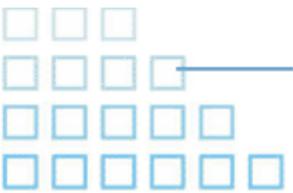


服务计算

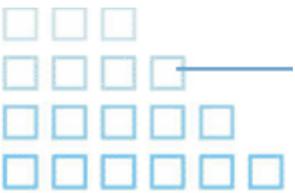
Service Computing

王旭

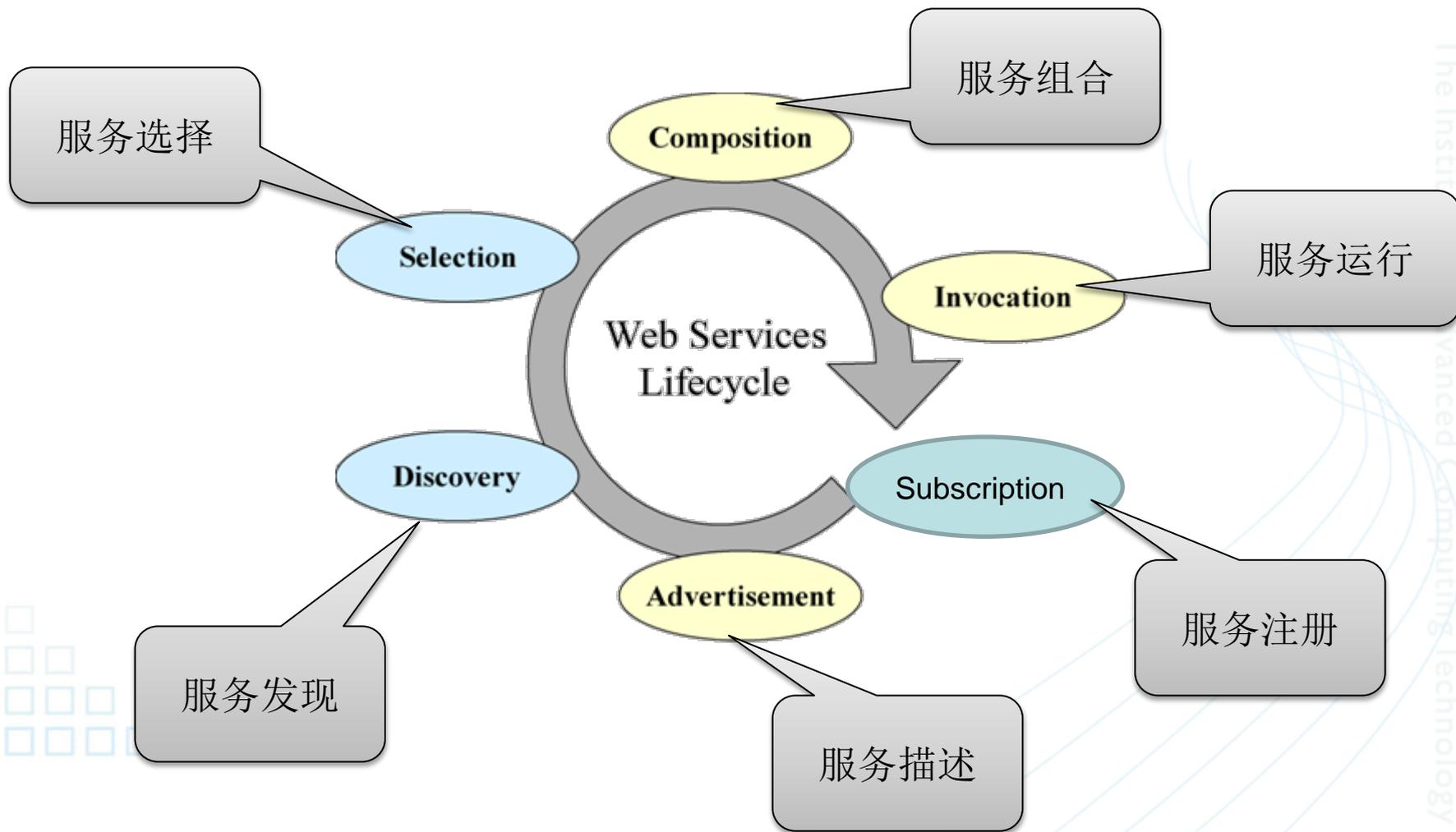
wangxu@buaa.edu.cn



五、服务选择与推荐



服务Lifecycle



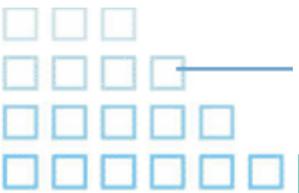
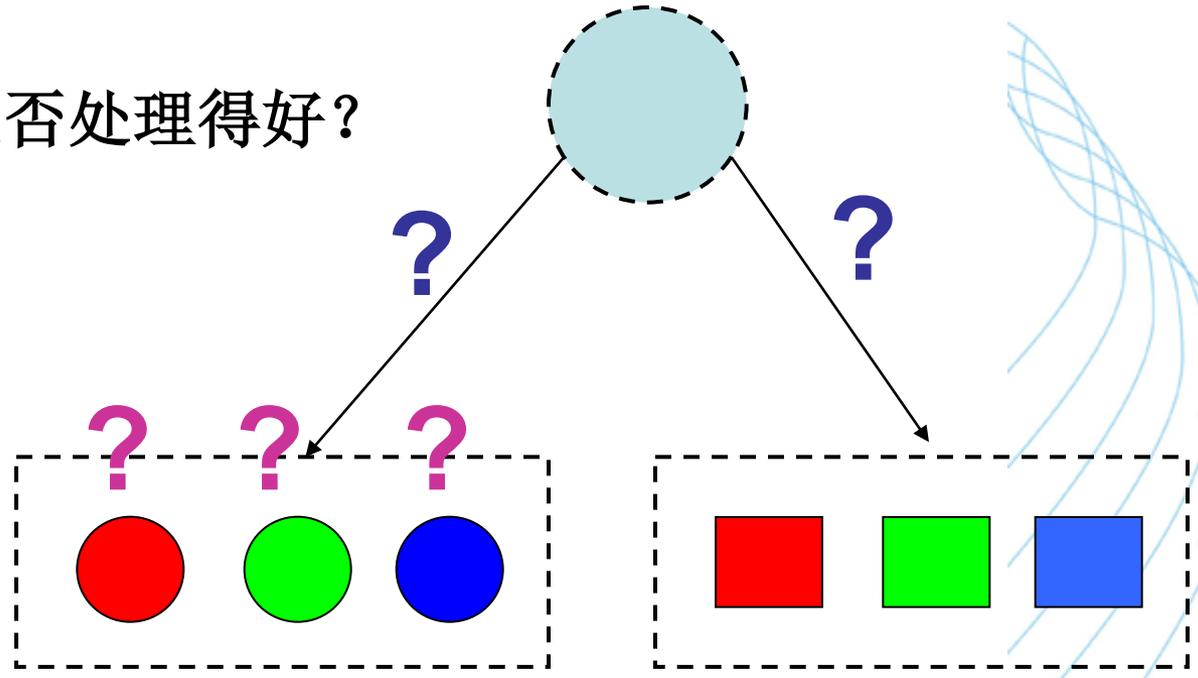
服务选择问题

- 两个方面
 - 功能性：能否完成处理任务？ **服务发现**

- 非功能性：能否处理得好？

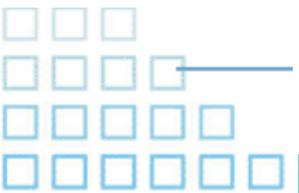
- 可用性
- 响应时间
- 吞吐量
- 花费
- 可靠性

QoS属性



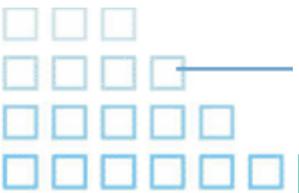
基于非功能性的选择

- 前提：基于服务发现，已经得到功能性可满足的服务集合，如何从中选择最优的**QoS**属性服务？
- 挑战性
 - **QoS**属性的多维性
 - 如何同时兼顾多个属性？
 - 针对不同用户，**QoS**属性不完全
 - 如何收集和预测缺失的属性
 - **QoS**属性是动态变化的
 - 如何建立准确的预测模型？



基于非功能性的选择

- 前提：基于服务发现，已经得到功能性可满足的服务集合，如何从中选择最优的**QoS**属性服务？
- 挑战性
 - **QoS属性的多维性**
 - 如何同时兼顾多个属性？
 - 针对不同用户，**QoS属性不完全**
 - 如何收集和预测缺失的属性
 - **QoS属性是动态变化的**
 - 如何建立准确的预测模型？



多维属性的处理

- 加权计算

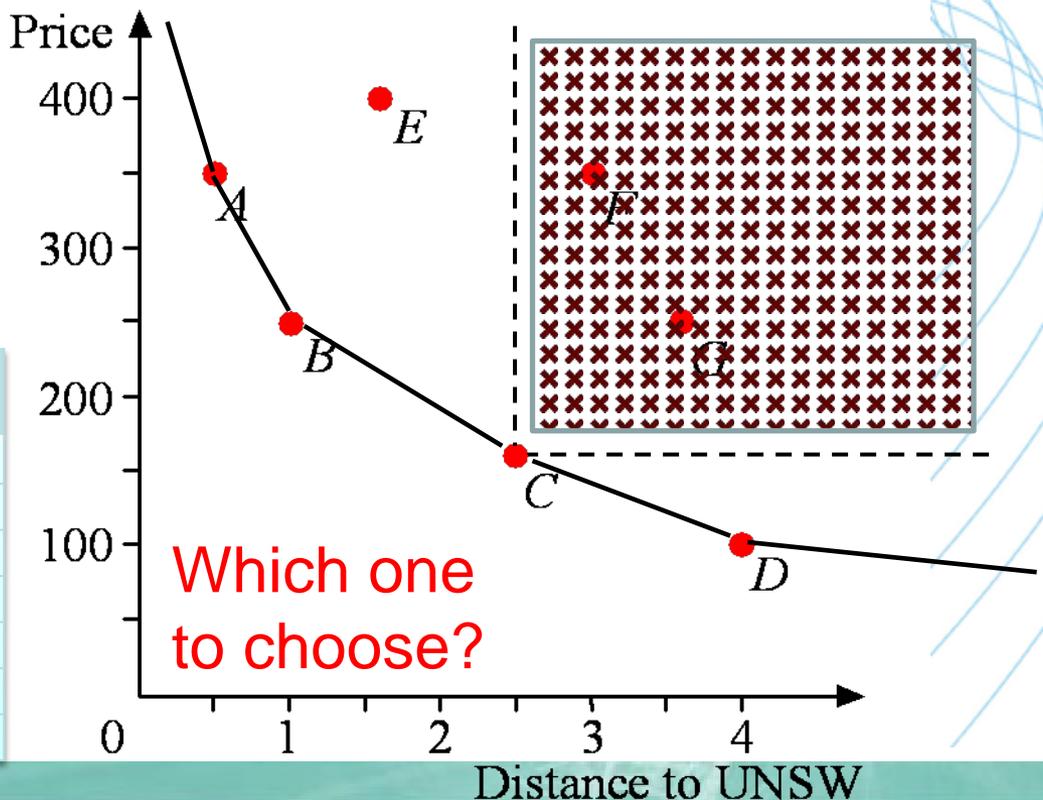
$$A = w_1 * A_1 + w_2 * A_2 + \dots + w_n * A_n$$

- 权重的确定：用户指定、机器学习

- 多目标优化问题

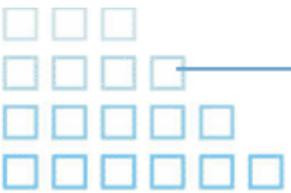
- 例如skyline query

Room	Price	Distance to UNSW
A	\$350	0.5 km
B	\$250	1 km
C	\$150	2.5 km
D	\$100	4 km
E	\$400	1.5 km
F	\$350	3 km
G	\$250	3.5 km



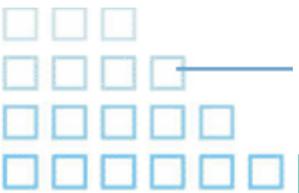
Skyline query的算法

- **Divide and Conquer**
- **Block Nested Loops**
- **Sort Filter Skyline**
- **Linear Elimination Sort for Skyline**
- **Bitmap**
- **Index**
- **Nearest Neighbor**
- **Branch and Bound Skyline**



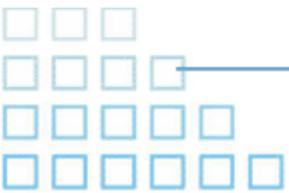
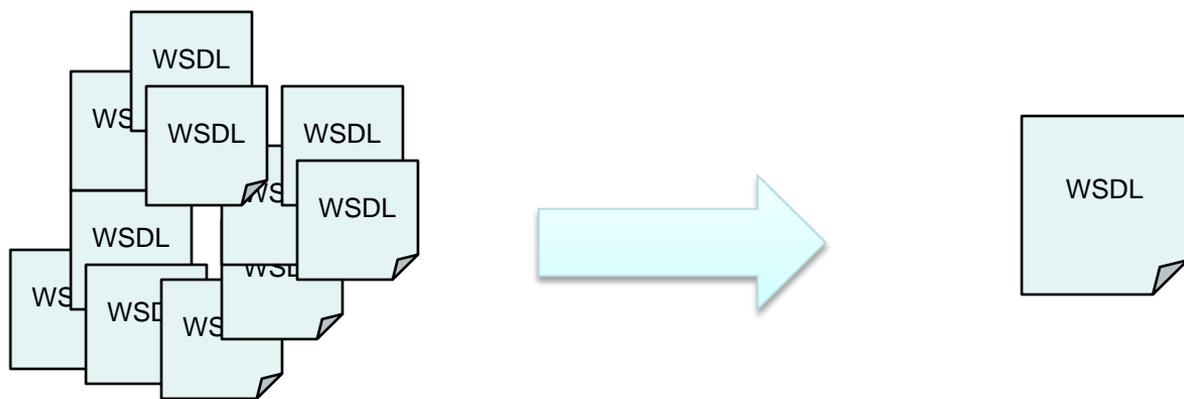
基于非功能性的选择

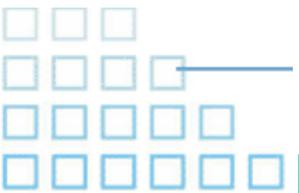
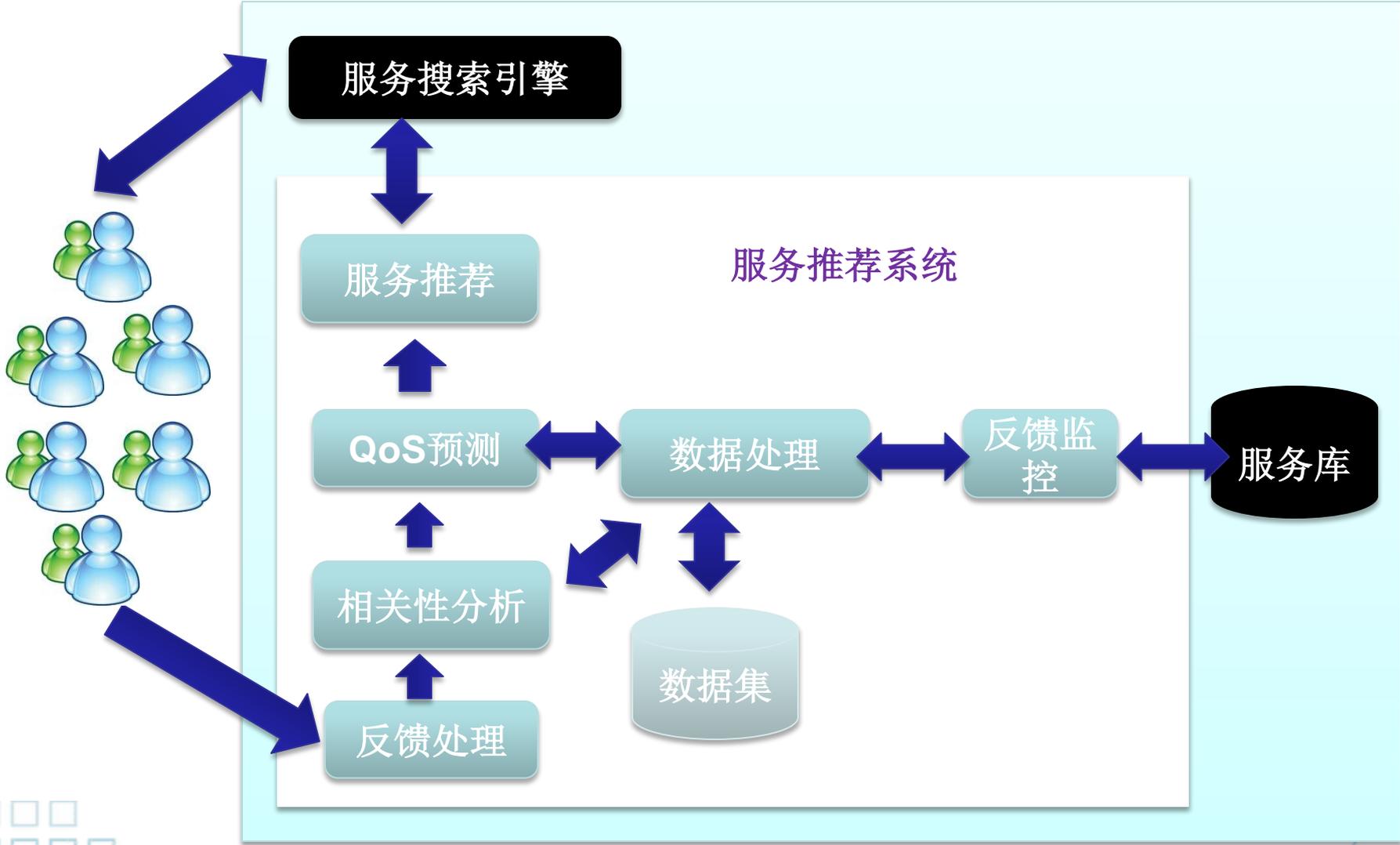
- 前提：基于服务发现，已经得到功能性可满足的服务集合，如何从中选择最优的**QoS**属性服务？
- 挑战性
 - **QoS**属性的多维性
 - 如何同时兼顾多个属性？
 - 针对不同用户，**QoS**属性不完全
 - 如何收集和预测缺失的属性
 - **QoS**属性是动态变化的
 - 如何建立准确的预测模型？



服务推荐

- 目标——挖掘用户反馈信息和服务信息，为用户
提供个性化的服务推荐，提高服务选择的效率





基于协同过滤的推荐算法

□ 基于邻居的方法

□ 基于用户相似度

□ 基于服务相似度

□ 混合上面两种的

□ 第一步： 确立 相似度系数计算方法

□ 第二步： 通过相似度系数选择相似邻居

□ 第三步： 对服务QoS进行预测

□ 皮尔逊相关系数 (PCC)

	s_1	s_2	...	s_m
u_1	0	245	...	389
u_2	2023	342	...	0
...
u_n	0	3040	...	498

$$Sim(u, v) = \frac{\sum_{i \in I_u \cap I_v} (r_{u,i} - \bar{r}_u) \cdot (r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{u,i} - \bar{r}_u)^2 \sum_{i \in I_u \cap I_v} (r_{v,i} - \bar{r}_v)^2}}$$

Zheng Z, Ma H, Lyu M R, et al. Wsrec: A collaborative filtering based web service recommender system[C]//Web Services, 2009. ICWS 2009. IEEE International Conference on. IEEE, 2009: 437-444.

基于协同过滤的推荐算法（续）

□ 基于模型的方法

□ 潜因子模型

□ 矩阵分解

- 它的直观物理意义可以理解为，服务的QoS值不仅依赖于用户或者服务的相似度，而且还依赖于用户和服务之间的相互关系

□ 基于矩阵分解和相似度

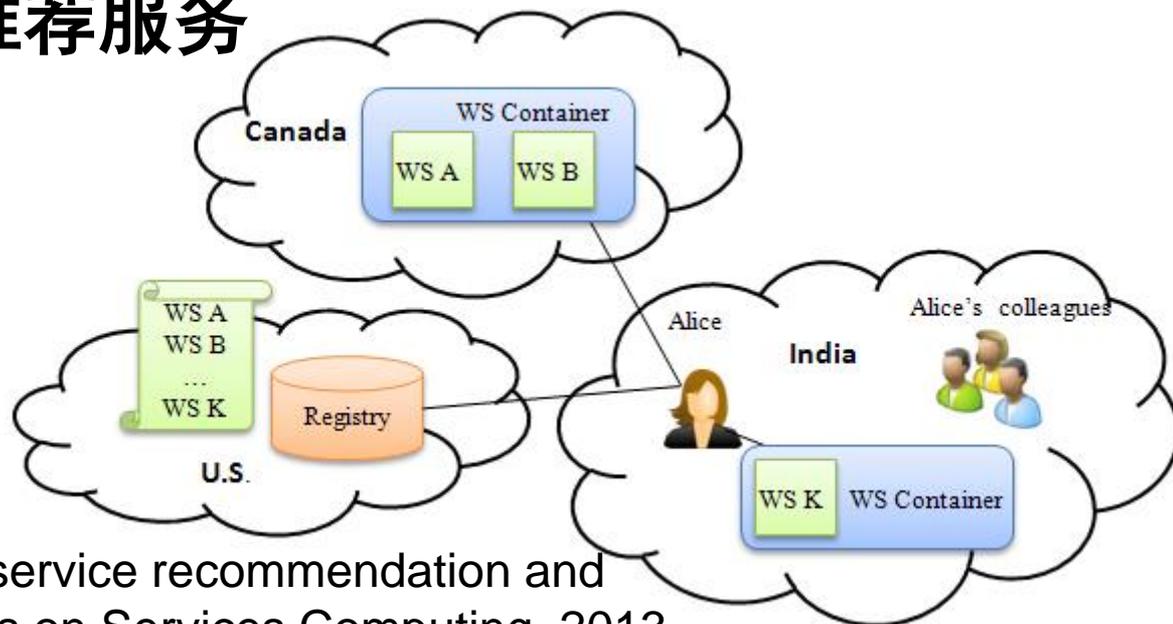
$$\begin{aligned} \mathcal{L}(R, S, U, V) &= \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n I_{ij}^R (R_{ij} - (\alpha U_i^T V_j + (1 - \alpha) \sum_{k \in \mathcal{T}(i)} S_{ik} U_k^T V_j))^2 \\ &+ \frac{\lambda_U}{2} \|U\|_F^2 + \frac{\lambda_V}{2} \|V\|_F^2, \end{aligned}$$

该方法的思想认为用户和服务在多个潜在的特征因子上共同作用了他们所构成的矩阵中的数据，因此可以将矩阵映射到较低维度的因子空间中

Zheng Z, Ma H, Lyu M, et al. Collaborative Web service QoS prediction via neighborhood integrated matrix factorization[J]. TSC2012

- 目标：考虑用户所处区域位置，向其推荐具有最优预测值的服务
- 思想：地理位置临近的用户比距离较远的用户更容易获得相似的QoS体验
- 用户贡献机制：用户向系统提交观察的Web服务QoS信息以获得推荐服务

位置感知的 服务推荐



RegionKNN

- ▶ 输入：用户-服务矩阵（user-service matrix）每一个用户与一个RTT向量相关联
- ▶ 输出：对目标用户（active user）未使用过的服务进行QoS预测，将具有最优预测值的服务推荐给用户
- ▶ RegionKNN算法的两个阶段
 - 区域建模——线下计算
 - 区域特性
 - 区域聚合
 - QoS预测——线上计算
 - 邻居选择
 - QoS预测

	s_1	s_2	...	s_m
u_1	0	245	...	389
u_2	2023	342	...	0
...
u_n	0	3040	...	498

用户-服务矩阵*

区域模型——区域特性

- ▶ 区域：一组用户的集合，同一区域内的用户满足
 - 所处地理位置相近
 - 对相同服务有相似的QoS体验
- ▶ 区域敏感服务
 - $R = \{R_1(s), R_2(s), \dots, R_k(s)\}$ 是所有用户提交的服务 S 的往返时延。服务 S 对区域 M 敏感当且仅当

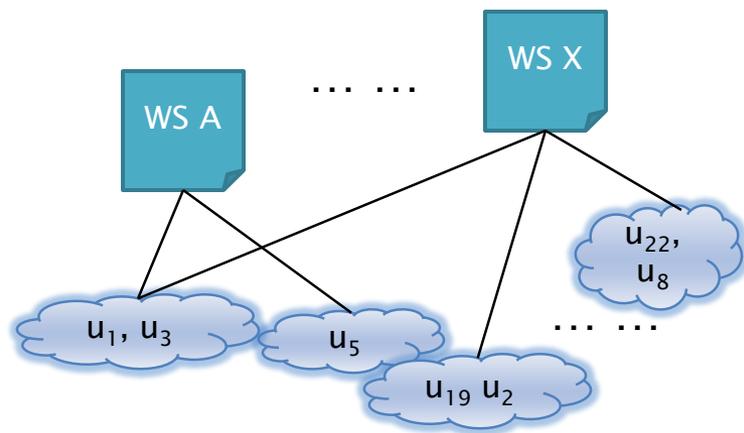
$$\exists R_j(s) \in R ((R_j(s) > \hat{\mu} + 3\hat{\sigma}) \wedge region(j) = M)$$

其中， $\hat{\mu} = median_i(R_i(s))$ $\hat{\sigma} = MAD_i(R_i(s))$

例： {120, 128, 200, 250, 258, 2000, 3500}

u_1 u_3 u_5 u_{19} u_2 u_{22} u_8

$$\hat{\mu} = 250, \hat{\sigma} = 130$$



区域模型——区域特性

▶ 区域敏感度

$$regSen(M) = \frac{|sensitive\ services|}{|services|} \quad (1)$$

▶ 敏感区域

- 区域 M 是敏感区域当且仅当 $regSen(M) > \lambda$

▶ 区域中心

- 区域内所有用户的RTT向量构成的中位数向量

区域模型——区域聚合

- ▶ 动机：初始各区域内用户提供的QoS数据较少，往往导致预测不准确
- ▶ 步骤
 - 根据用户IP地址划分初始区域
 - 每次迭代时，选取最相似(相似性大于阈值 μ)的两个非敏感区域进行聚合，更新该区域中心、敏感服务列表及与其他区域的相似性
- ▶ 时间复杂度
 - 设初始区域个数 l_0 ，算法输入是具有 n 个用户和 m 个服务的满阵，在最坏情况下时间复杂度是 $O(l_0^2 mn \log n)$

区域模型——区域聚合

▶ 区域相似性度量

- 使用区域中心度量两个区域间的相似性
- 协同过滤推荐算法通常使用皮尔逊相关系数（PCC）度量用户间的相似性：

	s_1	s_2	s_3	s_4	s_5
c_m	1	2	5	0	0
c_n	0	0	5	1	3

使用PCC计算两个区域的相似性

$$Sim(c_m, c_n) = 1$$

改进后 $Sim'(c_m, c_n) = 0.2$

QoS预测——邻居选择

- ▶ 邻居：对相同Web服务，有相似QoS体验的用户
- ▶ 基于区域的邻居选择
 - 无需搜索整个用户空间，只需与各区域中心进行相似性计算
 - 各区域特征由区域中心表示

QoS预测

在线计算时间
复杂度 $O(l_j m)$

- ▶ 对于当前用户 u 未使用过的服务 s_i 进行预测 $\hat{R}_u(s_i)$
 - 判断当前用户所属区域，若无合适区域，建立包含该用户的新区域
 - 判断服务 s_i 是否为该区域的敏感服务

- 若是

$$\hat{R}_u(s_i) = R_{center}(s_i) \quad (4)$$

- 否则，计算该用户与有该服务记录的各区域中心的相似性，选取相似度最高的前 k 个区域中心
 - 若用户所属区域有该服务的记录

$$\hat{R}_u(s_i) = R_{center}(s_i) + \frac{\sum_{j=1}^k (R_{c_j}(s_i) - \bar{R}_{c_j}) Sim'(u, c_j)}{\sum_{j=1}^k Sim'(u, c_j)} \quad (5)$$

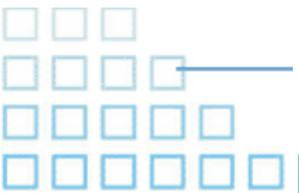
- 否则

$$\hat{R}_u(s_i) = \frac{\sum_{j=1}^k R_{c_j}(s_i) Sim'(u, c_j)}{\sum_{j=1}^k Sim'(u, c_j)} \quad (6)$$

- 否则不进行预测

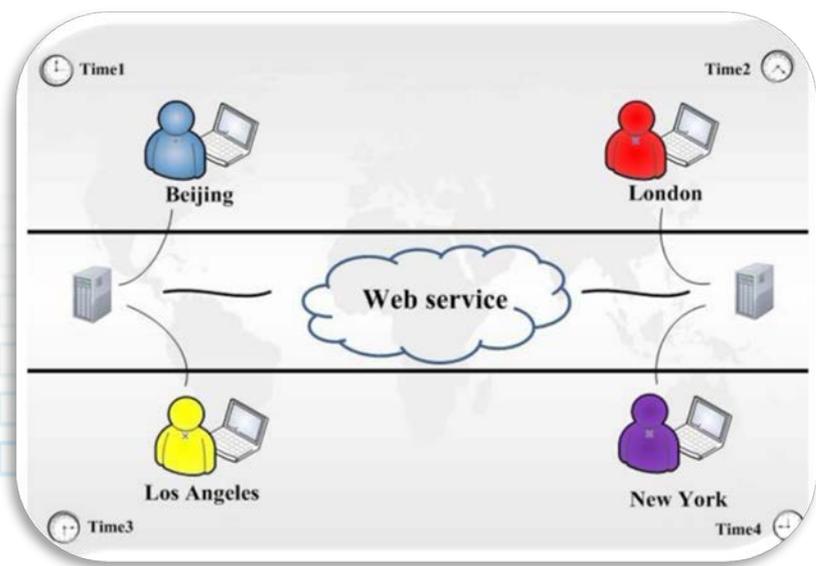
基于非功能性的选择

- 前提：基于服务发现，已经得到功能性可满足的服务集合，如何从中选择最优的**QoS**属性服务？
- 挑战性
 - **QoS**属性的多维性
 - 如何同时兼顾多个属性？
 - 针对不同用户，**QoS**属性不完全
 - 如何收集和预测缺失的属性
 - **QoS**属性是动态变化的
 - 如何建立准确的预测模型？



时空感知的Web服务推荐

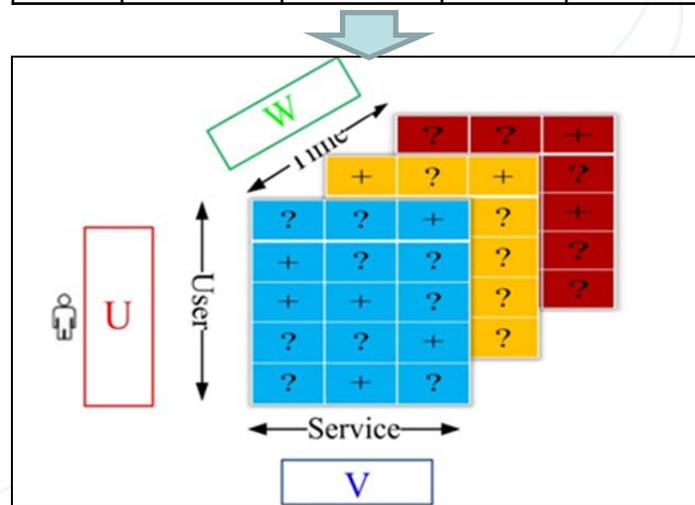
- 问题：如何根据已知用户的QoS，预测新用户的QoS，从而向用户推荐最佳的服务？
- 分析：Web服务的调用的QoS（如：响应时间）不仅与用户地理位置相关，还与调用的时间相关。



用户调用Web服务的过程

矩阵模型：不考虑时间因素

	s_1	s_2	...	s_m
u_1	?	245	...	389
u_2	2023	342	...	?
...
u_n	?	3040	...	498



张量模型：考虑时间因素

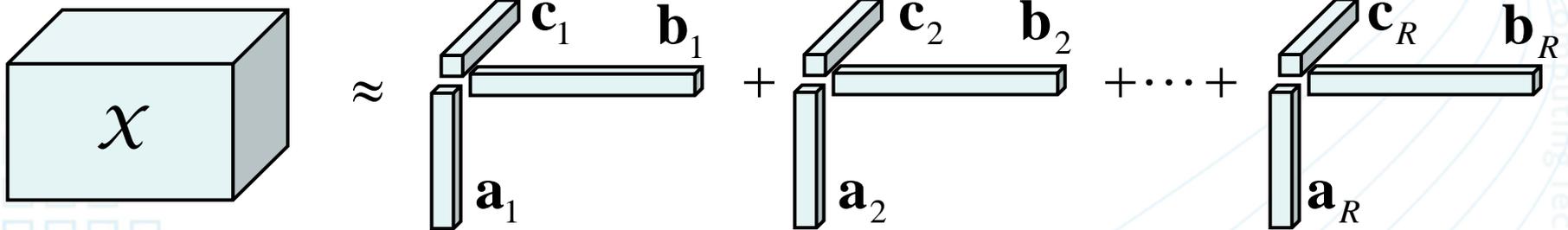
CP分解

□ **CP(CANDECOMP/PARAFAC)**分解的张量形式

□ 其数学表达形式如下

$$\mathcal{X} \approx [\mathbf{A}, \mathbf{B}, \mathbf{C}] = \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$$

□ 将一个张量表示成有限个秩一张量之和，比如一个三阶张量可以分解为



三阶张量的CP分解

非负张量分解 (NNCP)

□ 将NNCP形式化为最优化问题

$$\min_{\mathbf{u}_r, \mathbf{s}_r, \mathbf{t}_r} \frac{1}{2} \|\mathbf{y}_{ijk} - \sum_{r=1}^{Ry} \mathbf{u}_r \circ \mathbf{s}_r \circ \mathbf{t}_r\|_F^2$$

s.t. $\mathbf{u}_r, \mathbf{s}_r, \mathbf{t}_r \geq 0.$

□ 通过求偏导求解这个最优解

$$\frac{\partial f}{\partial \mathbf{u}_l^{(i)}} = \sum_{r=1}^{Ry} \mathbf{u}_l^{(i)} (\mathbf{s}_r \cdot \mathbf{s}_l) (\mathbf{t}_r \cdot \mathbf{t}_l) - \sum_{j,k} \mathbf{y}_{ijk} \mathbf{s}_l^{(j)} \mathbf{t}_l^{(k)}$$

□ 得到因子矩阵的迭代计算公式

$$\mathbf{t}_l^{(k+1)} \leftarrow \frac{\mathbf{t}_l^{(k)} \sum_{i,j} \mathbf{y}_{ikl} \mathbf{u}_l^{(i)} \mathbf{s}_l^{(j)}}{\sum_{r=1}^{Ry} \mathbf{t}_r^{(k)} (\mathbf{u}_r \cdot \mathbf{u}_l) (\mathbf{s}_r \cdot \mathbf{s}_l)}$$

预测QoS值

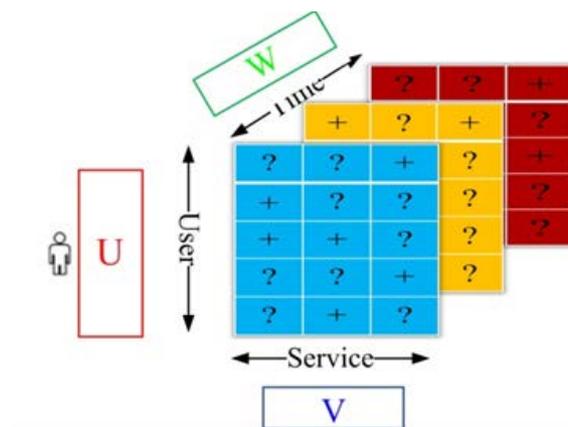
- 得到预测模型后，通过下式得到对原始张量重构后的近似张量，在不同的时间点上预测QoS值

$$\hat{y}_{ijk} \approx \sum_{r=1}^{R_y} \mathbf{u}_r^{(i)} \mathbf{s}_r^{(j)} \mathbf{t}_r^{(k)}$$



预测值：由用户，服务和时间三个维度所确定的下标在矩阵中所处的位置

计算公式：用户，服务和时间三个维度所对应的因子矩阵之积相加



基于非负张量分解的Web服务推荐

• 实验结果

- 数据集：基于PlanetLab收集343个节点对5817个服务的调用数据。每3小时调用一次，共4天。

Table 3: Web Service QoS Performance Comparison (A Smaller Value Means a Better Performance)

WS QoS Property	Method	MAE					RMSE				
		5%	10%	15%	20%	25%	5%	10%	15%	20%	25%
Response time	UMean	0.8156	0.7247	0.7161	0.6758	0.6361	2.3807	1.9589	1.9937	1.6229	1.4217
	IMean	0.5708	0.4919	0.4988	0.4158	0.4083	2.3344	2.0264	2.4146	2.0878	1.7216
	IPCC	0.6861	0.7972	0.5146	0.6014	0.4073	3.8511	3.8336	3.3770	2.5129	1.9188
	UPCC	0.5965	0.6627	0.6625	0.6014	0.5435	2.3424	1.8843	1.9331	1.5129	1.2671
	WSRec	0.5135	0.5252	0.5268	0.3947	0.3717	2.1838	2.0207	2.1533	1.7144	1.2975
	RSVD	0.9162	0.8375	0.8168	0.8088	0.7800	6.6970	5.2284	3.8099	4.9581	3.6419
	NNCP	0.4838	0.3589	0.3254	0.3178	0.3148	1.1470	1.0685	1.0502	1.0434	1.0399
Throughput	UMean	8.3696	8.4262	8.0827	7.7713	7.7113	32.7424	35.3732	32.8413	44.4918	40.9749
	IMean	6.7947	7.0433	6.4606	5.7356	5.2033	33.5447	34.5250	25.6687	22.7903	19.3721
	IPCC	8.2521	8.6508	8.1413	8.8179	8.3416	41.4411	40.9693	37.4096	48.9877	42.6471
	UPCC	8.0533	7.7259	7.1103	7.3437	7.0486	31.8687	32.9089	29.6238	29.2614	25.1004
	WSRec	6.3139	6.2608	5.9656	5.9222	4.7879	23.0171	24.6223	22.4384	22.3709	17.9580
	RSVD	9.6429	8.9885	7.5998	5.6261	5.1030	23.5928	25.4172	20.3695	19.7478	19.9420
	NNCP	6.0007	5.4889	4.9859	4.5001	4.0385	10.8098	10.1738	9.57085	8.98722	8.43047

在推荐的准确性方面，NNCP方法优于其他6种方法。

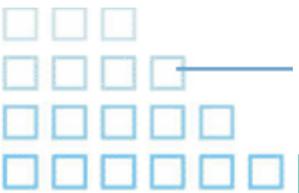
服务综合性评估：可信性

• 目前的Web服务可信性评估主要有两种技术方向

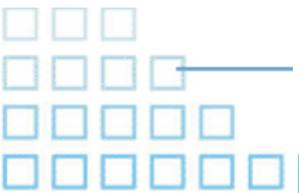
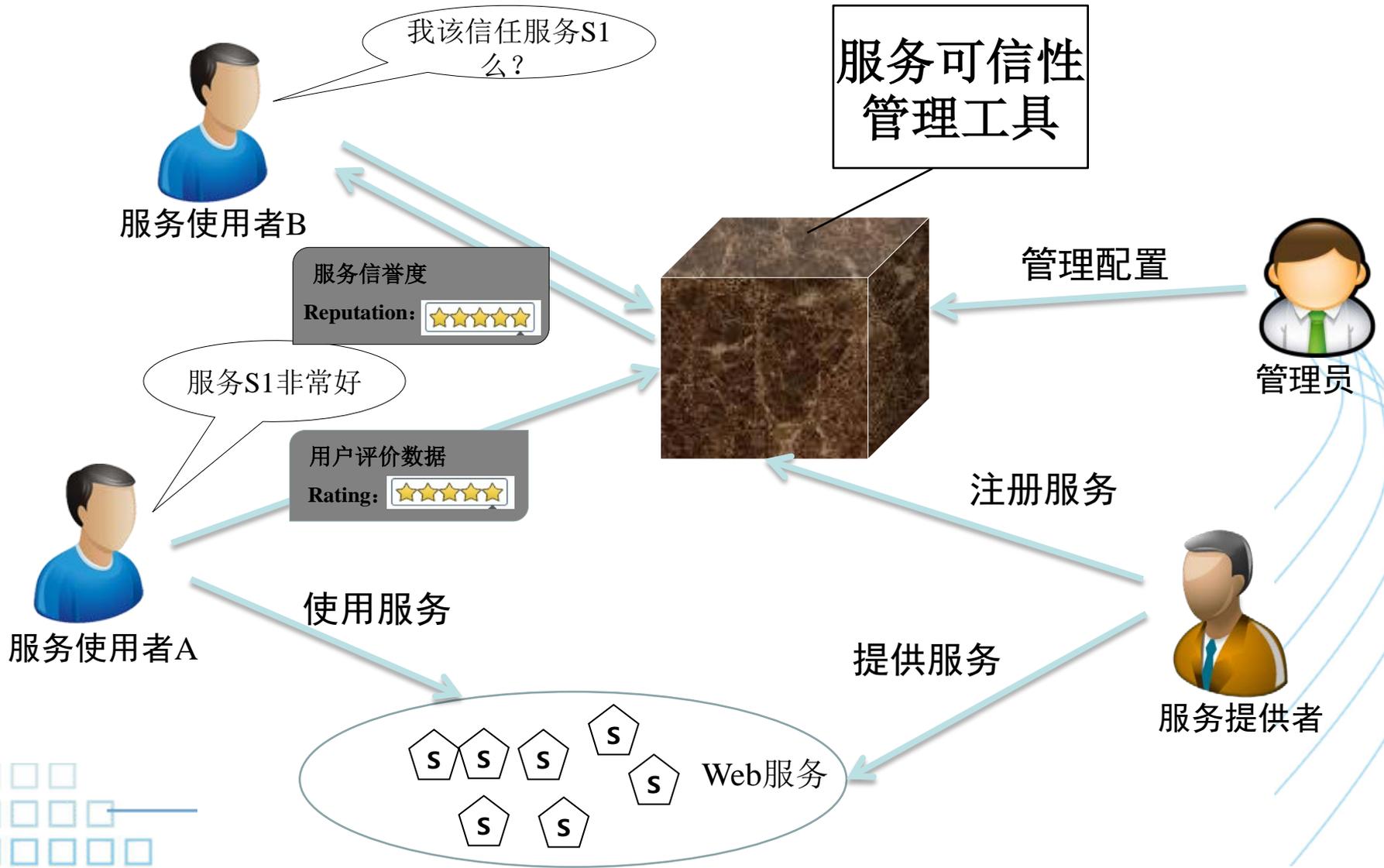
- 收集用户评价，计算出信誉度表征可信性
- 可以评估复杂的属性
- 利用前人经验，无需实际调用服务



- 通过分析测试结果度量可信性
- 只能监测少数几个属性
- 测试服务耗费资源



基于信誉度的可信性评估



结 束！

