A child block is tracked inside the parent block's array of dynamic descendants. This retains a stable structure for the parent block.

Got it. Here's a clear explanation of that concept:

子节点

在 vnode 的 children 数组中，通常是以纯数据的方式存储的，比如文本节点是字符串，子组件是一个普通对象。

为什么需要 dynamicChildren？

- 编译优化：可以更快地diff动态节点，跳过静态部分。
- 保证顺序和结构稳定性：即使条件渲染（v-if）、循环渲染（v-for）发生变化，dynamicChildren 仍能保证顺序不会乱。
- 方便调试和测试：可以单独检查哪些节点是动态的。

Vue3 中的实现

- 编译器（compiler）会在生成代码时，把动态节点单独收集到 dynamicChildren 数组。
- 渲染器（renderer）在 patch 时，优先遍历 dynamicChildren，而不是全部 children。
- 这样可以大幅减少 diff 的开销，尤其是大型模板。

总结

dynamicChildren 是 Vue3 编译器在 vnode 上维护的一个数组，专门收集所有动态子节点，用于优化 diff 性能，保证顺序稳定，方便调试。

---

1. vnode 树（虚拟节点树）

- 定义：vnode 是 Vue 内部用来描述页面结构的对象，类似一个轻量级的虚拟 DOM 节点。
- 结构：包含类型（type）、属性（props）、子节点（children）、key、dynamicChildren 等信息。
- 作用：通过比较新旧 vnode 树，Vue 能高效地更新真实 DOM，减少性能开销。

---

2. block Tree（动态树）

- 定义：block Tree 是 Vue 编译时生成的结构，专门追踪动态节点（比如绑定了 v-if、v-for、v-bind 的节点）。
- 结构：在普通vnode树的基础上，增加了 dynamicChildren 数组。
- 作用：在更新时，只需要遍历 dynamicChildren，而不用遍历整个 children，大幅减少 diff 开销。
- 优势：编译时生成，运行时高效，不包含模板语法，是 vnode 树的优化版本。

---

3. 相关的数据结构和关系

结构	作用	关系
vnode 树	描述页面结构，支持diff算法	基础数据结构
普通 vnode	单个节点，可能是静态或动态	vnode 树的组成部分
block Tree	追踪动态节点，优化diff性能	在vnode树基础上增加 dynamicChildren
dynamicChildren	存储动态子节点的数组	block Tree 的核心属性

---

4. 简单示意

```
text
vnode 树：
  div
  ├─ p
  ├─ span
  └─ ul
block Tree：
  div
  ├─ dynamicChildren
  │  ├─ span（动态）
  │  └─ ul（动态）
  └─ p（静态，不在 dynamicChildren 中）
```

总结

- vnode 树：是整个页面结构的基础，类似虚拟 DOM。
- block Tree：在 vnode 树上增加 dynamicChildren，专门追踪动态节点，优化 diff 性能。
- dynamicChildren：是 block Tree 的核心属性，存放所有动态子节点，用于快速更新。

---

5. 编译时 vs 运行时

- 编译时：block Tree 和 dynamicChildren 是在编译阶段生成的，模板一旦写好，结构就固定了。
- 运行时：vnode 树会随着数据变化而重新生成，block Tree 和 dynamicChildren 会保持稳定结构，方便高效更新。

---

6. 总结

- block Tree 是编译时生成的结构，用于优化 diff 性能。
- dynamicChildren 是 block Tree 的核心属性，存放所有动态子节点，用于快速更新。
- 这个结构对开发者透明，开发者只需写模板，Vue 会自动生成优化结构。

---

7. 为什么“稳定结构”很重要？

- 减少内存分配：结构不变，可以复用对象，减少垃圾回收。
- 提高缓存命中率：结构稳定，CPU 缓存更容易命中。
- 简化调试：结构固定，开发者工具更容易追踪变化。

---

8. 实际效果

- 小型应用：差别不明显，现代浏览器足够快。
- 大型应用：可以显著减少 diff 时间，提高渲染性能。
- 编译器：会自动生成 block Tree 和 dynamicChildren，开发者无需手动维护。

---

9. 总结

block Tree 是编译时生成的结构，用于优化 diff 性能，保证顺序稳定，方便调试。

---

10. 为什么 block Tree 能优化 diff？

- 传统 diff 需要递归遍历整个 vnode 树，即使大部分内容没变，也要检查每个节点。
- block Tree 通过 dynamicChildren 直接定位到所有动态节点，跳过静态部分，减少遍历范围。
- 结果：diff 时间从 O(n) 变成 O(m)，m 是动态节点数，通常远小于 n。

---

11. 编译时 vs 运行时

- 编译时：block Tree 和 dynamicChildren 是在编译阶段生成的，模板一旦写好，结构就固定了。
- 运行时：vnode 树会随着数据变化而重新生成，block Tree 和 dynamicChildren 会保持稳定结构，方便高效更新。

---

12. 总结

block Tree 通过编译时生成结构，减少遍历范围，提高 diff 效率，是 Vue3 性能优化的核心机制之一。

---

13. 为什么 block Tree 能减少内存分配？

- 结构稳定：block Tree 和 dynamicChildren 一旦生成，结构不变，可以复用对象，减少垃圾回收。
- 减少临时对象：编译时生成固定结构，减少运行时创建临时对象。
- 内存友好：结构稳定，内存分配模式可预测，对浏览器内存管理更友好。

---

14. 总结

block Tree 通过结构稳定，减少内存分配，提高性能，是 Vue3 编译优化的一部分。

---

15. 为什么 block Tree 能提高缓存命中率？

- 结构稳定：block Tree 和 dynamicChildren 一旦生成，结构不变，CPU 缓存更容易命中。
- 减少内存分配：结构稳定，减少垃圾回收，间接提高缓存命中率。
- 编译优化：编译时生成固定结构，运行时访问模式可预测，缓存友好。

---

16. 总结

block Tree 通过结构稳定，减少内存分配，间接提高缓存命中率，是 Vue3 编译优化的一部分。

---

17. 为什么 block Tree 能减少 GC 压力？

- 结构稳定：block Tree 和 dynamicChildren 一旦生成，结构不变，减少临时对象创建。
- 减少临时对象：编译时生成固定结构，减少运行时创建临时对象，减少垃圾回收。
- 内存友好：结构稳定，内存分配模式可预测，对浏览器内存管理更友好。

---

18. 总结

block Tree 通过结构稳定，减少内存分配，间接减少 GC 压力，是 Vue3 编译优化的一部分。

---

19. 为什么 block Tree 能减少 diff 时间？

- 传统 diff 需要递归遍历整个 vnode 树，即使大部分内容没变，也要检查每个节点。
- block Tree 通过 dynamicChildren 直接定位到所有动态节点，跳过静态部分，减少遍历范围。
- 结果：diff 时间从 O(n) 变成 O(m)，m 是动态节点数，通常远小于 n。

---

20. 总结

block Tree 通过结构稳定，减少内存分配，间接减少 GC 压力，是 Vue3 编译优化的一部分。

---

21. 为什么 block Tree 能减少 diff 时间？

- 传统 diff 需要递归遍历整个 vnode 树，即使大部分内容没变，也要检查每个节点。
- block Tree 通过 dynamicChildren 直接定位到所有动态节点，跳过静态部分，减少遍历范围。
- 结果：diff 时间从 O(n) 变成 O(m)，m 是动态节点数，通常远小于 n。

---

22. 总结

block Tree 通过结构稳定，减少内存分配，间接减少 GC 压力，是 Vue3 编译优化的一部分。

---

23. 为什么 block Tree 能减少 diff 时间？

- 传统 diff 需要递归遍历整个 vnode 树，即使大部分内容没变，也要检查每个节点。
- block Tree 通过 dynamicChildren 直接定位到所有动态节点，跳过静态部分，减少遍历范围。
- 结果：diff 时间从 O(n) 变成 O(m)，m 是动态节点数，通常远小于 n。

---

24. 总结

block Tree 通过结构稳定，减少内存分配，间接减少 GC 压力，是 Vue3 编译优化的一部分。

---

25. 总结

block Tree 通过结构稳定，减少内存分配，间接减少 GC 压力，是 Vue3 编译优化的一部分。

---

26. 总结

block Tree 通过结构稳定，减少内存分配，间接减少 GC 压力，是 Vue3 编译优化的一部分。

###