

# PODY管理软件

李梦寻 ([limengxun.com](http://limengxun.com))

# 理想的（小）计算集群

- 方便用户使用
  - 易于管理环境：能够自由安装依赖
  - 适合开发：易于连接，方便DEBUG
  - 学习成本低：网上教程多

用户视角

# 理想的（小）计算集群

- 方便管理运维

- 隔离用户环境：避免依赖混乱，安全考量
- 易于升级：支持最新软件
- 学习成本低

- 能有效利用资源

- 避免计算资源闲置
- 支持资源配额

管理视角

# 为什么不用SLURM?

SLURM拥有完善的队列系统，可最大化利用资源，但它架构“重”了一些：

- 依赖高速网络存储
- 维护门槛相对较高
- 用户环境配置不便
- 用户调试体验不佳（例如使用pdb）

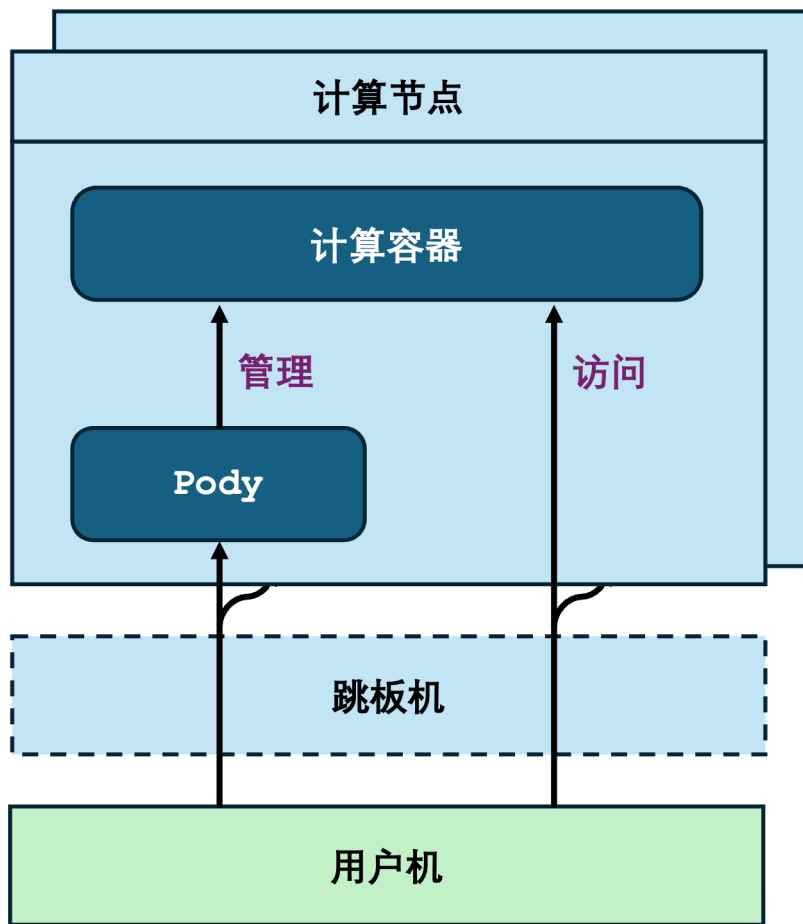
# DOCKER是好选择

- 轻量级虚拟化，轻松支持大量用户
- 用户使用无感，好似在裸金属服务器
- 学习资源丰富，部署、管理、使用好上手

# DOCKER的问题

- 用户数量多，则管理员任务繁重（例如频繁建立、重启、销毁容器，配置启动项等）
- 如直接将管理权限交给用户，则用户权限过高，而且提高了用户学习成本，犯错概率高
- 无任务队列系统

# PODY



- 通过Docker上的一层抽象提供统一的容器管理方案
- 将管理权限给用户, 但是通过策略限制、资源配额避免滥用资源和安全风险

# PODY+DOCKER优缺点

优点	<ul style="list-style-type: none"><li>• <b>部署与维护简便</b>：架构设计简单，整体代码量少，便于快速搭建与维护 [简单是可靠的先决条件。——Edsger Dijkstra]</li><li>• <b>充分复用成熟工具</b>：降低系统的开发与运维复杂度</li><li>• <b>用户上手门槛低</b>：接口设计简洁，功能集中在少量核心容器操作；用户在容器中几乎无感地使用 Linux 环境</li></ul>
缺点	<ul style="list-style-type: none"><li>• <b>资源分配模型有限</b>：按服务器分配用户与管理员权限，适合小规模、多租户场景，不适用集中调度和统一池化的大型集群。无作业队列等复杂资源调度能力</li></ul>