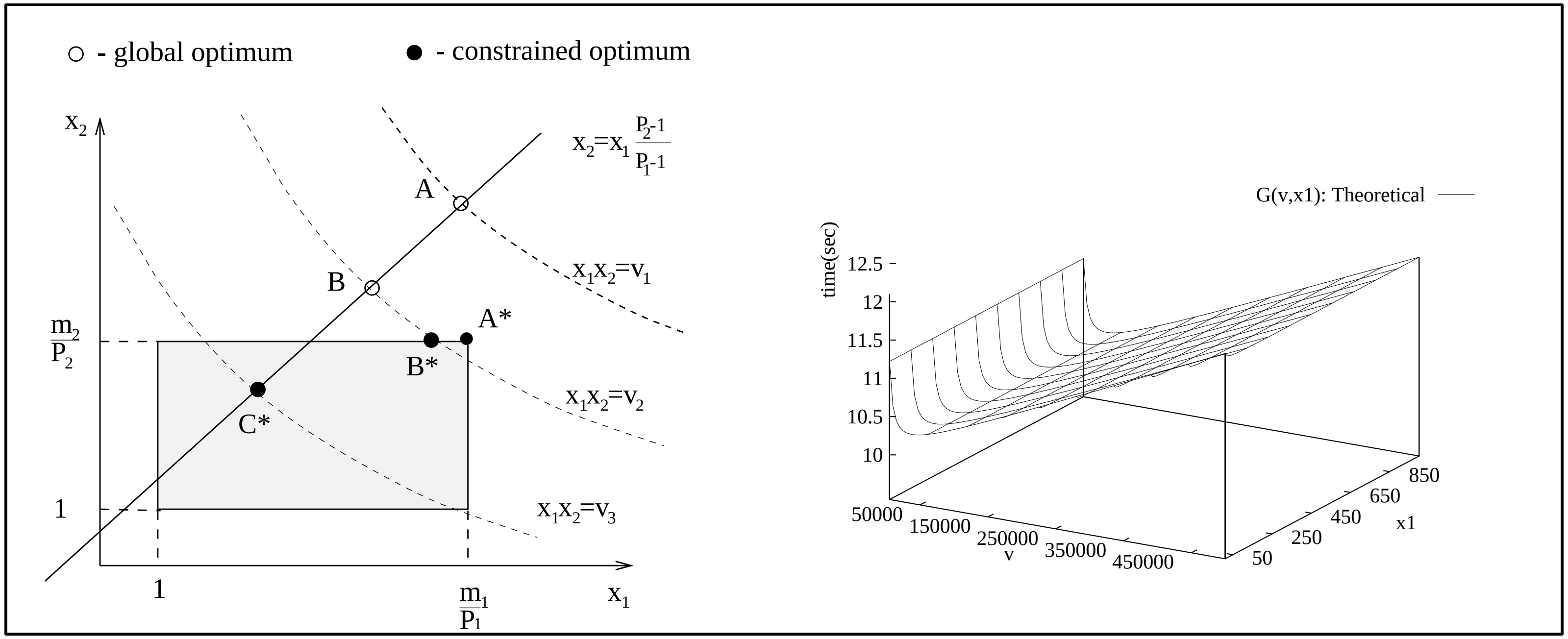
Result: The optimal volume is given by (similar to HKT model):

- $\tilde{v} = \sqrt{\dfrac{\beta W}{P \omega}}$

- $\tilde{T} = \omega \dfrac{W}{P} + \beta \widetilde{P_k} + 2\sqrt{\dfrac{\beta \omega W}{P}}$

where $\omega = (\alpha + 2\tau_c \widetilde{N_k})\widetilde{P_k}$

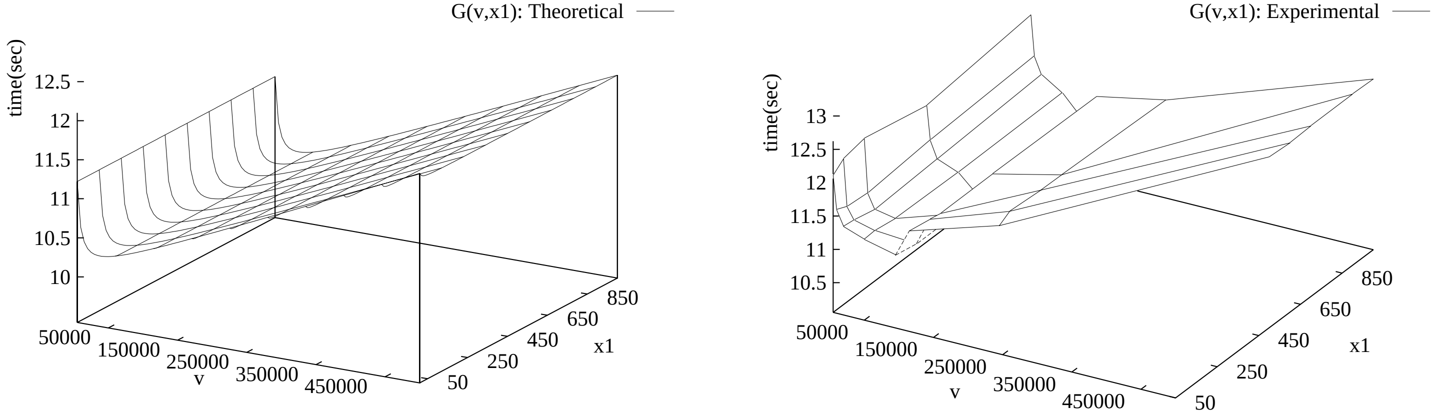
Case II: multiple pass solution



$$\tilde{v} \approx \sqrt{\frac{\gamma W}{P}} \quad \text{where } \gamma = (\beta + 2\tau_c)/((2\tau_c(k-1)+\alpha)\widetilde{P_k})$$

$$x_i^* = \sqrt[k]{\tilde{v}} \text{ for each } i = 1 \ldots k$$

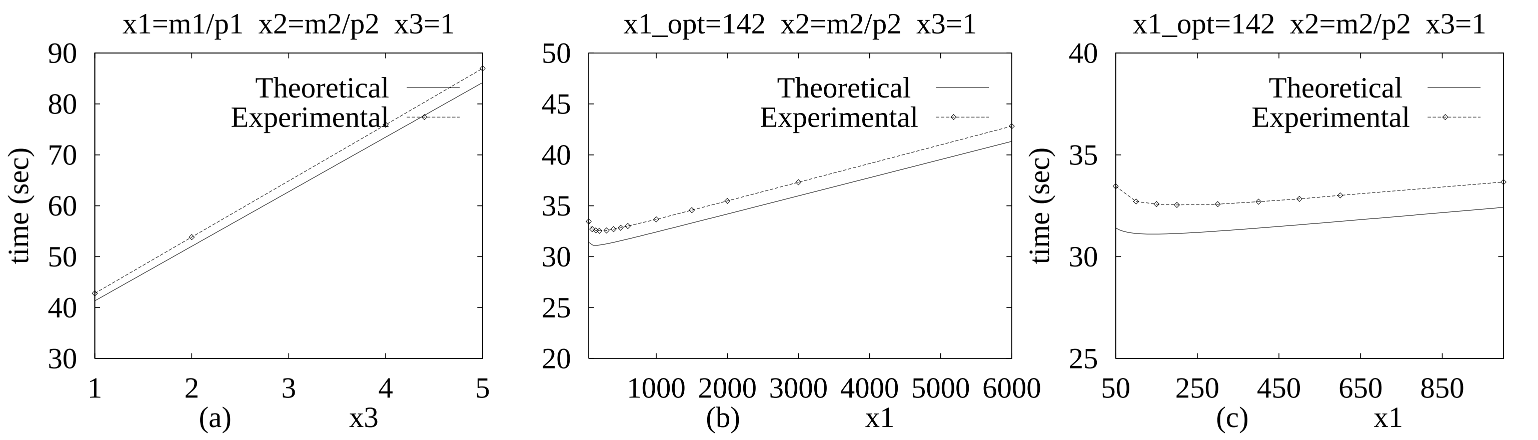
Experimental Validation



$$N_1 = 5000, N_2 = 5000, N_3 = 20, p_1 = 5, p_2 = 5$$

Intel paragon (53 processors), communication with NX library (`send/receive`) system calls. Application: simple loop with canonic dependencies (sequence comparison)

Experimental Results (3d case)



$$N_1 = 30000, N_2 = 2000, N_3 = 20, p_1 = 5, p_2 = 4$$

Example where the multiple pass solution (42 passes with $142 \times 500 \times 1$ size tiles) is optimal.

Conclusions

The n-D optimal tile sizing problem is formulated as a discrete non-linear optimization problem using

- an abstract model
- specific machine model
- the cost function subsumes most other models for DMMs

Analytical solution $\Rightarrow$ Orthogonal tiling is easy!

Open Problems

- Extension to the non-orthogonal case

- More experimental validation on a number of target machines

- Modeling caches