Sharing Incentives and Fair Division for Multiprocessors

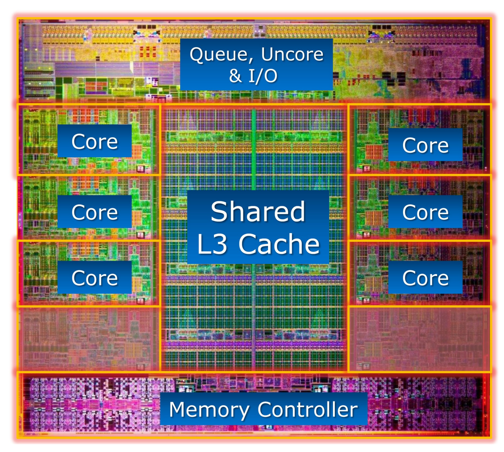
Seyed Majid Zahedi and Benjamin C. Lee (Duke University)

1. Case for Sharing

Big Servers
- Hardware is under-utilized
- Sharing amortized power

Heterogeneous Users
- Tasks are diverse
- Users are complementary
- Users prefer flexibility

Sharing Challenges
- Allocate multiple resources
- Ensure fairness



Intel Sandy Bridge E die [www.anandtech.com]

2. Conventional Wisdom in Computer Architecture

Users must share
- Overlooks strategic behavior

Fairness policy is equal slowdown
- Fails to encourage envious users to share

Heuristic mechanisms enforce equal slowdown
- Fail to give provable guarantees

3. Resource Elasticity Fairness (REF)

REF is an allocation mechanism that guarantees game-theoretic desiderata for shared chip multiprocessors

Sharing Incentives
- Users perform no worse than under equal division

Envy-Free
- No user envies another's allocation

Pareto-Efficient
- No other allocation improves utility without harming others

Strategy-Proof
- No user benefits from lying

4. Cobb-Douglas Utility

$$u(x) = \prod_{r=1}^{R} x_r^{\alpha_r}$$

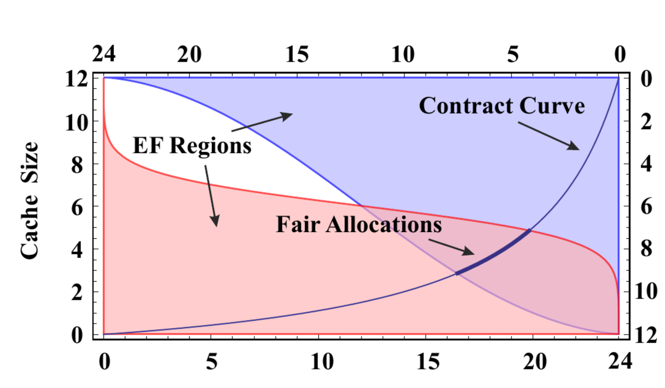
u	utility (e.g., performance)
x_r	allocation for resource r (e.g., cache size)
α_r	elasticity for resource r

- Cobb-Douglas fits preferences in computer architecture
- Exponents model diminishing marginal returns
- Products model substitution effects

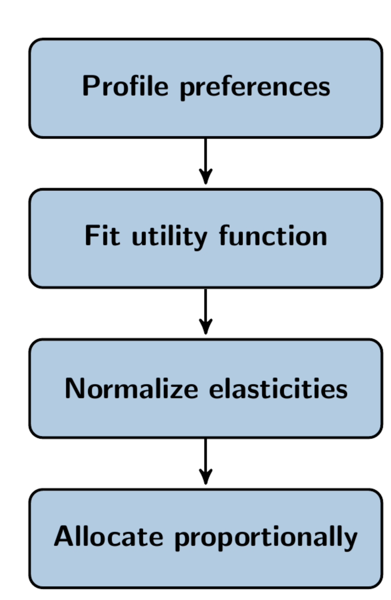
5. Fairness for a Real Example

"If an allocation is both equitable and Pareto efficient,... it is fair."
[Varian, Journal of Economic Theory (1974)]

$$u_1 = x_1^{0.6} y_1^{0.4} \qquad u_2 = x_2^{0.2} y_2^{0.8}$$



6. Mechanism for Resource Elasticity Fairness



$$u_1 = x_1^{0.6} y_1^{0.4} \qquad u_2 = x_2^{0.2} y_2^{0.8}$$

$$x_1 = \left(\frac{0.6}{0.6+0.2}\right) \times 24 = 18\text{GB/s}$$

$$x_2 = \left(\frac{0.2}{0.6+0.2}\right) \times 24 = 6\text{GB/s}$$

7. Dynamic Resource Allocation

Limitation of prior mechanisms
- Fairness is guaranteed within a single epoch
- Fairness is achieved by sacrificing performance

Dynamic system behavior
- System is defined by sequence of epochs
- Users may enter and leave the system
- Users demand for resources may change
- Resource availability may change

8. Allocation of Heterogeneous Cores with Token System

Virtual currency is used to allocate big cores
- Users send a request for big core
- Users spend one token for big cores
- Users receive redistributed tokens

Users are given the same number of tokens
- Mechanism guarantees big cores for certain number of epochs

Threshold strategy is used to optimize performance
- Users request big cores, if small core cannot provide comparable performance